Università degli Studi di Cagliari

PHD DEGREE

Electronic and Computer Engineering

Cycle XXXIV

TITLE OF THE PHD THESIS

Data traffic analysis to monitor and

understand the people's mobility in Smart Cities

Scientific Disciplinary Sector(s)

S.S.D.ING-INF/03

PhD Student: Marco Uras

Supervisor Prof. Luigi Atzori

Final exam. Academic Year 2020–2021

Thesis defence: February 2022 Session

University of Cagliari

Department of Electrical and Electronic Engineering
PhD Course in Electronic and Computer Engineering
Cycle XXXIV

Ph.D. Thesis

Data traffic analysis for monitoring and understanding the people's mobility in Smart Cities

S.S.D. ING-INF/03

Candidate
Marco Uras

PhD Supervisor
Prof. Luigi Atzori

PhD Coordinator
Prof. Alessandro Giua

Final examination academic year 2020/2021

"An investment in knowledge pays the best interest"
Benjamin Franklin

I would like to thank in particular prof. Luigi Atzori, who has always supported me during this work.

Contents

Abstract

The Smart City is an abstract projection of future communities, a virtual fence defined by a set of needs that find answers in technologies, services and applications which can be traced back to different domains: smart building, inclusion, energy, environment, government, living, mobility, education, health, and much more.

In general, the smart city concept involves different ICT technologies and several physical devices which are connected to the Internet of Things to optimize the efficiency of city services and with the objective to improve the quality of life of the citizens. In this context the most strenuous and crucial activity is to collect and organize useful data to plan and improve city's services. In this thesis I will focus on the may cheapest way to collect a high spatial granular human mobility data: Wi-Fi data traffic.

The traffic is analyzed in a passive and anonymous way, the privacy of the users is totally preserved. In the piratical side the Wi-Fi traffic which is relevant for this work is composed by Probe Request frames. Those frames are the packets sent by Wi-Fi enabled devices during the phase of Access Point discovering, in the context of IEEE 802.11 protocol, such phase is called Active Scanning.

The main contribute of my PhD journey was in the design and developing of complete cloud-enabled IoT system and in the design of a MAC address de-randomization algorithm based on the combination of temporal- and content-based probe request fingerprints. The developed algorithm is Artificial Intelligent based and could be used to count and track Wi-Fi devices in an anonymous way.

Introduction

The Smart City area is an abstract projection of future communities, a virtual fence defined by a set of needs that find answers in technologies, services and applications which can be traced back to different domains: smart building, inclusion, energy, environment, government, living, mobility, education, health, and much more.
In general, the smart city concept involves different ICT technologies and several physical devices which are connected to the Internet of Things to optimize the efficiency of city services and with the objective to improve the quality of life of the citizens. In this context the most strenuous and crucial activity is to collect and organize useful data to plan and improve city's services.

To understand why collecting mobility data in an organized way is so useful. We can resort on some use-cases for two major types of organizations in the city: businesses and municipalities. For the former, the main reasons are those of customer segmentation, organization of warehouses, having a clear view of customers' flows and defining strategies for managing events. For the latter, a very important factor is to plan and optimize public transport, sometimes to protect the environment or simply for public safety reasons. These reasons translate into 3 actions that see data collection and analysis as their milestone: Monitoring, data-driven planning and implementation of measures. Therefore, in this context, the most difficult and crucial activity is to collect and organize useful data for planning and improving the services of the city. For simplicity, we divide the types of data that can be collected into two broad categories, subjective data and objective data. They are, respectively, coming from questionnaires or interviews carried out in written, verbal or digital form. The second category, object of this thesis, can come mainly from measurements carried out by human personnel in the field or automatically from sensors, which can sometimes be connected to the Internet.

Wi-Fi traffic data has been a valuable source of information on mobility and crowd behavior for years, mainly through capturing the MAC address of our personal devices. Indeed, by analyzing frames generated by the different sources identified a different MAC addresses, it was possible to estimate the number of people in the area under analysis and track them. However, for obvious privacy concerns, from 2017 on-wards there has been a wide diffusion of some techniques that make the MAC address no longer globally unique but time varying and random. These measures, although aimed at safeguarding the privacy of users, have made all the

studies based on Wi-Fi analytics done in the field of mobility up to now in vain, making the Probe Request present outside the range of action of an Access Point a real "digital junk". Therefore, the major efforts made in the last 3 years for the drafting of this thesis have been to find an inexpensive and easy to implement way to make Probe Requests still be a resource but respecting privacy.

In this thesis I studied and developed a solution for data collection and analysis of the Wi-Fi frames in order to understand the volumes, density and travel times of people in urban areas. The solution was called People Mobility Analytics and represents a complete IoT solution to provide evidence on mobility in a simple and complete way to all decision-makers participating in the smart city scenario. To obtain this result, it was necessary to have a deep knowledge of the Internet of Things technological chain in all its functional blocks, such as the management and design of devices, design and implementation of transmission channels, design and implementation of the infrastructure for the analysis of data. In particular, around this last block, the skills of Data Science are crucial to extract real in-depth knowledge starting from raw data.

Over the course of the three years, the solution has undergone some changes. The following is the organization of the Thesis: after an introductory and background part on the IEEE 802.11 protocol, in the first chapter will focus on the first version of the People Mobility Analytics (PMA) system where all the indicators were obtained from the MAC addresses considered as globally unique. The second chapter sees an implementation of the PMA system in different real contexts, placing an important focus on localization algorithms based on RSSI and on arrival time differences (TDOA). The third chapter introduces the phenomenon of the randomization of the MAC address, and proposing an Artificial Intelligence solution to manage the phenomenon of randomization. The fourth chapter presents a real-world use case or that of local public transport, which exploits the PMA solution complete with a derandomization algorithm. The derandomization algorithm object of this thesis carries forward the state of the art on this topic, thanks to its applicability in real contexts and to the mix of different techniques that have previously been studied individually. The last chapter describes further analyzes made on vehicular traffic data, provided as Open Data by the Municipality of Cagliari, which this time do not come from Probe Request but from coils immersed in the asphalt.

Chapter 1

Background

1.0.1 MAC structure, Probe request and Information Element

Within computer networks, each terminal, for its operation within the network, must necessarily have a physical address called MAC (Media Access Control) which in origin has had the characteristic of being globally unique.
Fig. 1.1 shows the structure of a MAC address, in particular its six octets of bits. The octets are divided into two groups: Organization Unique Identifier (OUI) and Network Interface Controller (NIC). Respectively, the most significant bits are assigned by the IEEE to the producer and the least significant bits are assigned by the producer. This mechanism allows to identify the MAC address manufacturer and to assign each manufacturer with a unique address space from where take the needed addresses for each device.
An interesting part concerns the second least bit in the first octet, highlighted in red in the Fig. 1.1. In general and in particular WiFi applications if this is set to 0, then the MAC address should be globally unique and it is kept constant over the time. Otherwise, when this bit is set to 1, the MAC address should be locally administered which means that the MAC address is randomly generated and may change from one session to another, i.e. we can consider it such as a *virtual* MAC address.

A device that wishes to know which WiFi networks there are in the its nearby, sends a ping message called Probe Request. Specifically, it sends a burst of these

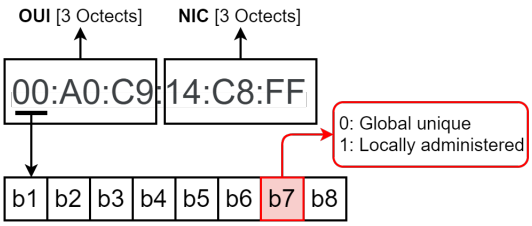

Figure 1.1: Global unique and Locally administered bit detail of a MAC address.

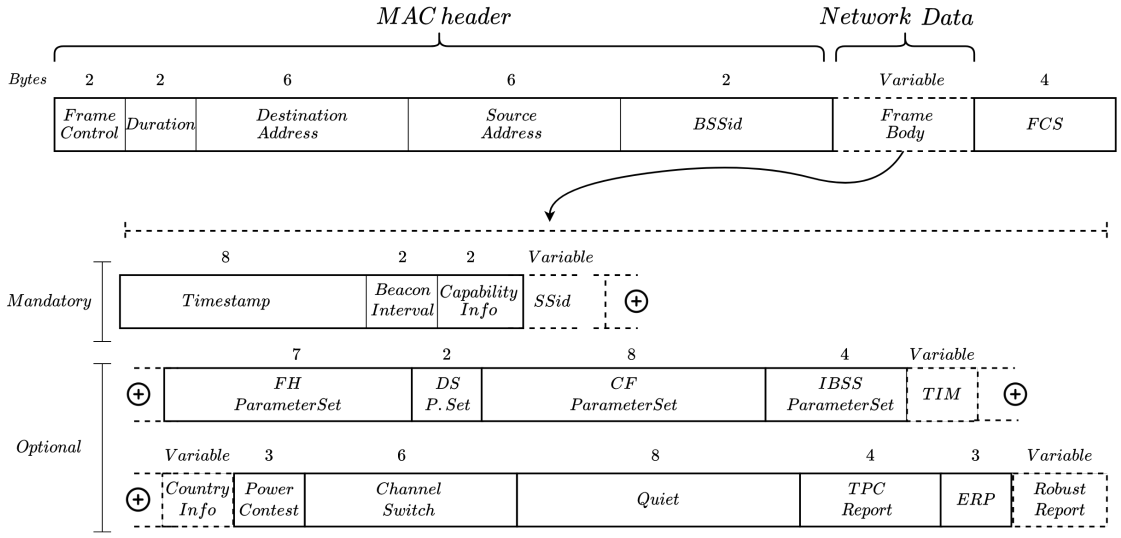

Figure 1.2: Management frame structure

messages and wait for a time limit within which it must receive reply to connect to the network. Every Access Points (APs) that receive these frames within the established time replies to the device by sending, in its turn, a Probe Response Frame with the information necessary to establish the connection. The Probe Request and the Probe Response are two sub-types of a particular frame called Management Frame which is divided into MAC Header and Frame Body. In the whole Frame Body, each field could has fixed length, or variable lengths both types of IE are called Information Elements (IEs). All IEs are labelled with an identification number and its size; the structure for each IE is defined by the standard[1] and follow the TLV (type-length-value or tag-length-value) encoding scheme. The first octet of the Information Element is reserved for the Element ID, the second defines the whole information element's Length and the remaining bits contain the information (Value). Accordingly, each IE conveyed in the Probe Request Frame is identified by its ID and its length, which indicates the number of octets used by the IE content.

1.0.2 MAC address Randomization

In the Wi-Fi context, randomization of MAC addresses is the process of generating virtual MAC addresses by end-devices during the phase of active scanning for an Access Point. Such activity is designed and performed to guarantee that devices' real MAC address remain unknown, and as reseult preventing users tracking issues. However, when the AP and the device find themselves, they set up the connection and only after this moment, the device uses its real MAC address due the fact which only starting from that moment the entire communication is encrypted. In detail,

[1] `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7786995`

Table 1.1 Element IDs

Element ID	Name
0	Service Set Identity (SSID)
1	Supported Rates
2	FH Parameter Set
3	DS Parameter Set
4	CF Parameter Set
5	Traffic Indication Map (TIM)
6	IBSS Parameter Set
7 (802.11d)	Country
8 (802.11d)	Hopping Pattern Parameters
9 (802.11d)	Hopping Pattern Table
10 (802.11d)	Request
11-15	Reserved
16	Challenge text
17-31	Reserved
32 (802.11h)	Power Constraint
33 (802.11h)	Power Capability
34 (802.11h)	Transmit Power Control (TPC) Request
35 (802.11h)	TPC Report
36 (802.11h)	Supported Channels
37 (802.11h)	Channel Switch Announcement
38 (802.11h)	Measurement Request
39 (802.11h)	Measurement Report
40 (802.11h)	Quiet
41 (802.11h)	IBSS DFS
42 (802.11g)	ERP information
43-49	Reserved
48 (802.11i)	Robust Security Network
50 (802.11g)	Extended Supported Rates
32-255	Reserved;
221	WiFi Protected Access [b]

the MAC address randomization process is managed by the device Operating System (OS); there are no standard algorithms for this process, and each OS implements its custom randomization operations.
Linux OS introduced the MAC address randomization starting from version 3.18 of its kernel [Gru]. Most of WiFi drivers are configured to change the MAC address every 60 seconds [MCRV16]. Moreover, the Linux OS has three methodologies for assigning a MAC address: use the real MAC address, use a completely virtual MAC address or use a partially virtual MAC address keeping the first three octets equal to the real OUI of the network card manufacturer.
Windows OS use another type of randomization, where support for randomization of the MAC address was indeed one of the most important innovations regarding wireless networking at the time it was announced in 2015[2]. The effort done in [w10] explains how randomization works on Windows 10, which shows that random MAC addresses are used in the Probes Requests and also during the Authorisation and Association phases before the network connection.
Google introduced the randomization of the MAC addresses starting with version 6 of Android OS; in version 8 it is enabled by default in every device and Google's own implementation uses a fixed set of virtual MAC addresses for network discovery[3].
Apple introduced the MAC address randomization in iOS since version 8. Based on laboratory experiments with various models, iOS devices randomize the MAC address for each burst of Probes Requests (generally composed by 2-5 probes). In general, as shown, each manufacturer implements proprietary algorithms; the consequence is a high variability of the MAC addresses randomization process.

Currently, for privacy reasons, the 95% of the devices [RNNS20] in the active scanning phase, i.e., when the device sends the Probe Requests, hide their *factory* (also called *real* in the following) MAC address using a randomly generated one. As mentioned above, this is done in order to guarantee that the real MAC address cannot be tracked over time. Although, some devices seems send their factory MAC address after connection process, new policies have been introduced in the latest versions of some operating systems allowing devices not to reveal their real MAC address even when already connected to an AP. More over, the MAC address randomization process is becoming more and more widespread and it is regulated by IEEE standards. This is the reason why, in Table 1.2 are compared the behavior of the Android Q, iOS 14 and Windows 10 operating systems with reference to the major actions. It can be noted how the randomization process when sending Probe Requests is always turned on in Android and iOS devices and cannot be turned off. These represent the OSs used by the majority of smartphones on the market, accordingly making the randomization almost a very useful feature to recognize a device dedicated to human use (such smartphones or smartwatch). Deeply understanding in the practice part the randomization and to keep in mind the basic concept, let's

[2] `https://channel9.msdn.com/Events/WinHEC/2015/WHT201`
[3] `https://source.android.com/devices/tech/connect/wifi-mac-randomization`

say that it can be performed daily in some cases, i.e., the same address is kept for the whole day or it could be changed every burst.

1.0.3 Related Works

In the literature there are several solutions that exploit WiFi traffic to count people or track devices, based on the simple tracking of the MAC address. Most of those solutions are out of date and do not take into account the effects of MAC address randomization or partially manage this aspect. Over time, the diffusion of MAC address randomization adopted by manufacturers has exponentially increased, reaching the 95% of devices on the market. Therefore, it is necessary to introduce techniques to counteract the phenomenon of randomization.

Fingerprinting

In [MCRV16] the author proposed a way to fingerprint the probe requests sent by a single device. The device's fingerprint is calculated using the Information Elements (IEs) contained within the probe requests. Inter-burst times were also added to this information. This allowed to obtain a more accurate analysis. Subsequently, the similarity between the various fingerprints is calculated and analyzed. In order to test the tracking tool through fingerprints, some datasets of scans made at different times were used. To be precise, two datasets created by the authors and a public dataset of the Sapienza (last update 2013-09-10). Each fingerprint correspond at one device. The results showed that they were able to count the detected devices with an accuracy of 75%. Franklin et al. [FMT+06b] proposed a system that analyzes the inter-frame time of packets allowing the creation of a device fingerprint. Any of the approaches introduced above have the same main problem. Use timing as feature could add errors in clustering and classification, timing information is not reliable due to scattering problems and multi-path phenomena that occur in real-world environments because those phenomena introduce some random delays between probes and bursts.

Active Sniffing Methods

Several authors have proposed or carried out tests using active sniffing methodologies, that is to say methodologies which involves action performed by the observer in order to change the behaviour of the observed.
In [VMC+16], the authors have analyzed various actives sniffing techniques in order to track smartphone reducing the error introduced by the virtual MAC addresses on over 170 thousand MAC address, generating over 8 million probe request. However, the approach that has given the best results uses the parameters related to WiFi Protected Setup (WPS) for the devices that support it. A parameter called Universally Unique Identifier-Enrollee (UUID-E) has been found to derive directly

from the device's real MAC address, which is the MAC address assigned by the manufacturer. However, the same work shown that a device's WPS UUID-E can often be used to derive its real MAC from the randomized MAC address. Martin et al. [MMD+17] analyze various techniques that can be used on a large scale to be able to trace random MAC addresses to a single device. In particular, active sniffing methods exploit various vulnerability and made attacks such as KARMA attack [WNDDZ06] (creation of fake Access Point from the list of probe BSSIDs) and RTS/CTS attack [SN14] in order to obtain the true MAC address during the negotiation of the connection with an AP, this approach requires the device's known SSID as knowledge of the attacker.

Physical Layer Analysis and Radio-Frequency Fingerprints

Other research has applied concepts of fingerprinting also to the ISO/OSI physical layer. In [BBGO08] Brik et al. they propose a technique that is able to identify the origin interface of an 802.11 frame by performing a passive analysis of radio frequencies. In particular, machine learning tools are used in order to have an accuracy of 99% for devices counting, but this approach is good only in a laboratory environment and it is useless if applied to a real world environment because of the high radio interference and the needing of a complicated setup to collect data.
The other technique it could be used is the Radio-Frequency Fingerprints (RFF) and the consideration underlying the use of it [SNYK20], often referred to by the (limiting) term of Spectral Signatures, is that physically different analog circuits, even if driven by the same signal, produce different output signals. The reason lies in the inevitable tolerances of the passive components and, even more so, in the differences between active components even with the same code and production lot. Effect that is stronger in radio-frequency components and in particular in power ones. To cite the most common example [SNYK20, US07], the IEEE 802.11b standard provides, before each transmission, a preamble with a gradual increase of the power up to the nominal power. Of course, the same principle can also apply to cell phone transmissions and are not limited to WiFi protocol. The final amplifier (and antenna) of different card manufacturers are obviously different. But what we have seen is that, due to the differences between the various components of the circuit, even different boards of the same model produce sufficiently different latch signals.

In other words, it is possible to associate a pair [Model-Serial Number] in a biunivocal way with a coupling signal. Obviously, this association can only be complete if it is acquired the signal knowing the Serial Number. But less invasive associations can also be helpful. The typical application of RFFs is for authentication in a LAN. If the lock signal of a particular device (PC / Tablet / Smartphone) is recorded in a controlled environment, it will be very easy to find it later. The procedure (described at block level) therefore provides:

- signal acquisition;

- extraction of a set of features from the signal and / or its spectrum;
- comparison of the signal features with those of the signals present in the database: if present, the time of the last acquisition is simply updated; if absent, the time of the first acquisition is entered, with the associated time;
- The attendance count (at a given moment) is done directly from the database.

The signal to be acquired is an analog signal. Therefore, the features to be extracted may instead be different depending on the radio system of interest. From a technical point of view, the sampling rate for the acquisition must then be chosen (which, however, does not is too critical), and the problem of identifying the instant of beginning of the transitory, a problem on which there is a fairly vast literature [SNYK20].

Privacy aspects

In last decade, the topic of privacy has become a prominent issue in any system that collects and processes data, particularly user-related information. In Europe, the General Data Protection Regulation (G.D.P.R. n. 2016/679[4]) defines the data content that can be exploited to identify an individual as "Personal Identification Information (PII)", providing a specific indication of which data should be considered as personal information.

In the Internet of Things arena (but not limited to it) MAC addresses and IP addresses have always been a problem for users privacy, due to lack of regulations in this regard. However, after the enactment of GDPR, both IP and MAC addresses must be treated as PIIs (art. 4 of GDPR regulation). In order to understand if there are privacy issues due to the data acquisition, we rely on the system defined by European legislators which is based on risk assessment (for the rights and freedoms of natural persons) deriving from the specific treatment of personal data.

Security measures must be implemented by the person who has been tasked with the role of "data controller". Data controllers must define security measures on the basis of a risk assessment. Furthermore, the data controllers must always provide maximum transparency on the purposes and methods of processing personal data. They must allow the data subject to control data processing, by making the rights provided for in the regulation easily and effectively manageable. Therefore, a careful analysis of the specific reference context is necessary in order to respect all the phases of the data treatment. In this case a preliminary assessment is needed on the type of data processed, to select only the data necessary to pursue the purpose of the processing. Unnecessary data must be deleted and data for which it is not necessary to maintain a connection with the identity of the persons must be made anonymous. Instead, the information that may be needed to reconnect to the

[4] `https://www.gdpr.eu`

persons concerned must be pseudo-anonymized. In this way, the data controller can reduce risks and apply appropriate countermeasures. Accordingly, to ensure people's anonymity, in the first version of People Mobility Analytics solution as soon as the MAC addresses are collected these are pseudo-anonymized by replacing these with dummy identifiers by applying the PBKDF2 encryption. Doing this process the original MAC addresses are not stored neither in the sensors nor in the cloud and cannot even be traced back.

Furthermore, in the context of this thesis and in the final version of People Mobility Analytics system, which is the main focus of mine PhD, it is mandatory to spent a few words on the collected data. In particular, although originally the MAC address uniquely identified a device (every Ethernet or wireless network card produced in the world has a unique MAC, with some exceptions), it is important to keep in mind which such address not identify a user; but this is no less relevant, as due to the uniqueness this can be used as a key for the crossing of multiple data relating to the user himself (previous tracking, visits to websites, online purchases, registration to services, etc.) and then ultimately to derive its identity. Therefore, pursuant to the General Regulation for the protection of personal data 2016/679 (GDPR), the MAC address is considered personal and must be treated as such. However, in recent years, in order to protect the privacy of their customers, the manufacturers of Wi-Fi and Bluetooth devices intended for human use and in this case of Smartphones, Tablets and Smartwatches, have implemented random MAC address generation techniques. This means they are no longer globally unique and each device continues to change it over time. In this way it is no longer possible to use it to identify people, only some older devices still keep the MAC fixed. On that note, as part of the People Mobility Analytics project in its final configuration, non-personal data are extracted. In fact, data with fixed MACs are discarded, also because they are no longer in use for devices dedicated to day-to-day human use. The other data, which in the follow will be called virtual or random MAC addresses, are analyzed to extract anonymous information on the number of devices present in an area of interest and how they move in this area. This analysis is very complicated due to the anonymization process mentioned above, and it is configured as a challenging task for the mobility research.

Table 1.2 OS behaviour comparison for MAC address randomization

Randomization behavior across latest releases of Operating Systems			
Action	**Android Q**	**iOS 14**	**Windows 10**
Probe Mode Randomization	ON. The users cannot change this setting.	ON. The users cannot change this setting.	OFF by default. The users can set it ON/OFF.
Randomize Daily	Optional	Not performed	Optional
Connection to an unknown SSID	On the first connection a random MAC is generated.	On the first connection a random MAC is generated.	*Randomization ON*: On the first connection a random MAC is generated. *Randomization OFF*: Factory MAC is used.
Connecting to a known SSID	When disconnecting and reconnecting, the same *virtual* MAC used for the first connection is used.	When disconnecting and reconnecting, the same *virtual* MAC used for the first connection is used.	*Randomization ON*: When dis/reconnecting, the same *virtual* MAC used is used. *Randomization OFF*: When dis/reconnecting, the factory MAC is used.
MAC Randomization for a specific SSID Disabled	Device is automatic reconnected to SSID with factory MAC address.	Device is automatic reconnected to SSID with factory MAC address.	You need to manually reconnect to the SSID and the device uses the factory MAC address.
MAC Randomization for all SSID Disabled	N/A	N/A	The device uses the factory MAC address.
SSID Profile Forget and Reconnection	If the SSID is forgotten, the device generates and uses a new random MAC address to connect.	If the SSID is forgotten, the device generates and uses a new random MAC address to connect.	*Randomization ON*: the device generates and uses a new random MAC address to connect. *Randomization OFF*: the device uses the factory MAC address to connect.

Chapter 2

Wi-Fi probes as objective data source for people mobility monitoring

In this chapter is shown how Wi-Fi Probe Request could be an important resource for humans mobility monitoring. In fact, the analysis of people mobility in urban contexts is of key importance to tackle major issues in different fields, such as those related to urban planning, citizen safety, telecommunications services planning, public transport service deployment. Such an analysis should provide key indicators, such as crowd density per area of interest, people flows, recurrent mobility patterns, mobility heat maps, just to cite a few. In the present chapter, it is described the first version and the original idea of the People Mobility Analytics (PMA) solution, which relies on the monitoring of Wi-Fi traffic. The main features of the solution are the followings: preservation of user privacy, extraction of key metrics on people behaviour, presented through charts and heatmaps. Extensive sessions of monitoring and analysis have been carried out in three different scenarios: an university campus during classes, an international fair, and a roundabout in a urban context.

2.1 Introduction

Studying human mobility is becoming more and more important because understanding the demand for mobility allows us to better plan major mass services, such as the public transport services [DPS+16] and the communication infrastructure [KBCP11], but also planning appropriate urban and green areas [FLN+18]. When addressing the issue of extracting data of people mobility, it is useful divide urban mobility into two broad categories, city-wide and buldings-wide depending on the scale of people mobility. The first relates to how people move in the city, while the second is how they move within large buildings or building complexes. The work carried out in this chapter focuses on the first category of mobility. As Wi-Fi devices

are very popular with people moving around a city, to obtain a good representation of the majority of the population behaviour we can consider the Wi-Fi data traffic a good source if information. However, the best way to collect large amounts of data with the least amount of resources are crowd-sensing and crowd-sourcing techniques. The objective of these techniques is to collect data directly from the people themselves through their personal devices, where an appropriate app is required to be installed [GWY+15], this kind of collecting data is named partecipatory corwd-sensing. The main drawback of participatory sensing in the crowdsensing approach is the need for the people whose mobility has to be observed to have an active role in the process as they have to install the relevant software. This requires many people to be involved as the people to be monitored is not known a priori. This also requires the user to be motivated to participate to the collecting campaign, often through rewards.

To overcome this issue, the passive approach to the collection of people mobility traces is adopted. In this context, low prices of Wi-Fi network interface cards are an attractive incentive to use Wi-Fi as the basis for a passive data collection system for user localization and significant research has been conducted over the last 15 years in this area [BP+00][KJBK15].

The system developed during mine research, People mobility Analytics, relies on the analysis of Wi-Fi probe requests, the network scanning package in IEEE 802.11 (Wi-Fi) used by both clients and Access Point (AP) in order to "see" each other. Depending on who initiates the communication, the client or the AP, the scanning is either active or passive. During an active scan, the radio client transmits a probe request and listens to a probe response from an AP. With a passive scan, the radio client listens on each channel to the beacons package sent periodically by an AP. Generally a passive scan takes longer times, as the client has to listen and wait for a beacon instead of actively probing to find an AP. Another limitation with a passive scan is that if the client does not wait long enough on one channel, then the client may lose an AP beacon.

Among other things, the probe requests contain information on the APs known from the device and the MAC address of the Wi-Fi interface. In this chapter it is showed how this information can be used to create traces of mobility, bearing in mind that usually the maximum range of Wi-Fi communications varies between 35 and 100 meters [KO17], it depends by what type of antennas is used, to estimate densities and flows within cities.

The major contributions of the PMA system in its first version are the followings:

- the design of a novel architecture for an easy deployment of Wi-Fi based people monitoring and analysis; the definition of real-time and post-processing mobility metrics;

- the analsyis of data collected from three different pilots, i.e., an university campus during classes, an international fair, and a roundabout.

The novelty with respect to the state of the art consists in the possibility of extract information about citizen's behaviour and mobility through real time and off line metrics, simply using the IoT technological chain and without the need for applications to be installed into the users' devices, and and simply looking at a web application that shows the results of the analysis. In section 3.2, it is presented a brief analysis of the state of art for people mobility monitoring and Wi-Fi localtization techniques; in section 2.3 it is described how the PMA (People Mobility Analytics) works with the major functionalities of data analysis; section 3.4 presents the experimental analysis; section 3.5 draws final conclusions for this chapter.

2.2 Past works

In the past years, many techniques have been proposed that observe the Wi-Fi traffic to monitor the position of people through their devices. Some of these make use of fingerprinting to achieve a better good accuracy especially in indoor environments [MVFB10]. However, this approach is not applicable to the case of a smart city environment.

Still with RSSI fingerprinting, but without the need to perform a survey of the studied environment, Potortì et al. in [PCG+18] obtained remarkable results in localizing people in indoor environments, such as shops inside a mall or an open space office exploiting the existing Wi-Fi network and making traffic analysis. The drawback of such an approach is that, when the devices is already connected to the network, probes are sent only occasionally [Fre15] this makes it technically difficult to implement localization solutions based on the only probe requests Wi-Fi in indoor environments.

Despite the technical complexity, there are solutions that exploit Wi-FI probe requests to implement footfall monitoring applications such as those done by Xu et al. in [XSK+13]. The system acquires information on a smartphone's MAC using wireless sniffing and uses an RSSI (Received Signal Strength Indicator) based localization method for positioning. The purpose of this system is to monitor pedestrian traffic and monitor the density of people based on tracking smartphones in a street and to explain how this information can be used to improve the service provided to people as a better bus time. Of course, nowadays those kind of approaches are totally out of date and useless. A similar approach is also proposed by [XSK+13], where the focus has been on the impact of many factors that influence the radio performance in this type of small environment, such as slow fading and fast fading. Those kind of phenomena impact on both the packet's RSSI and the time of arrival, which are the main features used for derive the position of a target. Other studies have been carried out in [SWM14], which suggest a hybrid approach for position estimation based on RSSI-based and time-based approaches. The time-based approach also considers the moment when a MAC address was captured on a monitor node. Based on the evaluation performed, they conclude that both Bluetooth and

Wi-Fi can be used to obtain approximations of the crowd mobility.

In [TJMK18] Traunmueller et al. demonstrate that Wi-Fi probe requests can be used to analyze external mobility and human trajectories in a large and densely populated urban area with high spatial resolution and time frequency. They use a dataset of Wi-Fi probe requests collected from 54 public APs for a week in Lower Manhattan in New York, NY, collected through the "Quantified Community" testbed. Demonstrate how these data can be used to analyze common trajectories, indicating the intensity of street activity over time. Again, although this work is relative recent (it is made in 2018), at the time when I writing this thesis such results are not more exploitable.

In [CDBvS18] the authors focused on the problem of separating the points where an individual stops (named stay points) from their movements (named trajectory). They placed 40 Wi-Fi sensors in the city of Assen, Netherlands, during the TT Festival[1] and collected Wi-Fi tracking data for each public holiday in the years 2015, 2016 and 2017. The raw tracking data consists of a set of positions with date and time. Because of the scarcity of data and considerably long intervals between the surveys, even if represented on the map, over time, it is difficult to make sense of the data.

2.3 The first version of PMA system

The first version of the PMA system relied on the fact that the percentage of smartphones favoring randomization in 2017 was only 40% [RNNS20]. Therefore, at the time, doing that type of study could still have statistic relevance. Fig. 2.1 shows the general architecture of the first version of PMA system, which is composed of two main modules: the cloud server and one or more station clusters that perform data collection. For simplicity, we could think of the sniffing station as a data collector while the cloud platform has the task of performing data storage and analysis, providing statistics and displaying them.

Fig. 2.2 shows the details of the solutions with all the functional blocks. The PMA Station includes all the functionalities necessary for the collection of probe requests and for their prepossessing before being sent to the cloud; moreover the device configuration part and the connectivity one is represented. The two upper blocks, Server and Storage Replica Set, are the cloud part of the system. In particular, the data storage part is implemented through a distributed storage infrastructure to make the system robust and reliable. The server block takes care of data operations, providing communication interfaces and the display part.

The following subsections provide more details of the system together with an analysis of the data flows.

[1] `https://www.ttfestival.nl/`

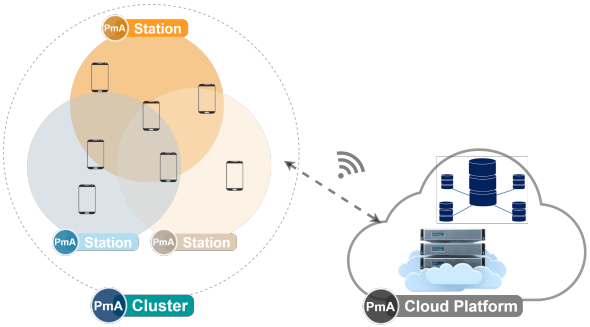

Figure 2.1: First version of PMA system: high-level architecture.

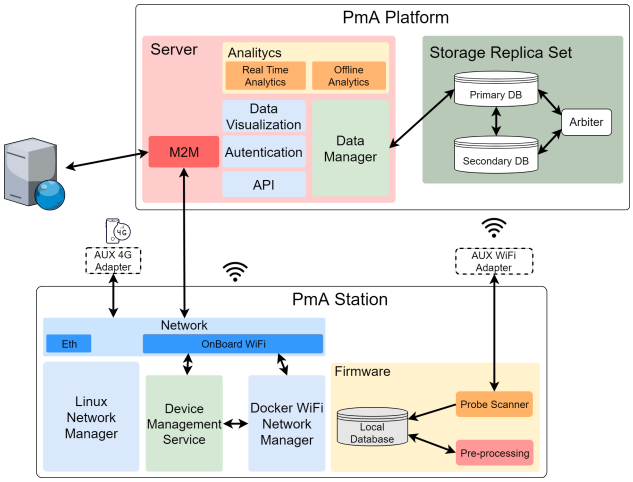

Figure 2.2: Detailed system architecture

2.3.1 The PMA Station in details

The sniffing station is the sensor connected to Internet network which collect data. It is made of cheapest hardware and it is composed of the following basic components:

- Raspberry Pi3 Model B+
- 1 Wi-fi USB with SMA (SubMiniature version A) connector and High Gain antenna (3dBi).

Connectivity to the Internet is provided through three possible interfaces:

- Ethernet
- Wi-Fi
- Mobile (up to 4G LTE)

The station can also be powered by batteries, mains power supply or PoE. These features make the station sufficient flexible to allow installation even in places that are complicated from the point of view of energy or connectivity. For this particular data collection campaign it was adopted a battery pack to power supply and a Wi-Fi connectivity to transmit data.
The software running in the single board computer consists of a firmware which, through a Docker container, manages a sniffing service and at the same time provides a configuration captive portal which can be accessed thanks to a Wi-Fi network created by the device itself.

The first configuration of the station was made through a management interface. There are two possible cases, the first is that the station has already been created in the cloud platform and is represented only by a virtual object, while the second case is the one in which the virtual dual does not yet exist. In both cases the configuration can be performed through a guided procedure, divided into various steps. The first of these will require authentication in order to understand whether the user is registered or not on the cloud platform where the virtual objects software will execute, if it is, the list of stations associated with that account is shown.
At this stage it is possible to choose whether to create the virtual object from scratch or download an existing virtual object configuration. In both cases the flow moves to to the next step, in the case of new station configuration it is requested to fill the form with the new configuration information or load a predefined configuration. Starting from this moment the station is correctly configured and ready for collect data.
The data captured during sniffing is partially processed before being forwarded to the platform, in particular on board the station the MAC addresses captured are pseudo-anonymized in order to avoiding incurring user's privacy problems. The packet that is sent to the server presents the following information in JSON format with information regarding: station ID, MAC, Time Of Arrival, Probe Packets. Furthermore, after sending to the platform, every trace of the sniffed packets is deleted. Therefore the permanence of the data on the device depends on the scanning window, set by default at 15 seconds and however configurable by the user.

2.3.2 The cloud services in details

The visualisation and data processing platform is hosted entirely on a cloud space. To be precise, we have chosen to use the Google Cloud Platform[2] one. The infrastructure built consists of 4 virtual machines configured as follows:

- 1 instance used for data processing and data visualization

[2] `https://cloud.google.com/`

- 3 machines equipped with MongoDB database, 2 of those are used for writing/reading data and the other one to used as referee machine for database management

The databases is configured in a Replica Set mode. Basically, a Replica Set is a set of MongoDB instances which contain the same mirrored data. In this type of configuration there is always only one instance called primary and a certain number of secondary instances, in the case above it is only one. The basic operation is quite simple, the primary server receives all write operations from clients, once written it performs asynchronous replication of the data on the secondary servers. The referee server, in this case, has the task of establishing the new primary server, this is done by continuously contacting the databases through calls called "heartbeats". If, for any reason, the primary server becomes unreachable by not responding to the "heartbeat", the referee takes care of electing a primary from all the available secondary computers using a certain algorithm, causing the system to become available again with all the pre-existing data (Automatic Failover management).
Therefore the advantage of using a Replica Set is that as long as there is an active node, the data will be available. This configuration also automatically provides: Data backup, one for each secondary server; Automatic Failover, that is the ability to ensure that the network continues to operate following a failure in the primary; the ability to read from secondary nodes avoiding overloading a single node for reading. All data operations are performed on the processing server in order to obtain useful information and statistics for mobility and crowding analysis.

2.3.3 Data Flow

Figure 2.3 shows how from the raw data we obtain the refined information, such as crowd density or people flow.

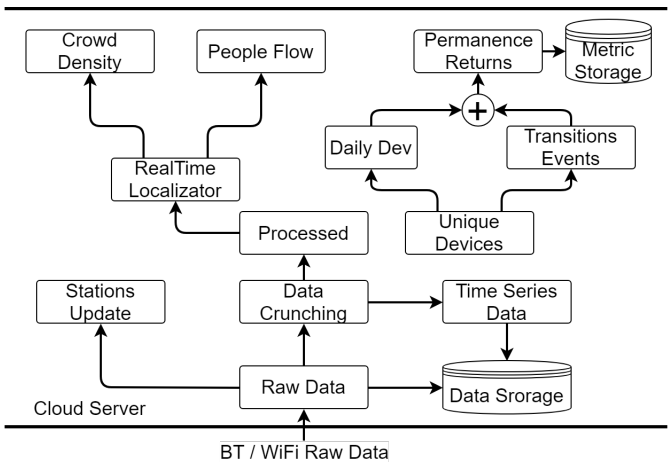

Figure 2.3: Data Flow

Each module is responsible for increasing the value of data, as it is analyzed, understood and interpreted at a higher level of abstraction. The Data Crunching makes a Time Series from raw data, but also stores represents which devices have left the station's coverage range in order to understand all transitions events (such going in or out coverage range). It's also useful for updating the table of uniquest devices seen during day, from this information we are able to understand how many people return to the same station and after how long, or how many people have remained near the station and how long. These metrics will be explained in detail later in the following.

2.3.4 Data analysis

The data collected by the stations are basically the probe requests produced by devices that have a Wi-Fi interface turned on. Of the "probe request" type packages, only a part of the transmitted information is used; some is useful for information extraction and some other is used to allow some device to perform MAC address randomization performed by some types of devices (iOS and some Android models) to preserve user privacy. This process, if not handled correctly, would cause a distortion of the extrapolated information bringing to a number of devices present, in the area of interest, mismatching the real value. For this reason, and considering the percentage of fixed MAC address over the virtual one [RNNS20], in this first phase, it was decided not to consider random MACs in the count. To recognize whether we are facing a random MAC or not we used vendor tables included in Wireshark and constantly updated. The idea is to do a comparison between the detected vendor code and those in the table, if the result is positive, the MAC is counted, otherwise it is discarded. This allows any MAC address to be filtered in the preliminary phase which, if analyzed without derandomization, would cause a distortion of the extrapolated information upwardly distorting the number of devices present in the area of interest with a consequent inconsistency between the actual crowding and that detected by the system.

The information extracted from the data concerns number and anonymized IDs of devices and their position, from which we also compute mobility patterns together with people returns and dwell time metrics, as described in the following. As to the people position, it is extracted on the basis of the position of the sniffing stations for which at least 3 sniffing stations are needed. We dived the metrics into two categories:

- Real-Time metrics:
  - Device position: the analysis of the collected data provided evidence how Wi-Fi probe requests could be exploited to localise devices. In the PMA system, it is used a range based algorithm derived by the Friis' transmission formula using the frequency of Wi-Fi communication (2,4 Ghz);

  - Device count: the time trend of the number of devices detected by a single sniffing station. It is possible to do custom aggregations to view the number of unique devices in different time windows;

- Post-processing metrics:
  - Site returns: this metric represents how many people come back to a given stations after different periods of time, Exactly after 5, 10, 30, 60, 120, 240, 480 minutes;
  - Site permanence: similar to the previous metric, the system takes into account how long people stay near the stations;
  - Crowd density: thanks to the position estimation obtained from the system, it is possible to obtain the crowd density as shown in section 2.4.2 by means of a heat map.

2.3.5 Device position and crowd Density

The analysis of the collected data provided evidence how Wi-Fi probe requests could be exploited to localise devices. The proposed positioning algorithm belongs to the family of range-based algorithms, which is based on the RSSI value contained within the probe request. Major efforts have been done by the scientific community to understand and improve those technique [PMLC16, BBS17, YBC05], S.Knauth provides a very well done study and evaluation of RSSI based positioning algorithm [Kna19]. In the PMA system, it is used a range based algorithm derived by the Friis' transmission formula using the frequency of Wi-Fi communication (2,4 Ghz), reported in equation 3.1, where it is figured out how to calculate the distance d. The used formula is the following, considering omni-directional antennas:

$$d = \frac{\lambda}{4\pi\sqrt{\frac{P_{rx}}{P_{tx}}}} \tag{2.1}$$

where λ is the wavelength computed at 2,4 Ghz and P_{rx} and P_{tx} are the power of receiver ad transmitter, respectively. Given a set of MAC address seen by sniffing stations during a scanning window $\mathcal{S}_w$, we can define a MAC address group $\mathcal{M}_\sqcup$ seen by multiple stations during a time-slot t in the scanning window S_w. Each element of $\mathcal{M}_\sqcup$ is represented by:

$$m_{t,i} = \{mac_i, s_{lat,i}, s_{lon,i}, P_{rx,i,t}\} \tag{2.2}$$

where s_{lat} and $s_{lon,i}$ are respectively latitude and longitude of the station who "seen" the MAC address mac_i, finally $P_{rx,i,t}$ is the power contained within the Wi-Fi probe request. From this information is possible to compute a trilateration in order to derive an approximation of the Wi-Fi device's position by means of the algorithm

Algorithm 1 First version of PMA positioning algorithm

```
1: for all t ∈ S_w do
2:     for all mac ∈ M_t do
3:         itx_{t,mac} = Intersections(mac)
4:         if itx_{t,mac} = Null then
5:             Set P_{t,mac} with Friis's based positioning
6:         else
7:             Set P_{t,mac} = CoG(itx_{t,mac})
8:         end if
9:     end for
10: end for
```

4. Depending on how many stations have been detected from the same device, the intersection points of the circumferences built by the trilateration could be zero. In this case our algorithm chooses a random point that lies on the circumference with radius given by the Friis's formula, considering the power contained in the probe request. Otherwise the algorithm calculates the centre of gravity (CoG) of the polygon resulting from the intersections of the circumferences $itx_{t,mac}$. Thanks to the position estimation obtained from the algorithm, it is possible to obtain the crowd density as shown in section 2.4.2 by means of a heat map.

2.3.6 Devices Counting

The device count is supplied real time. The count is showed by means a time trend of the number of devices detected by a single sniffing station. The time window can be customized to select the desired time interval and could be set with intervals of 1 minute, 10 minutes, 30 minutes, 1 hour, 1 day. The number of devices detected in the most recent scan is also provided with the percentage of change compared to the previous scan and the number of unique devices detected up to the time of display with the percentage change compared to those detected the previous day.

2.3.7 Site Returns

This metric represents how many people come back to a given stations after different periods of time. Exactly after 5, 10, 30, 60, 120, 240 or 480 minutes. The algorithm take into account a set of probe requests with a time reference collected in a selected *day* for each station i, the set is described in 2.3.

$$Probe_{day,i} = \{mac, timeslot_i\} \tag{2.3}$$

The analysis of those time series provides the elements to discover interesting information about the people behaviour and mobility near the stations. I this subsection

we will figure out how time series 2.3 could be analyzed, in order to discover how many people come back to the stations during a day.

Algorithm 2 PMA Devices Returning Algorithm

Require: $Probe_{day,i}$ descending order
  for all $timestamp_i \in Probe_{day,i-1}$ **do**
    $\Delta_i = timestamp_i - timestamp_{i+1}$
    if $\Delta_i > 5minutes$ **then**
      assign Δ_i to its bin
    else
      continue
    end if
  end for

2.3.8 Site Permanence

Similar to how we have seen in the previous section, the algorithm 3 take into account how long people stay near the stations. The idea is to sum the adjacent Δ_i less than 2 minutes then map this sum to correspondent bin.

Algorithm 3 PMA Permance Algorithm

Require: $Probe_{day,i}$ descending order
Require: $count = 0$
Require: $perm = 0$
  for all $timestamp_i \in Probe_{day,i-1}$ **do**
    $\Delta_i = timestamp_i - timestamp_{i+1}$
    if $\Delta_i < 2minutes$ **then**
      $count++$
      $perm+ = \Delta_i$
    else
      if $count > 3$ **then**
        map Δ_i into its bins
      end if
      $count = 0$
    end if
  end for

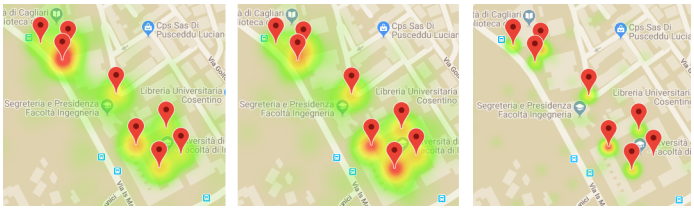

Figure 2.4: Crowd Density in the University of Cagliari

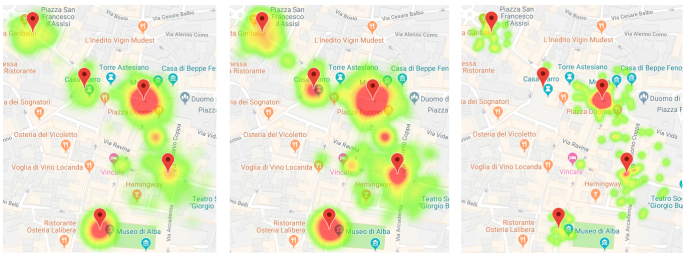

Figure 2.5: Crowd Density in the truffle fair in Alba

2.4 Experimental analysis

2.4.1 Experiments setup

The data were acquired on three different scenarios, in a nerve centre of the city of Turin, on the occasion of the truffle festival in the city of Alba and in the engineering faculty of the University of Cagliari. Each of these installations was created for a specific use case.

- Turin: counting, monitoring site returns and stays, flow of vehicular traffic;
- Alba: counting, pedestrian flow, stays and Site returns at the monitored points;
- Cagliari: estimate of the position of people in the air under investigation.

In each site a different number of devices has been installed:

- Turin: the roundabout to be monitored has 6 confluent arteries. A sniffing station was installed for each road with a distance of about 20 meters from the entrance to the roundabout. In such way is possible to understand the entry and exit points of the cars. By analyzing the time series it is also possible to see after how much the same car has returned and if it has returned, or by analyzing the times it is possible to estimate whether the identified MAC belongs to a pedestrian or a vehicle;

- Alba: 5 points of interest have been identified within the historical centre which are known to be the main points of attraction during the period chosen for the experimentation. For each site a sniffer has been installed that carried out the monitoring of the MAC transited in such point. In this case it was possible to obtain mobility patterns, in particular the path taken by each individual MAC was analyzed starting from the first time it was identified.

- Cagliari: 8 sniffers have been installed that cover the area of the park at the center of the engineering faculty to monitor overcrowding during the day. In this case the number of people actually present was counted. Furthermore, through a RSSI-based localization algorithm, the estimate of the position of each MAC address present was calculated. The position of a flagged MAC address was then compared with that calculated by the system to analyze the average error using, as in this case, the minimum number of stations needed to make a triangulation.

2.4.2 Performance analysis

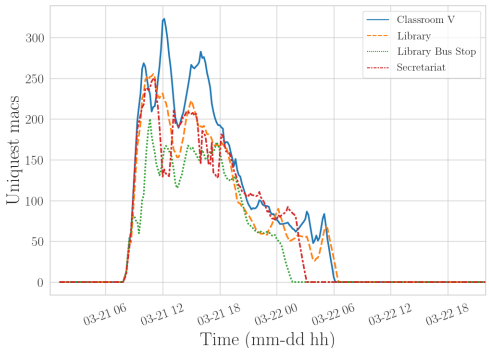

Figure 2.6: Uniquest MACs Distribution (University of Cagliari)

Below it is analysed the data collected from the various stations. In particular, in this section we present the data relating to a cluster of stations in Turin (6 stations) and a cluster in deployed in the University of Cagliari (4 stations) showing site permanence, site returns, devices with unique density of crowding. The latter is also shown for the scenario of Alba. Starting with the analysis of 2.8 and 2.9, indicating site returns and site permanence respectively, we immediately notice a peak of about 120 unique MACs that returned after a 2 hour interval. This is also visible in 2.6 where you see three peaks related to classroom V. These peaks correspond to the times of the morning classes (10am and 12am) and the afternoon (4pm). It should be noted that on the other hand, as at 2 pm, there are a smaller number of devices detected just in conjunction with the lunch break time, while in

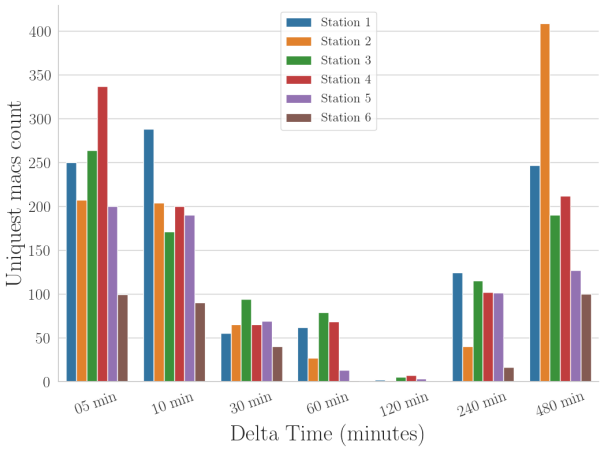

Figure 2.7: Number of returning devices (Turin)

the next hour the devices tend to increase until they reach the 16 hours peak. This justifies the number of site returns shown in the 2.8. Another interesting factor shown in 2.9 is related to the 5 and 10 minute time intervals showing how students may have taken breaks from 5 to 10 minutes. Another interesting data is given by the station installed near the secretariat which shows the 5-10 minute breaks and the lunch breaks of 30-60 minutes made by the employees. Interesting considerations cloud be done for figure 2.7, the interesting point here is provided by returning devices. We can see a considerable amount of people who come back to the stations after 480 minutes, this is attributable to the people who returns after a working day.

Through the coordinates estimated by the system it is possible to understand the crowding density of the places under investigation. In the fig.2.4, three snapshots of the monitoring day related to the engineering faculty campus were extrapolated. The first map indicates the crowded density at 11, the second at 14 and the third at 20. It can be seen that in the first map the most crowded area appears to be related to the cluster A rove there are two of the largest classrooms on campus and the library. In the second one the most crowded area is where there the group of 4 station on the right of the fig.2.4, where the park is very populated at lunch time. While the third shows a very inferior crowding compared to that of the others indicating, given the time, how the crowding less than the previous because nobody follow classes or work near the stations.

In fig.2.5 we can see the crowding density related to the installation performed in Alba on the occasion of the truffle festival. The last two days of the festival and the following day were taken into consideration. All three maps show crowding at 12:30 on those days. In the first two days there was the festival and the crowding was very similar, indicating the most crowded place was the square of the cathedral where the main events took place. In the third map, relative to the day after the

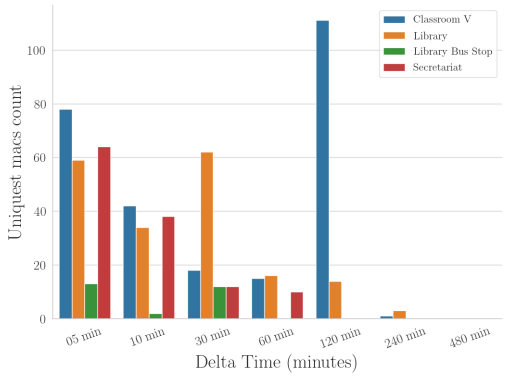

Figure 2.8: Number of returning devices (University of Cagliari)

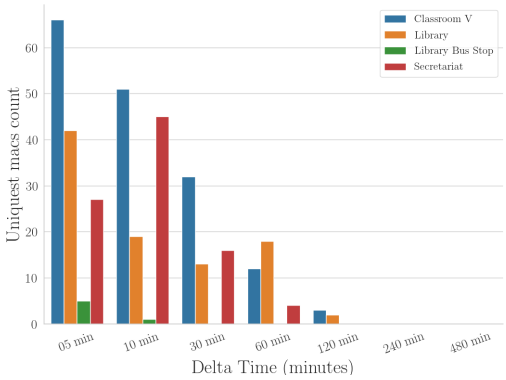

Figure 2.9: Number of stationary devices (University of Cagliari)

festival, we can see that the density of people near the stations has significantly decreased, indicating that the flow of tourists has dropped considerably compared to the previous two days.

2.5 Final consideration on the first version of PMA system

The aim of this work was to provide an easily achievable system to study the mobility and behaviour of crowds, through the use of a low-cost infrastructure and without the need to involve the user with the installation of applications and consequence without having to convince him to use it. The solution implemented has enabled the data required to be collected in a totally passive way in order to extract information on the crowding of a particular place. The system has proved to be quite reliable based on the monitored scenarios and the data that have been obtained and

analyzed. An idea for future works is to manage the randomization of MAC address during the scanning phase in order to improve the accuracy of data and the insight discovered from them. In the following a algorithm to manage the randomization of MAC address will be presented.

Chapter 3

Probe Request based approach for device localization

This chapter presents a real-world application of the first version of the PMA system, which collects probe requests generated by Wi-Fi devices when scanning the radio channels to detect Access Points. The PMA system processes the collected data to extract key insights on the people mobility, such as: crowd density per area of interest, people flows, time of permanence, time of return, heat maps, origin-destination matrices and estimation of people positions. The major novelty with respect to the state of the art is related to new powerful indicators that are needed for some key city services, such as security management and people transport services, and the experimental activities carried out in real scenarios. Furthermore, in this chapter has been analyzed different approach for devices localization, which are an upgrade from the very first version of the PMA system. The work done in this chapter was the result of teamwork, what I personally took care of, in addition to everything already done in the first version of PMA and reported in the previous chapter, was to carry out the localization of devices via Wi-Fi extending the topic to a passive approach based on the difference of arrival times (TDOA). The following text is partially extracted from the paper PmA: A real-world system for people mobility monitoring and analysis based on Wi-Fi probes submitted to the Journal of Cleaner Production n. 270.

3.1 Introduction

In last decades, more and more people have been moving from rural to metropolitan areas. As a results, UN estimates that 55% of world population already lives in cities and the projection shows that the urbanization index is expected to increase to 68% by 2050 [UN18]. The increasing number of people living in big urban conglomerates introduces increasing complexity in the deployment and management of services infrastructure and in the allocation of the appropriate resources to reach the required

sustainable urban living conditions. Luckily, the rapid developments in Information and Communications Technologies (ICT), including Big Data, Artificial Intelligence (AI), and Internet of Things (IoT), is contributing to the Fourth Industrial Revolution [VAB18], laying the foundations to turn our cities into Smart Cities [Top10]. In fact, the mentioned technologies are enabling significant improvements in terms of security, people mobility, health and overall citizens life quality. One of the main applications for this kind of technologies is the collection, analysis and interpretation of urban mobility data. Studying human mobility allows for making more efficient, larger-scale services, such as the public transportation service [DPS+16], the communication infrastructure [KBCP11] but also planning appropriate urban and green areas [FLN+18].

In order to get a good representation of citizens' mobility, it is mandatory to gather a large number of points, capturing people's position over the time. The easiest way to collect large quantities of data with the minimum effort in terms of time and costs is to use crowd-sensing and crowd-sourcing approaches. The main concept of these approaches is to exploit people's personal devices to extract different types of data, which can be achieved through a dedicated app installed in the user smartphone [GWY+15]. However, the main disadvantage of both participatory sensing and opportunistic sensing is that users have to play an active role in data collection because they have at least to install the app and, in some cases, they even have to provide required input. Additionally, this approach often requires awards to be given to the involved users.

In order to properly address these issues, a viable approach is to collect people mobility data using a passive approach, which does not require users to respond actively. In a passive data collection scenario, sniffing the packets sent by devices using the Wi-Fi technologies plays an important role thanks by its low implementation costs; this is the reason why significant research effort on people localization using this approach has been carried out in the last 15 years [BP+00, KJBK15].

Major studies that have been carried out so far focused on the following aspects: real-time devices localization; trajectory tracking and people density; raw data analysis to remove useless data. Still, this research field needs significant efforts to attain practical, robust and accurate solutions. In particular, there is a need to devise the appropriate processing that, starting from the raw data, can generate the information necessary for addressing the city challenges. Additionally, there is the need to perform extensive real-life deployment to learn from the wild. The data collection module should also be respectful of the monitored persons' privacy.

To advance further in this area, I have designed, developed and tested the PMA (People Mobility Analytics) system, which I have initially outlined in [UCA19b], and explore in depth herein. The main focus of the PMA system is to localize people and deduce key insights about the mobility of the crowd.

Specifically, it has been relied on the analsysis of the *probe request* packets that are sent by the user Wi-Fi devices when looking for Access Points (APs) to connect to. These probe request frames contain key information about the APs visited in the

past (Preferred Network List - PNL - although more and more rarely [DPČ19]) and the end device itself (e.g., the MAC address of the Wi-Fi interface). Our study aims to determine how this information can be used to reconstruct traces of mobility, estimate crowds' density and people flows key indicators.
The primary contributions of this work are as follows:

- design and development of an architecture for a Wi-Fi based plug and play solution for people mobility monitoring and analysis;
- definitions of real-time and post-processing metrics useful to understand people habits and behaviour;
- performing intensive experiments in several real world scenarios, i.e., university campus, international fairs, and roads.

The rest of the chapter is organized as follows: in section 3.2 it is briefly analysed the past works on people mobility monitoring and Wi-Fi localization techniques; in section 3.3 it is described the procedures designed and implemented to analyze the raw data; in section 3.4 it is presented the experiments that it has been done and the relevant results; finally, in section 3.5 are drawn final considerations about results and indicate future work possibilities.

3.2 Related works

As discussed in the first chapter, the use of Wi-Fi probe requests for location analytics and people tracking has been gaining attention in the literature [VÇG+16, RC18, WCHZ18, DLMS16, SCJ19]. This is the reason why in this section it is provided a brief summary of recent works, which are categorized in Table 3.1.

Table 3.1 Recent literature for Wi-Fi probe analytics.

Category	**References**
Localization Techniques	[MVFB10, SWM14, Aro77, SZT08, VH10, LO17]
Wi-Fi Localization	[XSK+13, PCG+16a, TJMK18, LMM18, KO17]
Trajectory Tracking	[CDBvS18, RC18, TJMK18, ANL+18, PCG+18]
Crowd Density and Flow	[PCB+17, GLRC19a, GLRC19b, XZL+14, KO17]

3.2.1 Localization Techniques

Over the last ten years, a number of well-known techniques have been adopted for Wi-Fi localization, i.e. RSSI-based ranging (Received Signal Strength Indicator), Time of Arrival, Time Difference of Arrival, Angle of Arrival, and so forth. However, other techniques which are in general more accurate, are also used in order to

localize people, especially in indoor environments. For instance, RSSI fingerprinting, as shown by Martin et. al. [MVFB10] is a very common technique for indoor positioning and localization. Nevertheless this type of positioning is inefficient in urban area scenarios such as the scenario considered in our work.

A completely different approach is to extract parameters from target signals that are depending on the position of the target itself. In the case of Wi-Fi protocol, some of those parameters are present in the probe request frame. Regardless of the technical complexity in the implementation of a pedestrian-monitoring application, Xu et al. [XSK+13] have provided an excellent example of this type of system. Their solution uses MAC address and RSSI information, acquired by Wi-Fi sniffers, in order to localize people and to study their mobility, with the purpose of improving busses scheduling. In the same work, very good considerations have been done about the effect of environment-dependent factors like slow and fast fading.

Schauer et al. [SWM14] figured out how to estimate the position using hybrid techniques based on RSSI and Time of Arrival information of both Wi-Fi and Bluetooth interfaces.
About Time of Arrival and Time difference of Arrival, several studies have been done [Aro77, SZT08, VH10, LO17], but only recently these have been applied to Wi-Fi packets [PCG+16a, TJMK18, LMM18, KO17].

3.2.2 Trajectory Tracking

As regards to the tracking of trajectories, Chilipirea et al. [CDBvS18] focused on how to recognize the points where people are stationary along to a predefined path. In their work, they deployed 40 Wi-Fi sniffers during the TT Festival[1] and collected data in three editions of the festival, i.e. from 2015 to 2017.

In the recent years progress with Machine Learning and Artificial Intelligence have brought enormous advantages in Wi-Fi probes analytics, as shown by Redondi A. et al. in [RC18]. In their work the authors used clustering algorithms in order to find the users' profiles, i.e., habitual or sporadic users, but also to find users' trajectories.

Also in [TJMK18], authors achieved good results in human mobility and human trajectories using Wi-Fi probe requests. They have used a large data-set built on the probe frames collected from 54 public APs installed in Lower Manhattan in New York, NY during a whole week. In [ANL+18], the authors performed a very interesting work about Wi-Fi tracking using a low cost infrastructure. They have monitored a University Campus that received about 4,000 people per day, during a whole year. The outcome of their work is a set of considerations about limitations of this system, e.g., it is crucial to design very well the position of the sniffing stations. But the main contribution was provided by clustering methods to find characteristic behaviour of people around the Campus.

[1] `https://www.ttfestival.nl/`

Finally, Potorti et al. in [PCG+18] presented another way to take advantage of Wi-Fi. The authors obtained noteworthy outcomes in indoor environments, such as museums and shopping malls. Without performing a survey of the environment but simply by means of existing Wi-Fi network traffic analysis and by computing the position using a trilateration approach, they have created some user trajectory into a museum and a shopping mall with accurate results.

3.2.3 Crowd Density and Flow

Wi-Fi data could be used also for crowd behaviour monitoring, as showed in [PCB+17], where the authors figured out how to clean Wi-Fi data before the analysis in order to extract relevant information about the crowd. In particular, they have extracted data during an event which had involved 100,000 people, spread in three days.

In [GLRC19a, GLRC19b] the authors proposed a different approach, suggesting to analyze also the Bluetooth packets in order to improve the accuracy of people count estimation, achieving better results in the crowd mobility estimation.

In [XZL+14] it is shown that the use of probe request information can be utilized to count people in crowds. Their contribution is provided by a device-free Crowd Counting approach based on Channel State Information (CSI). They discuss the relationship between the number of moving people and the variation of wireless channel state.

Kurkcu et al. [KO17] figure out how to estimate pedestrian densities, waiting times, and flows using both Wi-Fi and Bluetooth sensors. Their algorithm is used to aggregate and clean data but also to fuse additional information in order to improve the accuracy of waiting time estimation. The method was applied to a dataset collected in a transit terminal situated in New York, for a period of two months.

3.3 Data Analysis

In this section it is explained how data analysis has been carried out on the data extracted from the "Probe Request" collected by the PMA stations, which are located in urban areas. Probe request frames are composed of several fields, but only a few of them are used to extract information. The most interesting field is the "source", because it contains the MAC address of the device that has sent the probe.

Figure 1.1 provides key details about the MAC address, which includes 6 bytes in length, uniquely assigned by the manufacturer to each network card. The first three octets, referred to as OUI (Organization Unique Identifier), are directly assigned by IEEE to the individual manufacturers of devices compatible with the Ethernet standard; the following three octets, referred to as NIC (Network Interface Controller), are assigned by the individual device manufacturer, to ensure the addresses uniqueness. Looking at the second least significant bit of the first octet of the MAC address (as shown in red in Figure 1.1), this could administered either universally

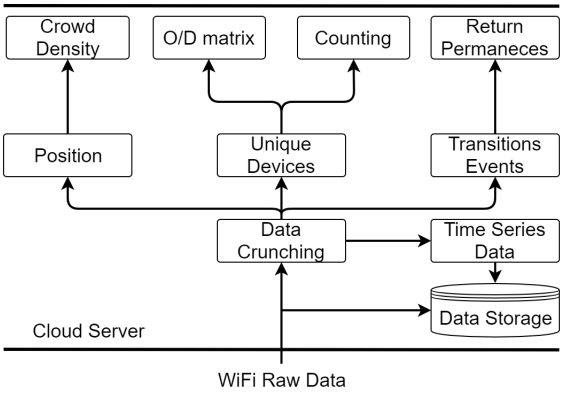

Figure 3.1: Data Flow

(setting it to zero) or locally by the end-devices (when it is set to one). A universally administered MAC address is globally unique; whereas this is not the case with a locally administered MAC address. This latter option is used to protect the users' privacy, for instance by periodically randomizing the MAC address, which allows setting fake MACs to make it more difficult to track devices.

User's tracking in relation to privacy has gained significant importance, to the point that the IEEE 802.11 working group has created a Topic Interest Group (TIG) on Randomized and Changing MAC addresses (RCM)[2]. This TIG is also focusing on the other issues of MAC addresses randomization, such as network analytics and troubleshooting, network performance, device manufacturer identification, MAC-based Billing and Access Control, and the need for a standard covering the whole randomization process.

One of the issues created by MAC randomization is that it makes it hard to perform necessary data analysis tasks, such as device counting and localization. However, in this chapter, I have therefore discarded all the probe request where the MAC address was locally administered.

Having clarified data collection, it is possible to get into the overall data processing flow, as shown in Figure 3.1, whereby each layer of the diagram adds value to the data. Firstly, the raw data is processed by the *Data Crunching* module, which is responsible for creating the time series (list of data points sorted in time) and saving them on the data storage. The time series are then processed by position, device, and transitions/events, to obtain the output metrics.

Let me now explain each module in greater detail. The localization module computes the coordinates for each MAC address, within specific time ranges. The unique device module analyzes all the MAC addresses received, returning a list of unique MAC addresses, which is then used by the others modules. The transitions events module computes all the presence transitions, allowing the Return Perma-

[2] `https://mentor.ieee.org/802.11/documents?is_dcn=DCN%2C%20Title%2C%20Author%20or%20Affiliation&is_group=0rcm`

nence module to compute the specific metrics. Thanks to this process is possible to identify all the unique devices seen during the whole monitoring process, allowing to obtain the count of people. Furthermore it is possible to know which users are stationary, which ones are returning to previous locations, and the duration of each event/transition (further details can be found in the following subsections).

For the sake of completeness, I summarize again the output metrics provided by the system which can be divided into two categories:

- Real-Time metrics:
  - *Counting*. This is the number of devices detected by each single station within a certain *counting time range*, which can be chosen among different values (e.g. last hour, last day, custom range);
  - *Position*. This is obtained via a trilateration algorithm based on the Friis' formula, which uses the received power of a signal and the transmission frequency.
- Post-processing metrics:
  - *Site returns*. This indicates after how long a device returns to the same place. It shows the number of devices that have come back after 5, 10, 30, 60, 120, 240, and 480 min, respectively;
  - *Site permanences*. This is similar to the return metric; it indicates for how long a device has been seen at a given place; the considered intervals are the same as the ones used by the return metric;
  - *O/D Matrix*. This shows how people have moved within a given PMA cluster; data is provided with a minimum interval of one day;
  - *Crowd density*. Starting from the single person localization within the monitored perimeter, people's density is shown using heat maps.

In the following subsections, I provide a more detailed description of the crow density improvements the advances obtained in paper [UCF+20c], which is the subject of this chapter, compared to what was done in paper [UCA19b] where it is presented the first version of the PMA solution.

3.3.1 Crowd Density improvements

The PMA platform can create heat maps directly correlated to people's density, within specific monitored areas. Although several papers have dealt with indoor/outdoor localization based on tracking via Wi-Fi [BBQL13, RRBP+14, HPL18], in this section the work is focused on RSSI-based and Time Difference of Arrival techniques. Selecting a time range in which to compute this information, PMA exploits some probe request packet fields to estimate the position of the devices, as further explained in the following two sub-sections that deal with device localization.

RSSI-based localization

The first proposed algorithm belongs to the family of range-based algorithms, which is based on the RSSI value contained within the probe request. Major efforts have been done by the scientific community to understand and improve these techniques [PMLC16, BBS17, YBC05], for which I refer to Stefan Knauth's work [Kna19].

In PMA, I use a range-based algorithm derived by the Friis' transmission formula [Fri46], using the frequencies of Wi-Fi communication, as reported in equation 3.1, where it is figured out how to calculate the distance d between a PMA station with known coordinates (anchor) and the target device (whose position is unknown). The use the following formula, assuming omnidirectional antennas:

$$d = \frac{\lambda}{4\pi\sqrt{\frac{P_{rx}}{P_{tx}}}} \tag{3.1}$$

where λ is the wavelength computed at 2,4 Ghz and P_{rx} and P_{tx} are the power of receiver (PMA Station) and transmitter (e.g. smartphone), respectively. Given a set of MAC address seen by the sniffing stations within a scanning window $\mathcal{S}_w$, I can define an MAC address group $\mathcal{M}_t$ seen by multiple stations during a time-slot t in the scanning window S_w. Each element of $\mathcal{M}_t$ is represented by:

$$m_{t,i} = \{mac_i, s_{lat,i}, s_{lon,i}, P_{rx,i,t}\} \tag{3.2}$$

where $s_{lat,i}$ and $s_{lon,i}$ are, respectively, the latitude and longitude of the station i that has "seen" the MAC address mac_i. Finally $P_{rx,i,t}$ is the power contained within the Wi-Fi probe request. From this information it is possible to compute a trilateration in order to derive an approximation of the Wi-Fi device's position by means of Algorithm 4

Algorithm 4 PMA Derive Positions algorithm

```
1: for all t ∈ S_w do
2:     for all mac ∈ M_t do
3:         itx_{t,mac} = Intersections(mac)
4:         if itx_{t,mac} = Null then
5:             Set P_{t,mac} with Friis's based positioning
6:         else
7:             Set P_{t,mac} = CoG(itx_{t,mac})
8:         end if
9:     end for
10: end for
```

The algorithm calculates the centre of gravity (CoG) of the polygon resulting from the intersections of the circumferences $itx_{t,mac}$. The radius of circumferences is

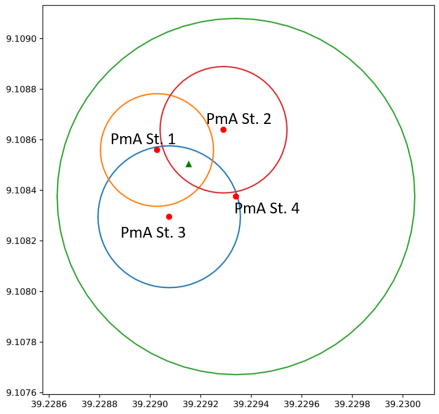

Figure 3.2: Target position estimated by RSSI-based algorithm.

equal to the distance d computed in 3.1, considering P_{rx} as the RSSI contained into the probe request packet and P_{tx} as the mean power for a common Wi-Fi antenna [3]. If the target device is seen only by one station, there is only one circumference, and it will not possible to find intersection points. In this case our algorithm chooses a random point that lies on the circumference with radius given by the Friis's formula, considering the power contained in the probe request.

Figure 3.2 shows the results of RSSI-based localization in the controlled scenario, whereby it is possible to identify the intersections of circumferences. Thanks to the position estimation obtained from the algorithm, it is possible to obtain the crowd density, by means of a heat map.

TDOA-based localization

In this section I explain how to use the Wi-Fi probe request frame to derive the devices position. The basic idea is to collect the Time of Arrival (ToA) of the probe request management frame from each station and, then, consider the packets having the same sequence number within a short scanning window (e.g. 1 second). Subsequently, I use this information to solve the following system of equations.
Let us define the target position with $P_t(x, y)$ and the anchor-stations positions as $P_a(x_a, y_a)$, $P_b(x_b, y_b)$ and $P_r(x_r, y_r)$. Furthermore, the time taken by the signal emitted by the target to reach the stations as:

$$\begin{cases} T_a = \frac{1}{c}(\sqrt{(x - x_a)^2 + (y - y_a)^2} \\ T_b = \frac{1}{c}(\sqrt{(x - x_b)^2 + (y - y_b)^2} \\ T_r = \frac{1}{c}(\sqrt{(x - x_c)^2 + (y - y_c)^2} \end{cases} \tag{3.3}$$

[3] `https://android.googlesource.com/platform/frameworks/base/+/master/core/res/res/xml/power_profile.xml`

Where c is the speed of light. Let us take P_r as reference anchor. Accordingly, I can define the differences between the previous arrival times, as:

$$\begin{cases} \tau_a = T_a - T_r = \frac{1}{c}(\sqrt{(x-x_a)^2 + (y-y_a)^2} - \sqrt{x^2+y^2}) \\ \tau_b = T_b - T_r = \frac{1}{c}(\sqrt{(x-x_b)^2 + (y-y_b)^2} - \sqrt{x^2+y^2}) \end{cases} \tag{3.4}$$

With no loss of generality, I can make the equations general and rewrite the system as:

$$\begin{cases} (x-x_r)^2 + (y-y_r)^2 = d_r^2 \\ (x-x_a)^2 + (y-y_a)^2 = (d_r + l_{a,r})^2 \\ (x-x_b)^2 + (y-y_b)^2 = (d_r + l_{b,r})^2 \\ \cdots \\ (x-x_n)^2 + (y-y_n)^2 = (d_r + l_{n,r})^2 \end{cases} \tag{3.5}$$

Where d_i is the distance between the target point P_t and the $i-th$ anchor point, $l_{i,r}$ is the TDOA range estimation. To improve readability let us substitute:

$$(x_i - x_r) = \overline{x_i} \tag{3.6}$$

and

$$(x - x_r) = \overline{x} \tag{3.7}$$

This form of equations 3.5 is quite hard to understand for a calculator and is rather inefficient. Therefore, in this implementation it has been used the *Least Squares Method* to simplify and solve the equations. Finally, from 3.5 subtracting the first one at the others equations and putting them in matrix form, it is possible to rewrite the system of equations as follow:

$$2\begin{bmatrix} \overline{x_a} & \overline{y_a} \\ \overline{x_b} & \overline{y_b} \\ \cdots & \cdots \\ \overline{x_n} & \overline{y_n} \end{bmatrix} \begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix} = \begin{bmatrix} \mu_a - l_{a,r}^2 \\ \mu_b - l_{b,r}^2 \\ \cdots \\ \mu_n - l_{n,r}^2 \end{bmatrix} + d_r \begin{bmatrix} -l_{a,r} \\ -l_{b,r} \\ \cdots \\ -l_n, r \end{bmatrix} \tag{3.8}$$

Where $\mu_i = \|P_i\|_2^2 = x_i^2 + y_i^2$. Given the following substitutions:

$$\underline{A} = \begin{bmatrix} \overline{x_a} & \overline{y_a} \\ \overline{x_b} & \overline{y_b} \\ \cdots & \cdots \\ \overline{x_n} & \overline{y_n} \end{bmatrix}, \underline{X} = \begin{bmatrix} \overline{x} \\ \overline{y} \end{bmatrix}, \underline{\Phi} = \begin{bmatrix} \mu_a - l_{a,r}^2 \\ \mu_b - l_{b,r}^2 \\ \cdots \\ \mu_n - l_{n,r}^2 \end{bmatrix}, \underline{\Lambda} = \begin{bmatrix} -l_{a,r} \\ -l_{b,r} \\ \cdots \\ -l_n, r \end{bmatrix} \tag{3.9}$$

The matrix equation can be described as:

$$2\underline{A}\,\underline{X} = \underline{\Phi} + d_r\underline{\Lambda} \tag{3.10}$$

The equation could be solved by means the *Least Squares Method* [CSMC04] and its solution is:

$$\underline{X} = \frac{1}{2}(\underline{A}^t\underline{A})^{-1}\underline{A}^t(\underline{\Phi} + d_r\underline{\Lambda}) \tag{3.11}$$

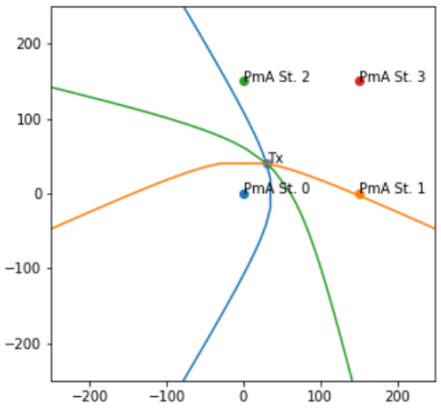

Figure 3.3: Target position estimated by TDOA-based algorithm.

The previous equation contains the parameter d_r and constitutes a non-linear expression. Therefore, solving the equation 3.11 leads to the final solution and to identifying the target.

In Figure 3.3 it is shown how the algorithm computes the solution of equations 3.11, using one of the four PMA stations as reference anchor. In particular, I did a simulation in a Cartesian plane, where all parameters (i.e. time of arrival and distances) had already been computed.

3.4 Experimental analysis

The experimental activity has been conducted in both a controlled scenario (to evaluate the performance of the positioning algorithm) and in real scenarios through different pilot studies (to evaluate the performance of the other metrics). In the following, it is firstly presented the setting and, then, analysed the results.

3.4.1 Experiments setup

Controlled scenario

To test and validate the localization algorithms, an outdoor empty open space has been selected, namely a non-utilized parking area of the Engineering Faculty of the University of Cagliari. This area has been selected due to the absence of obstacles and objects, which could have otherwise interfered with the stations. As shown in 3.5, the experiment has been performed using four PMA stations (black dots). In figure 3.6 I can see the stations used for testing in the controlled scenario. These were placed at the corners of a rectangle with a perimeter of 107 meters and an area of 710 square meters.

Once the stations were in place, I followed a path with an Android smartphone that was collecting the position in order to have a ground truth useful for perfor-

Figure 3.4: Ground truth trajectory (green) and trajectory (red) computed with RSSI-based algorithm.
Left: 40 seconds of time-aggregation; *Center:* 80 seconds of time-aggregation; *Right:* 120 seconds of time-aggregation.
The experiment was done in the Faculty of Engineering, University of Cagliari

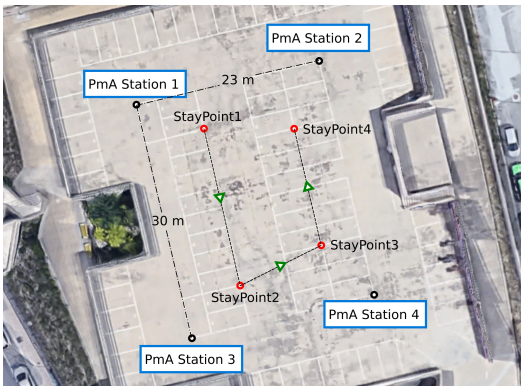

Figure 3.5: Map of the controlled scenario area.

Figure 3.6: PMA Stations used for tests in the controlled scenario.

mance analysis. In figure 3.5 I have marked in red the stay-points of our path, spending 3 minutes for each point, to be sure that the station would collect enough data for the experiment. The basic idea of the experiment was to try and understand the level of reliability of the two methods (RSSI-based and TDOA-based) applied to a pedestrian mobility scenario.

Pilot studies in controlled scenarios

In this section it is explained which pilot it has been done in some real-world situations, analyzing in the details what it was anticipated in the previous chapter. The data was acquired on three different scenarios: in the city center of Turin (Italy); during the International Truffle Festival in the city of Alba (Italy); and at the Engineering Faculty of the University of Cagliari (Italy). Each of these installations was created for a specific use case. The Turin center experiment was characterized by the following features:

- Objective: device counting near a roundabout, monitoring device site returns and site permanences, flow of vehicular traffic with O/D matrix.
- Installed devices: the devices were installed in a roundabout with 6 confluent arteries. A sniffing station was installed for each road, with a distance of about 20 meters from the entrance to the roundabout.

The Alba International Truffle Festival experiment was characterized by the following features:

- Objective: crowd density in the historic center;
- Installed devices: 5 stations were installed in as many points of interest identified in the historic center of Alba. Each station was installed near the road to facilitate data acquisition. The purpose of this installation was to understand how many people visited the points of interest during the truffle fair.

The Engineering Faculty experiment was characterized by the following features:

Table 3.2 Error evaluation in meters.

	Aggregation [seconds]				
	40	**60**	**80**	**100**	**120**
Point 1	4.5	4.5	4.5	4.5	4.5
Point 2	5.1	5.1	5.1	5.1	5.1
Point 3	7.0	7.0	7.3	7.0	7.7
Point 4	10.3	8.3	8.1	8.3	7.2
RMSE	14.2	12.8	12.9	12.8	12.5

- Objective: people counting, crowd density.
- Installed devices: 8 sniffers have been installed, covering the area of the park at the center of the engineering faculty, to monitor overcrowding during the day and count people.

3.4.2 Experimental results

The experiments were conducted following the setting explained in section 3.4.1 and the results are summarized in Table 3.2. After data collection, in-depth data processing was performed.

Initially, all the probe requests captured during a specific time interval are grouped based on the aggregation parameter. This parameter has the function to increase or decrease the time interval centered at each moment in which the target remained stationary in the stay points (red marker in Figure 3.5). Then all the probe requests inside a time interval are taken into account for the estimation of the position. To compare localization errors at the different stay points, I used the Root Mean Square Error (RMSE) as gauge.

Evaluating the results found, I can see that in Point 1 and Point 2, the error is fixed for each aggregation interval. This behaviour is due the usage of median to find out the average power from the probe received. The outliers points have less weight so the median value is stable across the different aggregations. A different situation appears for Point 3 and Point 4, where the median error changes somewhat more unpredictably. This is due to scattering and multi-path effects that are most probably responsible for interference on the signal propagation.

In general, it could be noticed that, by increasing the time observation interval, RMSE decreases. Obviously this is true only if the target is staying still at the same point for an amount of time comparable to the window time interval.

At first, the TDOA algorithm was performed through simulation, leading to very promising results. I then performed the same experiment in a pilot study, employing the RSSI algorithm. Unfortunately, the experiments have shown that by means the PMA stations equipped with this low-cost hardware, is not possible to obtain

satisfactory results. As a matter of fact, the weakness in this type of approach is represented by the time measurements accuracy and precision, which requires hardware with extremely precise time resolutions. Thus, it is crucial to employ higher-spec hardware as anchor device, and pursue strong time synchronization among the different anchors, as preconditions for an effective TDOA-based algorithm in real-world scenario with short distances among anchors.

I has been also identified further problems, in relation to the event chosen as the trigger in the time-counting systems or the latency between the different process layers (eg coding, synchronization, etc.). Moreover multi-path and NLOS problems may occur between transmitter and receiver. Finally, since the PMA stations are based on Raspberry Pi and Linux OS, the OS process scheduling policy could not be a precise time-acquisition process without using an implantation of Precision Time Protocol of the local area network among the anchors.

In fact, without extremely accurate synchronization at the moment it is not possible to reach an accuracy in the nanosecond range, which is the key limitation pinpointed in our real-world settings. One possible solution is to implement the GPS or the Precision Time Protocol IEEE 1588 [4]. The adoption of the IEEE 1588 Protocol would indeed allow us to fix this important issue in our TDOA algorithm. I verified that, at the moment, a raspberry version of linux-ptp exists, which, however, is currently incompatible with the most recent versions of Raspbian OS, necessary to run our scripts to perform the other operations, such as probe requests collection and processing. Another obstacle for a rapid implementation of this solution is that the IEEE-1588 protocol was designed to work on LAN networks, and not on WLAN networks, as it is our case. However, this issue could be fixed by following the solution proposed in [CSZ+15], which needs further investigation.

3.5 Conclusions and Future Works

The aim of this work was to design and develop a low-cost device to monitor and analyse the mobility of people in urban areas through a passive strategy. This is the reason why several metrics has been adopted several metrics, relating to crowd density, the people flow moving between different areas. The solution has been tested first under controlled conditions and then in three pilot deployments. Furthermore it has been tested two different people localization approaches, using RSSI-based and TDOA-based algorithms, respectively.

I found that the RSSI-based approach is better suited to the resource constraints of our method. Whereas, the TDOA-based method would require a higher-spec system providing the necessary level of time resolution. One possibility would be to employ RTOS (Realtime Operating Systems) with FPGA (Field Programmable Gate Array) or the IEEE 1588 Precision Time Protocol, in order to achieve sufficient time-synchronization between the devices.

[4] `https://www.nist.gov/el/intelligent-systems-division-73500/ieee-1588`

Chapter 4

MAC Address de-randomization: Wi-Fi Probe Request still be a source of information

To improve city services, local administrators need to have a deep understanding of how the citizens explore the city, use the relevant services, interact and move. This is a challenging task, which has triggered extensive research in the last decade, with major solutions that rely on analysing traces of network traffic generated by citizens WiFi devices. One major approach relies on catching the probe requests sent by devices during WiFi active scanning, which allows for counting the number of people in a given area and to analyse the permanence and return times. As showed in the previous chapter, this approach has been a solid solution until some manufacturer introduced the *MAC address randomization* process to improve the user's privacy, even if in some circumstances this seems to deteriorate network performance as well as the user experience. In this chapter, I will focus on how artificial intelligence can enable techniques to tackle the limitations introduced by the randomization procedures and that allows for extracting data useful for smart cities development. The proposed algorithm extracts the most relevant *information elements* within probe requests and apply *clustering algorithms* (such as DBSCAN and OPTICS) to discover the exact number of devices which are generating probe requests. Experimental results showed encouraging results with an accuracy of 65.2% and 91.3% using the DBSCAN and the OPTICS algorithms, respectively.
My contribute in this work, which is inspired to the paper [UCF+20b], was to graft the idea of using the length of the information content of the information elements to create a vector space on which to project the probe requests, and subsequently, to use as characteristics for cluster them using unsupervised clustering algorithms such as DBSCAN or OPTICS. Furthermore, I contributed in providing a solution for the reduction of the space of the features by introducing the concept of measuring their variability through a coefficient.

4.1 Introduction

In the last decade, understanding how people move around the city is becoming increasingly important for the local administrators for a better design of smart cities services. An approach that is often followed in this context is based on the analysis of traffic generated by our personal mobile device. For example, in the UK, the government started an experiment where some smartphone-monitoring bins can track people through the WiFi interfacing, thus obtaining key information on people's behaviour and adjusting the service accordingly [Goo13]. Another solution has been implemented by Cloud4Wi, which tracks people when moving around in shops and malls and are then able to provide various information about the areas of greatest interest among the shoppers [PCG+16b]. In [GC18], a system for Smart Cities scenarios is presented; on the basis of the WiFi signals, the authors have demonstrated to be able to distinguish walking pedestrians from those waiting in the sidewalks in the proximity of a pedestrian crossing. They are also able to estimate the exact position for people that are waiting to cross the street. All the previous solutions are based on the analysis of Wi-Fi Probe Requests analysis, which are configuration frames in the Wi-Fi communication setup. It is clear that the Probe Requests contain an a lot of information; an example is the Preferred Network List (PNL), i.e. the SSID of the Access Points known by the device and its MAC address, even if it is an information less and less used [DP9]. In spite of the lack of information related to PNL, there are papers that show how the Probe Requests information can be used to create traces of mobility in order to estimate density and flows within cities. In this context, a good example is provided by a feature of the system People Mobility Analytics built in [UCF+20c], a real-world system for people monitoring based on WiFi probes. The information used to obtain this data is considered, according to the GDPR, as personal identification information (PII) since it can be used to identify a specific person's movements. This appear to be the reason why some manufacturer started to implement MAC address randomization. In the next subsection, it is provided a brief description of the Probe Request frame and the Information Elements (IEs) which crucial to send important information from device to the Access Points, for instance the client hardware capabilities. In order to test the de-randomization algorithm it has been collected 83127 Probe Request packets in 3 different scenario as figured out in tab. 4.2 . Exploring better the dataset it can be noticed which a subset of them is composed of packets with randomly generated MAC address as explained in section 4.3.1. Based on a particular data cleaning pipeline, it is possible to extract from probe request some good features which allow to count and track people, simply exploiting WiFi standard weaknesses. The major contributions of this chapter are the followings: the design and implementation of a pipeline to WiFi probes data cleaning and analysis; the implementation of a clustering model in order to find the real number of devices which are generating Probe Requests in a not crowded environment; the definition of a system to dynamically extract features in order to

create a device signature, which is the novelty with respect to the state of the art.

4.2 Proposed Algorithm for MAC address derandomization

The proposed algorithm is based on the analysis of the probe requests sent by the WiFi devices, with particular attention to the information elements (IEs) and the lengths of the information (LENs) which are taken in account as algorithm's inputs. Some, IEs within the probe request are not mandatory but they are sent because they are necessary to explain which functionalities are supported by the device itself [VMC+16]. This work drew inspiration from the paper [UCF+20b] and where I noticed through experiments which each device could send only mandatory IEs or also some optional. These differences seem to depend on the choices of the manufacturer of the wireless network card and the logic of implementation of the operating system. Therefore, the experiments have shown that the IEs are generated in an pseudo-univocal way for each device, with little variations in time. Variations may still happen according to the context in which the devices are located. In order to clarify these concepts, Fig. 4.1 shows how the length of information elements changes over time. Taking into account those inputs was already done in other works [MCRV16, VMC+16], however they do not manged the IEs in a dynamic way so they may not consider the reserved Information Element IDs which are generally manufacturer-dependent. Instead, the proposed algorithm works in adaptive way, so it can recognise which Information Elements ID are most frequent and with "enough variability", this concept will be more clear in the following. However, in order to obtain the right output from the algorithm, the the raw data must be conditioned. In the following subsection, it is firstly described the procedures which has to been performed in order to clear and pre-process the data to make it ready for the algorithm; secondly, it is presented the proposed clustering based on the Information Element IDs and Lengths to estimate the real number of devices generating all the data collected. This pipeline flow is shown in Fig. 4.2 and detailed description is provided in 4.2.1.

4.2.1 Data cleaning and preparation

The data cleaning and preparation is the most important step of all Data Analysis processes, therefore this step is done as first. Indeed, the collected raw data still needs a preliminary checks for errors, deletion of unnecessary information, identification and eventually imputation of missing values. Initially, all the corrupted packets are discarded to avoid the introduction of errors in the following operations. Then, the first important filter is applied to divide the flow of packets into those that contain random MAC addresses and the remaining ones. This filter returns two data subsets: one contains all the packets sent by devices that are sending they real

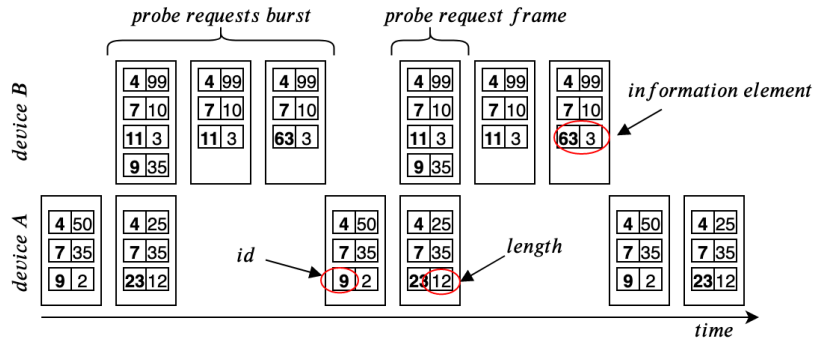

Figure 4.1: Probe requests burst and their information elements sent by two devices over time.

Table 4.1 Most frequent Element IDs

Element ID	# Packets	Meaning
50	1436	Extended Support Rates and BSS Membership
45	933	HT Capabilities
127	723	Extended Capabilities
3	590	DSSS Parameter Set
191	89	VHT Capabilities
238	24	Reserved
128	15	Reserved

MAC address and the one that is composed by all the packets sent by devices that are implementing the MAC address randomization. As explained in section 1.0.1, by checking the second least significant bit in the first octet of the MAC address is possible to recognize the randomized MAC address. The challenge is now to extract the number of devices that are generating all the MAC address of the packets belonging to the second subset. To obtain the needed insights and perform the clustering operations, it is necessary to extract and to order the mentioned features, i.e., the IE IDs and the LENs values for each packet in the random MAC addresses subset. Each packet is then deeply examined, by extracting the MAC, time, IDs and LENs, generating a Data Frame. The resulting table shows a first overview on the data regarding all the packets sent by the random MAC addresses, but it is necessary to have a better view to continue the analysis. The resulting sparse matrix is then converted into a dense matrix: here for each MAC address, the values of the LENs for each IDs are grouped. This view is necessary to have a kind of signature for each MAC address and this could be useful to cluster the different addresses that come from the same device because the base theory is that sign is the same or similar.

$$E_{ies} = e_{i,j} \in \mathbb{R}^{m \times n} \tag{4.1}$$

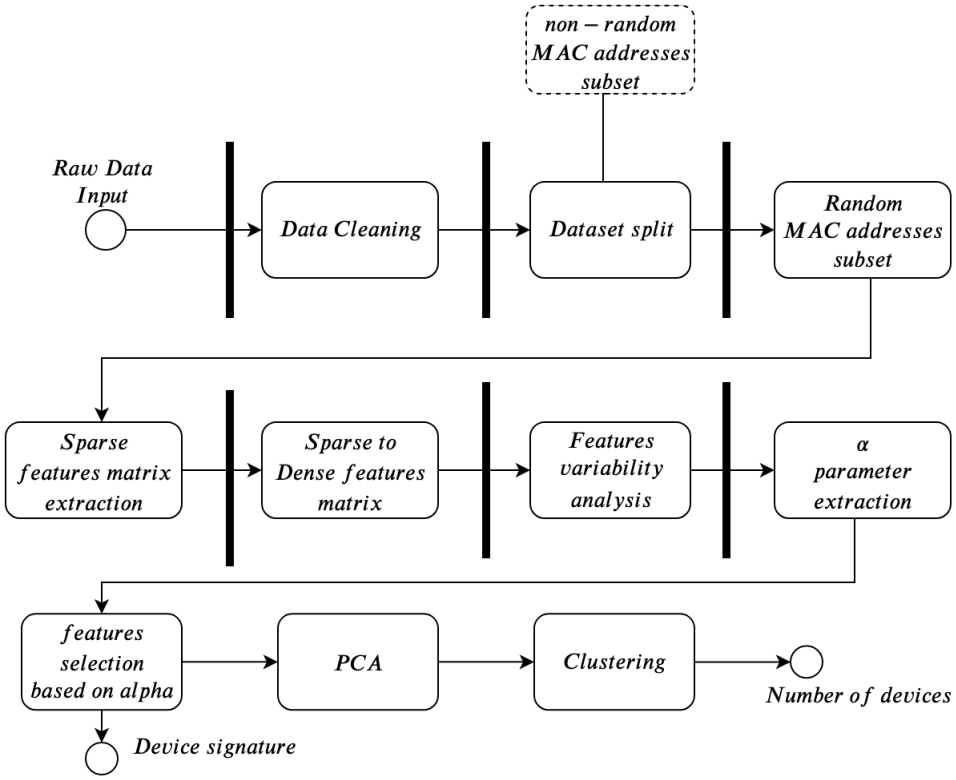

Figure 4.2: Data Analysis pipeline

Where m is the number of different MAC addresses present in the whole dataset and n is the number of different IDs founded. The matrix that come from this step, it is huge enough to create confusion and not all the features are obviously significant to discriminate the different devices. For this reason, the next step is to choose the features to take in account. A previous analysis has been done, it concerns the features variability, in detail all the columns of the 5.1 are analyzed and the unique values for each ID are counted.

Let is defined IDS as the vector of Information Elements IDs founded in the previous step:

$$IDS \in \mathbb{R}^n \tag{4.2}$$

In order to reduce the dimension of the vector IDS, it has been introduced the threshold parameter α. Its value represents the minimum unique values that the column j of matrix E_{ies} needs to have to be taken in account. Changing α it is possible to change also the dimension of IDS vector simply applying this rule:

$$IDS = \{unique(E_{:,j}) > \alpha\} \ \ \forall j \in E_{ies} \tag{4.3}$$

After this filter is applied to E_{ies} matrix, as result of the filter, the IDS vector will contains only the columns of E_{ies} which have at least α different values. The best α

is computed by an iterative method where 1 is considered as starting value and the stop condition is set to:

$$dim(IDS_k) - dim(IDS_{k+1}) \leq 2 \tag{4.4}$$

Where k is the current value of α. To make it more concrete, Fig. 4.3 shows the characteristic behaviour of the Information Element IDS dimension, which it is considered as features, on changing the threshold α. To check the validity of the solution it is useful to compare the IEs resulting by the previus experiment with the IEs identified by Vanhoef et al.[VMC+16] Consider that some differences due to the time of the study are present; an example is the preferred network (PNL) list that is an information that is less and less available in probe requests, in fact the SSID field is now almost always empty[DP9]. However, if IDS_k dimension is less than the group of IEs mentioned above, the algorithm takes in account the IEs identified in tab. 4.1.

In all the tests which has been done the number of features IDS_k reach the stop condition, written in the inequality 4.4, using a α value between 5 and 10, therefore in this way it is possible reduce the features number to the first value below 15. This reduction, allow to continue the procedure with an higher knowledge on the behaviour of the different devices. Indeed reducing the features number in this way and taking a look at the remaining, it appears correct and intuitive because they have a physical meaning. Table 4.1 shown the most frequent IDs present in the whole dataset, the related information are specific from the device that is sending the probe. In the end, all the data prepared and conditioned are passed to the clustering algorithms.

4.2.2 Density-based Clustering Modelling

There are several types of clustering algorithms in literature, density-based approaches rely on the amount of points which are within a predefined radius in the features space. They have the advantage of being able to create arbitrary clusters and if properly configured they enjoy good scalability. Among the density-based algorithms it can be found, there is a well-known algorithm that is called Density Based Spatial Clustering of Application with Noise (DBSCAN) [EKS+96]. However, there are also other alternatives such as Fuzzy Joint Points (FJP) [Nas06], and Noise-Robust Fuzzy Joint Points (NRFJP), finally the successor of DBSCAN is mentioned, also known as OPTICS: Ordering Points To Identify the Clustering Structure [ABKS99]. Density-based clusters are defined in the features space as variable density areas separated each other by more rarefied areas. The idea could be explained introducing the definition of *core-points*, *density-reachable points*, *density-connected* and *outliers* or *noise*. In order to define a core point, its neighbourhood of radius ε has to contain at least $MinPts$ points, i.e. we can say which the points density in the neighbourhood of p has to cross a threshold so that a point p can be defined a

core-point. Furthermore, a point u is defined a *directly-density-reachable point* if it is in the neighbourhood of p. However, a point could be only *density-reachable* if there is a transitive closure of direct density-reachability. Finally, outliers are defined as the set of points in the dataset which not belonging to any cluster.

Once the previous definitions are clarified, the concept of cluster in both DBSCAN and OPTICS algorithms could be introduced highlighting the main differences, because they are the algorithm that has been subject if this study.

DBSCAN: once ε and $MinPts$ are given the clusters' density is defined and is not possible to change it during the clustering process.

OPTICS: it is based on the principles of DBSCAN and follows all its definitions. However, as first step the patterns are ordered such that spatially closest points become neighbours in the final ordering, subsequently the additional definitions are applied in order to find clusters with different densities [ABKS99], for simplicity in the following it is defined the parameter χ_i to identify the determines the minimum value of steepness on the OPTICS reachability plot which allows the algorithm to constitutes a cluster boundary.

4.3 Results

4.3.1 Data collection and dataset characterisation

In this section it is provided an overview of the datasets and how the collections have been done. To test and validate the proposed algorithms was necessary to collect sufficient data to have different brands and operating systems. The very first acquisitions have been done using a laptop with Ubuntu OS, Wireshark and an external WiFi antenna set in monitor mode.
The data acquisition has been done in three different kind of scenarios, obtaining data saved in three different dataset.

Laboratory scenario: the dataset has been done inside a semianechoic chamber considering 23 devices;

Controlled scenario: were acquired during a workshop at university of Cagliari, into a room with 30 people with smartphone turned on and without notebook or smartwatch;

Real-world scenario: were acquired inside the university campus where the number of present people was counted by human and was 45. However, in this situation is not possible to take into account if all the people whom was present had only one WiFi device turned on.

The collections has a mean duration of 30 minutes. In the laboratory dataset it is gathered data from 23 devices as shown in tab. 5.5.

Table 4.2 Details of data-sets

Scenario	Probe Pkts	Virtual MAC add.es	Real MAC add.es
Laboratory	15151	181	8
Controlled	23665	179	28
Real-world	44311	1508	162

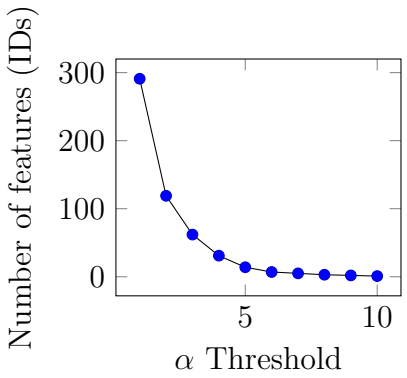

Figure 4.3: Number of total features taken into account changing the threshold α

The controlled scenario dataset, in the first analysis provided 28 real MAC addresses, however only 12 of them have been taken into account and the other 16 have been eliminated because of they was only noise. In fact, the discarded MAC addresses had a received signal strength indicator (RSSI) lower than -54 dBm. That value has been chosen as threshold based on the modal value of RSSI distribution. It highlights how those MAC addresses were associated to device outside the area of interest. Actually this is the very first data filter in the Data Cleaning module.
A similar cleaning approach has been done to the real-world dataset. In this scenario is also possible to collect data provided by people whom were walking or driving outside the acquisition area. For this reason it is fundamental to make a good data cleaning in order to avoid over-counting situation due to the near road where some probe request could be collected. Even if is not possible to have the mobile details for each scenario presented, the acquired data has been really useful to check and compare the IEs behaviour inside the probe request.

4.3.2 Clustering results

As explained in section 4.2.1 after the splitting in two subset, the algorithm takes in account only the packets where the address was generated randomly. After the transformation from sparse to dense matrix, the algorithm evaluates the feature variability (shown in fig.4.3) and select the α value to obtain less than 10 features to take in account. After that, the resulting values are sent to a PCA algorithm to reduce the feature space to three dimensions and then the features were sent to

Table 4.3 Devices under study

N°	BRAND	MODEL	OS
1	Huawei	Tag-L01	Android 5.1
2	Samsung	J5	Android 7
3	Huawei	P10 Lite	Android 8
4	Huawei	Mate 10 Lite	Android 8
5	ZTE	Axon 7	Android 8
6	Samsung	J3	Android 9
7	Samsung	Note 8	Android 9
8	Samsung	Galaxy A50	Android 9
9	Xiaomi	Mi 9T	Android 9
10	Xiaomi	Redmi Note 7	Android 9
11	Honor	9 Lite	Android 9.1
12	Apple	Iphone 6	iOS 12.4.4
13	Apple	Iphone 6S	iOS 13.3
14	Apple	Iphone X	iOS 13.3
15	Xiaomi	Mi 2Lite	Android 9
16	Motorola	Moto G (2014)	Android 7
17	Xiaomi	Redmi 4 Pro	Android 6.01
18	Huawei	P20 Lite	Android 9
19	Apple	Iphone XR	iOS 13.3
20	Apple	Iphone 7	iOS 13.3
21	Samsung	S6 Edge	Android 7
22	Samsung	S4	Android 4.2.2
23	Huawei	P7 Lite (2015)	Android 6

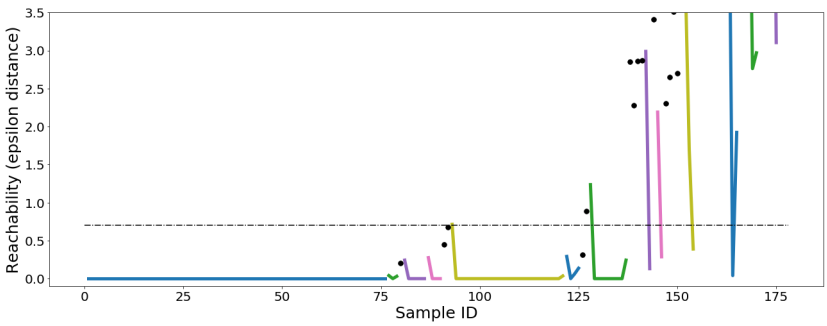

Figure 4.4: Reachability plot

the clustering part of the algorithm. Following the clustering theory to obtain the most affordable number of cluster, it is primary to choose the correct χ_i parameter. The proposed algorithm iterates χ_i between 0 and 1 with a step of 0.01 calculating the resulting number of clusters. Let is define $N(\chi_i)$ as the number of cluster given χ_i, the algorithm chooses the max value of frequency in the histogram of $N(\chi_i)$ distribution, in other words choose the modal value. Once χ_i is computed the output of OPTICS clustering is 13 as shown in fig. 4.4, where the line with different colours represents the clusters. In the same figure it is represented also the threshold ε used for DBSCAN, obtained with the same procedure used for χ_i and which value is 0.7; with these conditions the algorithm returns 8 clusters. To evaluate the accuracy of the algorithm developed using the two different clustering methodologies (DBSCAN and OPTICS) it is useful to calculate the number of devices identified by the algorithm over the real number of devices present in the testing chamber. Using DBSCAN the recognized devices are 7 that added to the same 8 non random devices returns a total of 15 over 23 real devices, obtaining an accuracy of 65.2%. Instead, using OPTICS the recognized devices are 13 that added to 8 non random devices returns a total of 21 over 23 real devices, obtaining an accuracy of 91.3%.

4.4 Conclusion

This chapter have addressed the fundamental topic of MAC address randomization weakness due to the IEEE 802.11 standard vulnerabilities. It is designed, implemented ed evaluated an adaptive algorithm which creates a signature based on the Information Elements (IEs) contained in WiFi Probe Requests collected in not crowded environment. The signature is a powerful tool to count the real number of devices which are present near the data collection station, furthermore it is possible to use the same signature to track a device among different station in order to extract mobility pattern in smart cities context. In my vision, the MAC address randomization is still problematic because is not enough in order to completely protect users' privacy in very particular situations. It also a problem for networks efficiency by worsening the user experience of non-AP WiFi stations, as highlighted by IEEE

which created the study group on Random and Changing MAC addresses (RCM) [1] in order to modify the standard.

[1] https://mentor.ieee.org/802.11/documents?is_dcn=DCN%2C%20Title%2C%20Author%20or%20Affiliation&is_group=0rcm

Chapter 5

Can Artificial Intelligence turn digital junk into insights?

To answer the question in the title, we need to think that to preserve people privacy and prevent device (and people) tracking, WiFi MAC address randomization is been introduced by an ever increasing number of operating systems. Accordingly, mobile devices make use of different *virtual* addresses over time so that not a single fixed *factory* address is used that may identify a specific user. This has the consequence that it is not even possible to extract anonymous information on people mobility by analyzing WiFi traffic traces, which would be useful for many purposes (e.g., counting the number of people in a mass transport vehicle).

To address this issue, in this chapter it is presented a novel MAC address de-randomization algorithm which groups the Probe Requests generated by the same physical device. With respect to past works, it is considered a combination of the features that have been previously considered in isolation, which are associated to the content and length of the optional fields conveyed in the sent frames and the rate at which the frame are numbered over time. These features are then used by density-based clustering algorithms (i.e., DBSCAN, OPTICS, HDBSCAN) to group frames sent by the same device. Additionally, it is also considered the presence of pseudo-random MAC addresses, which are those that do not change every frame but only when the emitting device switch on and off the Wi-Fi interface. To this I have developed an heuristic to detect these sequences of frames so as to improve the algorithm efficacy. Experiments have been initially performed in a controlled environment where I reached an accuracy close to 96%. Then, experiments in a real scenario have been conducted where the people taking the bus when moving in an urban area have been counted; in such a scenario an average accuracy of of 75% has been obtained.

My personal contribute in this work was to formal describe a "pseudo-virtual" behaviour for devices, and to introduce the idea of burst rate as an angular coefficient of the linear regression of the points identified by the sequence number and by the arrival time of the probe requests.

5.1 Introduction

In the last decades, the randomization of the MAC address by devices for human use (such as smartphones and tablets) during the AP research phase has become increasingly widespread and pervasive. As a result, these devices use a virtual MAC address that changes continuously over time. Therefore, algorithms that intend to extract anonymous information (e.g. by counting the number of devices in a certain area) can no longer be effective as they are designed to receive a globally unique address as input. Other consequences have been introduced, and the IETF and IEEE 802 standardization committees are evaluating the impact of these MAC address randomization processes on existing use cases for network and application services. Indeed, the MAC address was designed to be static and many services rely on this logic, but with randomization those services may no longer function correctly. For example, think of all those authentication procedures based on MAC address or handover to keep the device authenticated when moving from one Access Point (AP) to another in an extended network.

The first appearance of the randomization process dates back to 2014, when Apple, in order to protect the privacy of its customers, introduces the random variation of the MAC address within the probe requests. Since then, several *de-randomization* algorithms have been proposed, whose objective is to cluster frames generated by the same device and observed in given period of time. It is natural to understand how the main proposals fall within the area of passive sniffing, as by collecting and analyzing the frames generated by client devices when searching for networks to connect to, you can have a complete and clear picture of how the devices "behave" on the Wi-Fi spectrum. The features that are considered are mostly related to: the content of some optional fields that are conveyed in these frames, which vary from a device to another (e.g., [FMT+06a], [VMC+16]; the temporal distribution of the sent frames (e.g., [MCRV16]); and the inter-frame time (e.g., [NPP+20]).

The proposed algorithms are able to correctly group frames that use different MAC addresses but belonging to the same device up to 75 % of the cases (best result). Although these results may be satisfactory for some application domains, it should be noted that randomization techniques are evolving, also involving the starting point of sequence numbers and the GAP that is used between one probe request burst and another. Furthermore, many of the above techniques are based on the order of arrival of the eprobe request and therefore basically on the sequence number and time of arrival of the frame. Most of the time using a recursive approach, with high computational and time cost, leaving no room for real or industrial applications.

Based on the previous considerations, in this chapter it is described a novel passive sniffing de-randomization algorithm. The provided contributions are the following:

- it is considered a combination of the features that have been used in isolation in past works and that are associated to the content and length of the

optional fields conveyed in the sent frames and the rate at which the frames are numbered over time. These features are then exploited by a density-based clustering algorithms (i.e., DBSCAN, OPTICS, HDBSCAN) to group frames sent by the same device.

- it is considered the presence of pseudo-random MAC addresses, which are changed by the emitting device only it switches off and on the Wi-Fi interface. I have developed an heuristic to detect these sequences of frames (with almost static virtual MAC addresses) so as to improve the overal de-randomizzation process efficacy.

- the algorithm has been tested in a controlled environment, i.e., inside an anechoic chamber, so that the ground truth data was available. In this scenario, I reached an accuracy close to 96%, which is far higher the performance achieved by previous works.

- an algorithm for counting the number of people inside a mass transport vehicles has been defined, which relies on filtering the frames on the basis of the received signal power. In such a scenario an average accuracy of 75% has been obtained. Whereas, these results seem to be in line with previous works, it is worth mentioning that the increasing number and complexity of the randomization algorithms adopted by the continuously evolving smartphone operating systems make these results a significant outcome. Additionally, this scenario suffers from the intrinsic error introduce by the fact that some people in the vehicle may either not have a device with a WiFi active interface or have more than one.

The rest of the chapter is organized as follows. Section 5.2 provides the background information and briefly review the past works. Section 5.4 describes the features that are used in the proposed de-randomization algorithm, the clustering algorithm and the detection of the pseudo-random MAC. Section 5.5 presents the results obtained in a controlled environment. Section 5.6 presents the algorithm that has been developed to count the people on-board of a mass transport vehicle. Section 5.7 provides final conclusions.

5.2 Related past works

In the past, a lot of work has been done on Wi-Fi analytics that manage the randomization process in order to achieve different goals. The canonical case that anyone who approaches passive Wi-Fi analytics faces is the count of devices in a given area. After randomization, the most reliable way to perform quality Wi-Fi analytics is to give up the totality of the representative sample and use an active approach, which requires the user to connect to the Wi-Fi network. Below, the best known works in

the state of the art are analyzed and the innovativeness of the proposed algorithm is highlighted.

Passive sniffing methods

In [MCRV16], C. Matte et al. proposed a method to address the randomization process by means of an algorithm which takes as input a set of Probe Request frames and associate them to the different devices. The principle is to create a signature based on the inter-frame time, the inter-burst time and on the frequency of frame sending; a measure of similarity between signatures based on the Franklin [FMT+06a] distance is then defined. Finally, the MAC addresses whose similarity distance is below a certain threshold are aggregated. The resulting algorithm is recursive as at the end of each iteration it provides a list of groups with are again used to evaluate the distance of each frame with respect to the features of the different groups to eventually obtain a better association.

A major weakness of this approach is that, due to some electromagnetic phenomena such as scattering or multi-path, the inter-frame times vary significantly between one burst of frames and the next, thus leading to multiple signatures for the same device. From the performed experiments, the resulting accuracy reaches 75% in a controlled scenario.

In the work presented by M. Nitti et al. in [NPP+20], a solution is proposed to count the number of passengers present in public transport vehicles. To identify whether two Probe Request frames have been issued by the same device, a score is computed which depends on the difference of the time of arrival and the difference of the sequence numbers. These differences are computed for all the possible couples of MAC addresses which have the same Information Element IDs, regardless of the length or content associated with it. Accordingly, the resulting algorithm assumes a recursive form and because of that is very computational intensive and very difficult to be used in a real scenario. The authors claim an accuracy of 100% in a controlled environment (closed room) and of 94% in dynamic environment (a car). However, the tests have been performed with a limited number of devices (5 devices) and in a partially simulated environment. Also, in [FMT+06b] J. Franklin et al. proposed a timing-based approach; also in this case, the resulting solution is affected by errors due to uncontrollable physical phenomena, such as scattering and multipath.

Another work that relies on Probe Requests fingerprinting is [VMC+16]. Herein, Vanhoef et al. have proposed a device tracking algorithm based on IE IDs fingerprinting. Their approach follows two phases. In the first one the IEs IDs are used to group Probe Requests into clusters, regardless of their temporal order and their content. In the second phase, the algorithm tries to distinguish devices which are in the same cluster as they share the same IEs group. To do that, the algorithm relies on the predictable behavior of the sequence number. To this, it assumes two probe requests belong to the same device if the difference in arrival times is lower than 500 seconds and their sequence number difference is less than 64. However, the

probability of a device being successfully identified is less than 30% if the devices are more than 16. A corollary of this work is that to pre-grouping probe requests based on their IEs is a good clue to find the MAC address pool which a device has changed over time and may reduce the computational cost of the previous approach, which was implemented in a recursive form, but paying an heavy price in terms of accuracy.

In [SWS+20], N. Suraweera et al. used WiFi packet sniffing to collect device-related compressed beamforming reports (CBRs). Indeed, downlink beamforming is facilitated by transmitting CBR from each wireless device to its AP. They exploited this information using the discrete 2-D Fourier transform (2D DFT) for feature extraction, similar to what is done for image matrices. The system was tested in an different environment from the training one and with devices were not present during the training. The results obtained indicate 100% accuracy with no device, 97.8% with one device, 78.3% with two devices, and 93.9% with three devices present in the environment.

In [RNNS20], M. Ribeiro et al. present a counting and tracking method based on automatic classification techniques. This very well done work was based on a 4-year probe request acquisition period. The collected data were input to 7 unsupervised classification algorithms; finally the average accuracy was calculated by comparing the data collected by the authorities that administer the stadium, port and airport as ground truth. Thanks to the long observation period, the authors provided an overview of the rate of increase of devices randomizing the MAC address. In particular, at the beginning of the acquisition the devices that used the factory MAC addresses were just over 50%, in the last period they are just under 5-10%.

The authors of [RNNS20] applied the same method in [RGPN20] to create mobility tracks by monitoring passengers using public transport. Origin-destination matrices were created that provided an overview of which bus stops were most used by passengers. The analysis in this case was done without taking into account the random MAC addresses. Power filters were also applied to make it possible to understand which probes not to take into account for the analysis. The information extracted compared with the ground truth (number of tickets sold for the route) decreed that the proposed solution is not adequate to estimate the entrances and, consequently, the exits. However, it emerged that such information makes it possible to visualize and detect unusual situations and to raise awareness, support and facilitate communication between the different needs of stakeholders.

Active Sniffing Methods

The methods that belong to this category implement attacks that manipulate packets and fool devices on the network. In [VMC+16], Vanhoef et al. have analyzed over 8 million probe requests with active techniques, counting around 170000 MAC addresses. The best-performing approach takes advantage of WiFi Protected Setup (WPS) parameters for devices that support it. The device being connected provides

a parameter called Universally Unique Identifier-Enrollee (UUID-E). They found that this parameter is directly linked to the device's factory MAC address. This allowed the authors to trace the real MAC by greatly reducing the device count error.

Martin et al. [MMD+17] analyze various techniques that can be used on a large scale to be able to trace random MAC addresses to a single device. In particular, active sniffing methods exploit various vulnerability and made attacks such as KARMA attack [VCS03] [Kum20] (creation of fake Access Point from the list of probe BSSIDs) and RTS/CTS attack [KJ20] in order to obtain the true MAC address during the negotiation of the connection with an AP, this approach requires the device's known SSID as knowledge of the attacker.

5.3 The data acquisition device

The data acquisition focuses on capturing the Probe Request frames emitted by the Wi-Fi clients in a given area. It is performed by a single board computer configured as Wi-Fi packet *sniffer*; it is shown in Figure 5.1 and relies on the following hardware:

- 1 Raspberry Pi 4 with a custom firmware based on Raspbian Lite;
- 3 Wireless USB adapters with MT7601 chipset;
- 1 GNSS USB adapter;
- 1 LTE USB dongle to grant access to the Internet;
- 1 Li-ion battery.

A Python-based software module allows for collecting data over multiple Wi-Fi channels through either fixed channel listening or channel hopping, as discussed in the following. The software module architecture allows for storing the Probe Requests locally and to send them at regular and modifiable time intervals. In case there is no Internet connection available, the collected data is stored in an internal SQLite database, which is then sent when the connection is restored. Some pre-processing operations are also performed at the stations to condition, compress, and clean the data. For instance, Probe Requests eventually sent by APs and malformed packets are discarded (e.g., packets whose length declared differs from the actual length). Another important operation implemented at the station is aimed at preserving user's privacy. Before sending the captured data to the cloud, the source MAC address is hashed with the PBKDF2 hashing algorithm which is one of the most resistant to "brute-force" attacks even if it is locally administered.

WiFi clients may follow two ways to access to the network: passive scanning and active probing [BHP07]. According to the passive mode, APs broadcast beacons packets to signal their presence, and the clients listen on channels for a fixed period

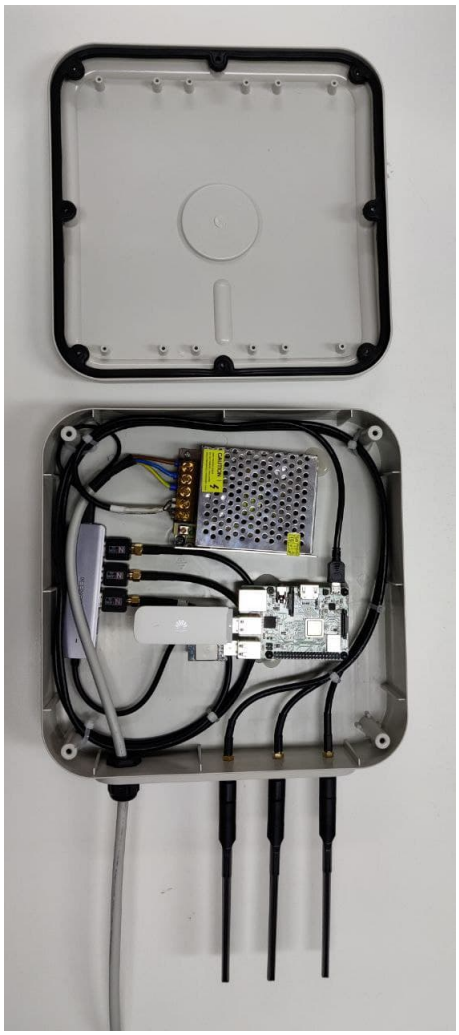

Figure 5.1: Probe requests sensor with multiple interfaces.

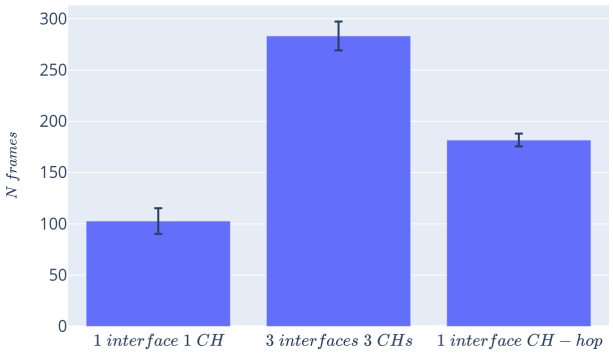

Figure 5.2: Number of frames captured using three different sniffing configurations: a single interface with fixed channel, three interfaces in three fixed channels, and a single interface with channel hopping (frame frames generated by a single smartphone in a semi-anechoic chamber).

of time. This approach, although completely passive, has a negative impact on the AP discovery process duration because the client needs more 1.1 seconds to listen in all channels if an AP is present [PR10]. Instead, in the active probing mode, the clients continuously send Probe Request packets to discover the presence of APs and when doing it they hop over the used channels with a brief pause between them (a typical pause value is between 10 and 50 ms) [PR10]. As a consequence, the active mode is shorter and for this reason is the preferred approach.

An important mechanism that must be investigated when implementing a Wi-Fi sniffer is the use of multiple monitoring interfaces and the eventual implementation of channel hopping. As many other not off-the-shelf WiFi sniffers, also the first version of PMA sensor had only one interface. However, it is in general true that increasing the number of antennas leads to an increase in the total number of packets captured. The ideal situation is when 14 interfaces listen on the different channels in parallel, i.e., one interface per channel for the 100% of the time.

To evaluate the benefits and drawback of different configurations, it has been conducted extensive experiments to analyze the number of packets captured using 1 interface or 3 interfaces for the sniffing operations. Herein, it is showed the results of a specific test that was carried out analyzing the total number of frames sent by a single smartphone (Honor9 with Android9.0) for 5 hours inside a semi-anechoic chamber. To acquire all the data, a sniffer with 4 antennas was used. One antenna was set to acquire data on all channels by performing either fixed sniffing in channel 1 (*1Interface1CH*) or channel hopping with 1 second of permanence time per channel (*1InterfaceCH-hop*), whereas the other 3 were set to acquire packets on fixed channels, specifically channels 1, 6 and 11 (*3Interface3CHs*).

Figure 5.2 shows the results in terms of the total number of Probe Request frames acquired by the station with the three configurations. It can be noted that an increase in the number of used interfaces leads to a higher number of captured

frames; specifically, around three times the number of frames have been captured with the three interfaces sniffing in parallel with respect to the case with only one interface. When using the channel hopping technique an intermediate result has been reached. In the developed sniffer I have used this approach.

5.4 The proposed de-randomization algorithm

The derandomization algorithm is applied to a trace of Probe Request frames that are captured during an observation window of length T^W. As it will be shown in the performance analysis section, the longer the observation window the better the performance of the proposed algorithm to correctly aggregate captured frames generated from the same device. On the other hand, the shorter the observation window the higher the temporal resolution of the monitored varying number of devices located in the area of interest. The number of captured frames heavily depends on the number of devices, the artifacts present in the area and the gain of the antenna; it ranges from 50 to 200 when $T^W = 10\,\text{min}$ considering the whole spectrum. The first operation performed is data cleaning, which is aimed at: checking for data errors, which may be introduced by the acquisition phase; deleting unnecessary information which would unnecessarily increase the computation burden; and introducing some corrections in case of missing values. The data cleaning process starts by discarding all the corrupted packets to avoid the introduction of errors or false information in the following operations. On the resulting trace, a first filtering is applied that is aimed at isolating the probes sent by devices that uses their real MAC addresses. It is performed by checking the $7th$ less significant bit of the $1st$ octet of the MAC address, as it has been shown in Figure 1.1. As a result, two subsets are obtained: the one of packets with real MAC addresses, from which it is straightforward to count the relevant devices if necessary; and the other one which is further processed to extract meaningful information. To further process this trace, I rely on some empirical observations that emerged from the analysis of several datasets:

- Most devices that implement the randomization process generate burst probe frame sequences where the virtual address is kept constant. In addition, the same device keeps IE IDs and the associated information content lengths constant over different bursts, even if the virtual MAC address changes. An example of this phenomenon is shown in Figure 5.3, where frames of only two devices are considered for an observation window of 60 min. In the graph, a burst of frames is represented by a green or red bullet. The frames within each bursts have the same virtual MAC in the source address. A burst detail is also shown in order to visualize the individual frames and their SEQs and TOAs variation. It has been possible to group the bursts in the two categories red and green associated to two different physical devices by checking the LENs of each IE ID for each frame. This has been possible because the two devices do not share the ID fingerprint. On the bottom of the figure, the used IE IDs

are shown for the two devices and the associated virtual MAC addresses that have been used.

- However, when many devices are in the same acquisition area, it happens that they share the same fingerprint and for this reason it is not enough to use only the ID fingerprint to discriminate the devices but a more complete analysis is required which considers the *speed* at which the sequence number field in the frames is incremented over time. Each device uses a speed which often characterizes its Request Probes generation process. This feature needs to be used together the ID fingerprint.

- Some other devices do not generate a new virtual MAC address every burst but only when they switch off and on their WiFi interface. These devices are easier to be detected and counted. If these sequences are detected, it is possible to make easier the identification of the devices for the remaining of the acquired trace. However, it is not always straightforward to identify the beginning and last frames sent by the same source. In the following, these devices will be called: *pseudo-random* devices.

These considerations have been used to develop the proposed algorithm, whose workflow is provided in Figure 5.4. In the following subsections it is described the feature extraction process, the filtering of the pseudo-random frames, and the clustering algorithm. To make easier the reading, the Table 5.1 summarizes the notation used.

5.4.1 Features extraction

As mentioned before, the features that have been found to be relevant for the proposed algorithm are linked to the IE ID, IE length, the sequence number and the time at which the probes are generated. In the following, it is described the trace processing procedures to explain the final features used. The first step consists in removing the unnecessary information from the captured traces and to represent the remaining data in the most convenient way. Specifically, the fields that are kept from each frame are: IE IDs and the IE length (called LEN in the following). Note that the IE content is not kept as this has demonstrated not to convey any additional distinctive information for the device counting purposes with respect to the LEN field. Together with these fields, the MAC address, the probe request's Time of Arrival (TOA), the sequence number (SEQ) and the received signal strength indicator (RSSI) are stored. An example of this information is shown in Table 5.2. For each frame captured, more than one row can be present as it is the case in the considered example for frame with $SEQ = 250$. For convenience in processing by keeping the same information, this table is converted into a sparse matrix as shown in Table 5.3, which highlights that there is a column for each IE ID associated with the corresponding length. As some Probe Requests do not convey some IEs, the

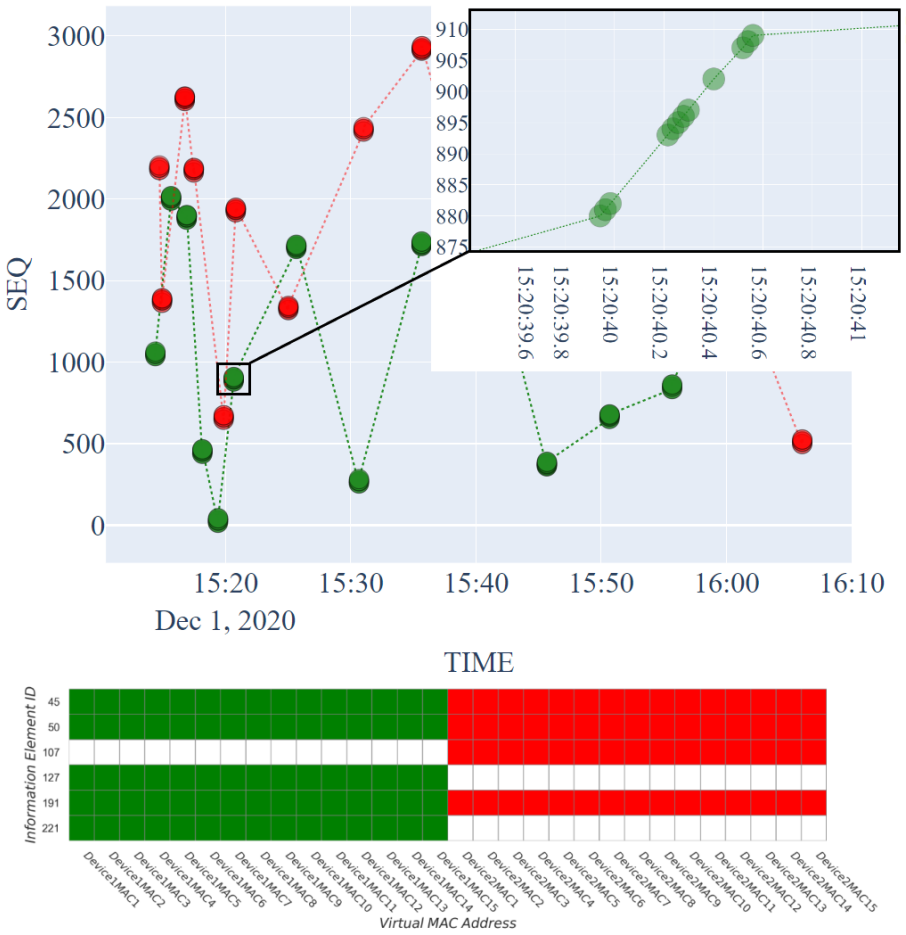

Figure 5.3: Sequence of frames generated by two devices in an observation window of 60 min with SEQs and IE IDs details.

corresponding content is not available ("NaN" value in the examples of Table 5.3). The resulting matrix has the following size.

$$\mathbf{M} \in \mathbb{R}^{n \times m} \tag{5.1}$$

where n is the total number of different Probe Request frames received in the observation window T^W and m is the total number of different IE IDs found over all the received messages plus 4 (due to the columns for the fields RSSI, SEQ, MAC and TOA). Due to the variegate set of devices that have emitted the captured frames, each row has different IE columns with no information. By analyzing different datasets, it was noticed that in the frames sent by the same device they contain a constant list of IEs and with constant lengths, as already discussed.

Furthermore, the different IE IDs have a different importance for the objective of the analysis as some are more frequently used. For instance, Table 5.4 shows the IE IDs that have been observed in a typical two hour-long capture and that are used by

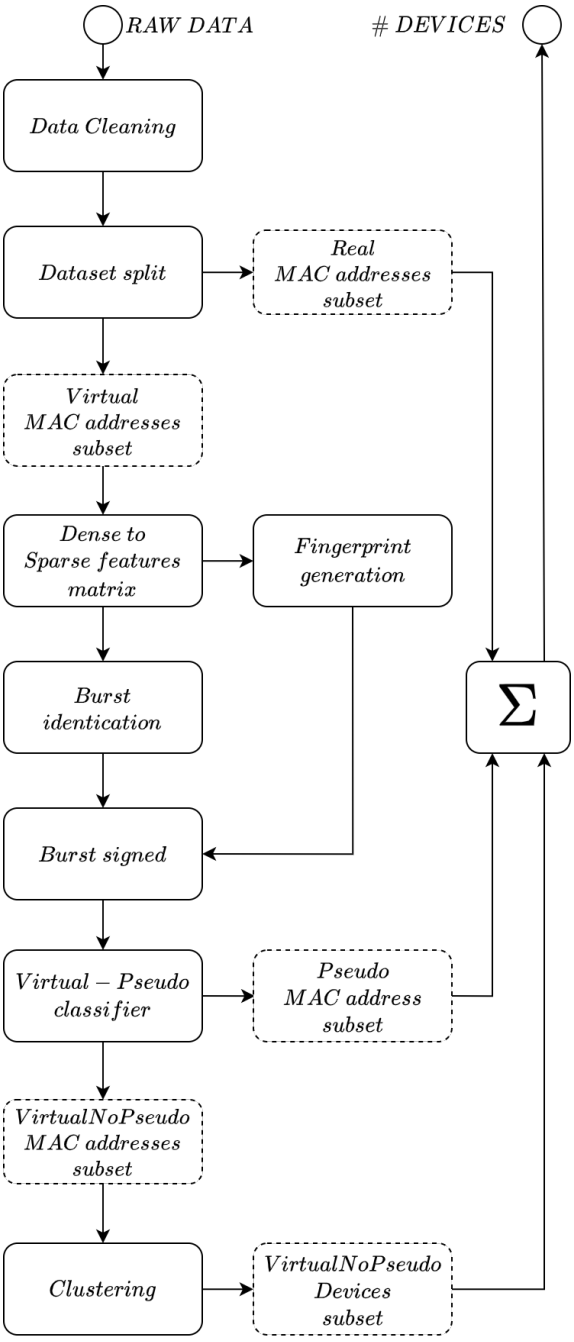

Figure 5.4: De-randomization algorithm workflow

Table 5.1 Used notation

Parameter	Meaning
T^W	Observation window
n	Index of the number of unique MAC address during T^W
M	Sparse matrix with probe request details
m	Number of IE IDs found in all captured packets
InF	Time between two consecutive frames
TOA	Arrival time for a frame
SEQ	Frame sequence number
k	Specific burst
i	Number of frame in a burst k
t, g	Used for point P to compute the burst rate
j	Specific virtual mac address
l	Length of burst k with mac address j
Θ	Cumulative time of all burst
Λ	Number of burst with MAC address j
Ψ	Percentage burst time presence
χ_i	Internal parameter for Optics clustering algorithm

Table 5.2 Example of information extracted from the Probe Requests

MAC	**Time of Arrival**	**SEQ**	**IE ID**	**LEN**	**RSSI**
MAC_1	1604571222.4325	250	45	5	-63
MAC_1	1604571222.4325	250	50	8	-63
MAC_1	1604571222.4325	250	221	6	-63
MAC_2	1604571281.9861	3500	127	10	-41

the devices at least 5% of the total number of frames. On the basis of this feature, it appears that the above mentioned IDs it is possible to identify a unique fingerprint for each device that uses different virtual address over separate bursts. This aspect was also analyzed in previous works, e.g. [UCA19a, UCF+20a]. Accordingly, I focus on smaller matrix focused only on the mentioned 8 IE IDs:

$$\hat{\mathbf{M}} \in \mathbb{R}^{n \times 12} \tag{5.2}$$

In this way, considering only 8 IE IDs, it is possible to reduce the features space and obtain an improvement in the performance of the clustering algorithm, in a such way it is ready for real-world applications. The following operations are then performed:

- *Burst identification:* the objective is to identify all the bursts and to give an identifier. A burst is defined as a sequence of frames with the same MAC

Table 5.3 Features extracted from Probe Request packets as sparse matrix

MAC	TOA	SEQ	RSSI	ID 45	ID 50	ID 127	ID 221
MAC_1	...	250	-63	5	8	NaN	6
MAC_2	...	120	-41	NaN	NaN	10	NaN

Table 5.4 Presence rate of the IEs on the frame total number

IE ID	Frames number	Presence Rate
50	24663	0.998
45	14953	0.605
3	13779	0.557
127	11979	0.485
0	10519	0.426
221	8659	0.350
107	2031	0.082
191	1635	0.066

address with inter-frame time always shorter than a given threshold, called maximum inter frame time t^{InF} (a typical value for it is 0.5 sec). A subsequent burst begins as soon as there is an inter-frame time that is greater than t^{InF}.

- *Fingerprint generation:* in this step the fingerprints based on the IEs are created. To this, starting from $\hat{\mathbf{M}}$, the whole set of IDs and LENs combinations associated with the different MAC addresses are identified and labeled with a unique identifier (named *fingerprintID*). Additionally, when some MAC addresses were observed to use multiple fingerprints (rarely) there were discarded not to create ambiguity in the analysis.

- *Computation of burst rates:* at this stage, all the frames are grouped by burst and an aggregation operation is performed to extract the descriptive characteristics of each burst. Then compute the *burst rate* is computed, as a representative characteristic of the single burst. To this let define $t_{k,i}^{TOA}$ and $g_{k,i}^{SEQ}$ as the TOA and SEQ number of frame i in burst k, respectively. Points $\mathbf{P_{k,i}} = (t_{k,i}^{TOA}, g_{k,i}^{SEQ})$ for burst k are then represented in a plane and a linear regression is computed. The angular coefficient of the resulting line is the burst rate, which is a distinctive feature of each burst and represented by P_k. This feature has been empirically observed to be similar for burst generated by the same devices and it is used for burst clustering.

Figure 5.5 shows an exemplary capture lasting 25 min where each dot represents a distinctive burst of frames with the same MAC address. A different color has been assigned to each different MAC address. It can visually detected that there is at

least one pseudo-random device (violet dots) whereas some random devices can be easily detected as the sequence numbers increase regularly drawing a line on the 2D plane.

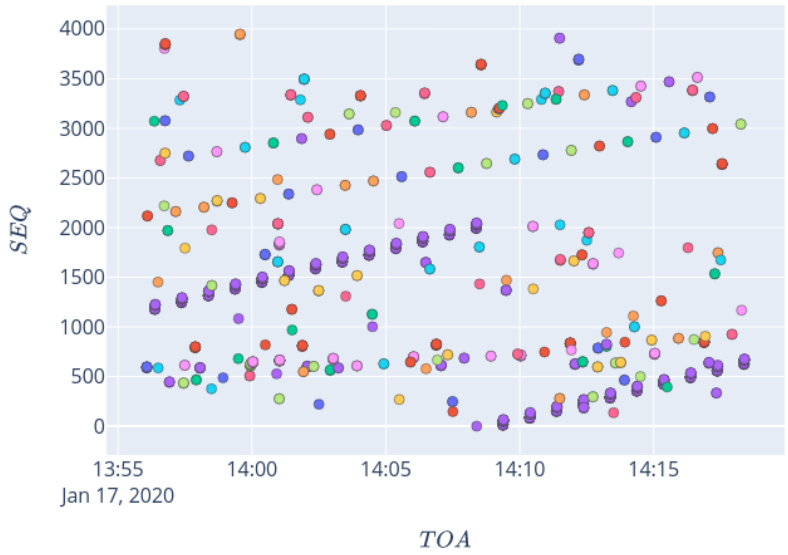

Figure 5.5: 25 minutes capture of Probe Requests: each dot represents a burst of frames, different colors means different MAC addresses.

5.4.2 Pseudo-random MAC filtering

Pseudo-random devices are those that use the same virtual address for many bursts. It is then appropriate to identify these devices and to remove the generated bursts before applying the clustering operation. The reason is twofold: i) the fact that the same MAC is kept constant in a group of bursts makes this operation simple and reduces the complexity of the next clustering operation; ii) they can create some noise in the clustering and reduce its performance. However, the identification of these bursts is not that straightforward as there is not a fixed length of this sequence of bursts and the MAC address used by the pseudo-virtual device may be also used by other virtual address devices. Accordingly, a specific procedure had to be devised. To this, three metrics has been considered:

- the *cumulative time of all bursts* having the same virtual address j

$$\Theta_j = \sum_k l_{k,j} \tag{5.3}$$

  where $l_{k,j}$ is the *length of burst* k having the same MAC address j;

- the number of bursts Λ_j with the same MAC address j;

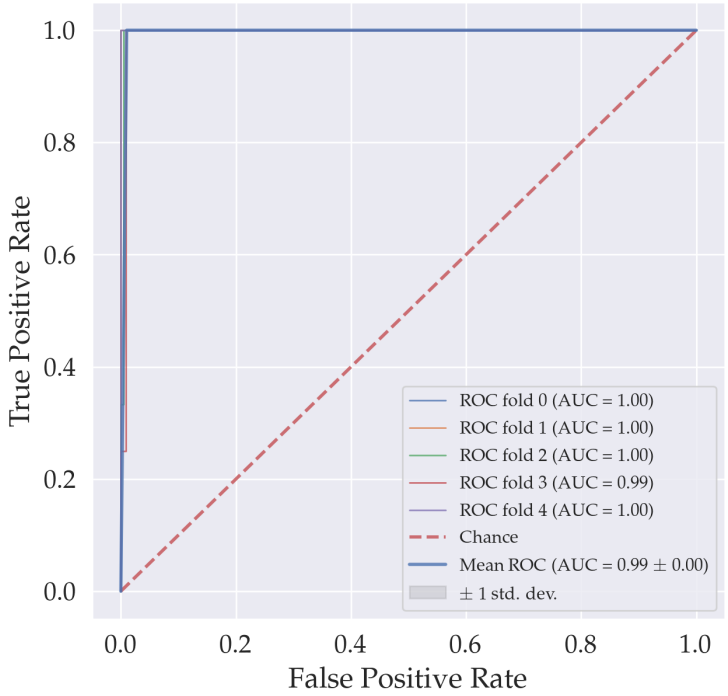

Figure 5.6: Receiver Operating Characteristic (ROC) curve of the SVM classifier applied to the identification of the Pseudo Virtual frames.

- the *percentage of time in the observation window* T^W during which the bursts are observed which

$$\Psi_j = \frac{t^{TOA}_{k,last} - t^{TOA}_{k,first}}{T^W} \tag{5.4}$$

  where $t^{TOA}_{k,first}$ and $t^{TOA}_{k,last}$ are the TOA of first and last frames having the same MAC address in T^W.

The higher the value of these parameters the higher the probability that these bursts have been generated by a pseudo-random device. These features have been used as features space for a Support Vector Machine (SVM) binary classifier [P+99], which has been trained using long traces where the pseudo-random devices were known. Figure 5.6 shows the Receiver Operating Characteristic (ROC) curve. AS it is well know, when the curve reaches the top left corner of the plot I have very good results, i.e., a false positive rate of zero and a true positive rate of one. The chance line represents the ROC curve when the classifier has an equal probability to predict correctly or wrongly. From the mean ROC curve it is possible to deduce how the chosen features are able to divide the samples well into the two classes of belonging (Virtual and Pseudo Virtual), reaching 100% accuracy values in almost all the 5 iterations of the k-fold cross-validation.

5.4.3 Frame clustering

The output of the previous operations is the division the trace into frames with *real*, *pseudo random* and *virtual* source addresses. It is possible to discard the first group and for the second group counting the number of devices is straightforward. For the third group, a proposer clustering operation has to be implemented. To do this, the following features are used for the identified bursts: the *burst rate*, the set of IE's LENs and the average inter-frame time InF_k of the burst k. From the obtained features it is possible to proceed with the clustering of the bursts. Among the possible approaches, the density-based one was selected as it has the advantage of being able to create arbitrary number of clusters and can reach high level of scalability if properly configured. Among the density-based algorithms, there is a well-known algorithm which is called Density Based Spatial Clustering of Application with Noise (DBSCAN) [EKS+96] and its Hierarchical version HDBSCAN [CMS13]. Other alternative solutions have also been proposed, such as the Fuzzy Joint Points (FJP) [Nas06] and the Noise-Robust Fuzzy Joint Points (NRFJP) algorithms; however, these methods suffer from the low speed of the FJP algorithm. Therefore, it is not recommended for use in those applications which need to manage large datasets. Finally, the a successor of the DBSCAN has been proposed, i.e., the OPTICS algorithm (Ordering Points To Identify the Clustering Structure) [ABKS99]. Density-based clusters are defined in the features space as variable density areas separated each other by more rarefied areas. The idea could be explained introducing the definition of *core-points*, *density-reachable points*, *density-connected* and *outliers* or *noise*. A point is a core point if its neighborhood of radius ε contains at least $MinPts$ points. A point u is defined a *directly-density-reachable point* if it is in the neighborhood of a core point p. However, a point could be only *density-reachable* if there is a transitive closure of direct density-reachability, i.e. if there is a third point r from which both u and p are density-reachable. Finally, outliers are defined as the set of points in the dataset which do not belong to any cluster.

In the following strategy will be specifically adopted the DBSCAN, HDBSCAN, OPTICS algorithms, which have the following specific features:

- *DBSCAN*: once ε and $MinPts$ are given, the clusters' density is defined and is not possible to change it during the clustering process. In the experiments, good results has been obtained with with $MinPts = 2$ and ε varying from 0.0001 to 0.1.

- *OPTICS*: it is based on the principles of DBSCAN and follows all its definitions, but addresses one of the major weaknesses of DBSCAN, i.e. finding important clusters in the data varying the density threshold ε. To this, the samples are linearly ordered in such a way that spatially close points become neighbors. Putting these points in a x-y plane with the ordered points in the x-axis and the reachability-distance on the y-axis, I see a *reachability plot*. Such kind of plot allows to see different density clusters and to calibrate the bound-

ary of a cluster simply by using the value of its derivative χ_i as a threshold. In this implementation its value is has been set to 0.1.

- *HDBSCAN*: Hierarchical DBSCN has been developed by Campello, Moulavi, and Sander [CMS13], with the objective to provide only a *flat* (i.e., non-hierarchical) labeling of the patterns, based on the global density threshold ε. This allows HDBSCAN to find clusters with different densities (unlike DBSCAN), and be more robust to parameter selection.

The features space is composed of all the LENs, the median inter-frame time and the rate burst. Therefore, by applying the clustering algorithms it is possible to discriminate the different devices. A results comparison with the various clustering algorithms is shown in section 5.6.3, in fact once the clustering has been finally performed, the algorithm has produced three subset of devices:

- Devices with a Real MAC address.
- Devices with a Pseudo Virtual MAC address.
- Devices with a set of Virtual MAC addresses.

Discarding the real MAC addresses, the sum of element belonging to each subset gives the estimation of the total number of unique devices which have been observed in T^W in the area where the capture has been performed. However, the lists are kept separate as other metrics can be derives, as those that rely on the identification of devices whom come back after an interval of time for the devices which has a real MAC address.

5.5 Results in a controlled environment

To analyze the performance of the devised de-randomization algorithm, some experiments has been conducted where it has been possible to isolate the test devices from other external ones. For this reason it was conducted data acquisition sessions in a semi-anechoic chamber located at the Faculty of Engineering of the University of Cagliari. Figure 5.8 shows the setup with all the test devices and the laptop that implemented the sniffing process and the proposed algorithm for de-randomization. In this experiment only one interface was used that was sniffing on channel 1. Table 5.5 lists the number of test devices with the operating systems.

Figure 5.7 shows the results at varying observation window T^W. As expected, the longer T^W the better the performance. The reason is twofold: when the observation time is short, the station is not able to capture any frame from some of the test devices; when only few frames are captures from each station, it is not possible to compute the *burst rate* feature that is used to separate frames emitted by different stations and that are then separated by means of this feature. Indeed, note that at

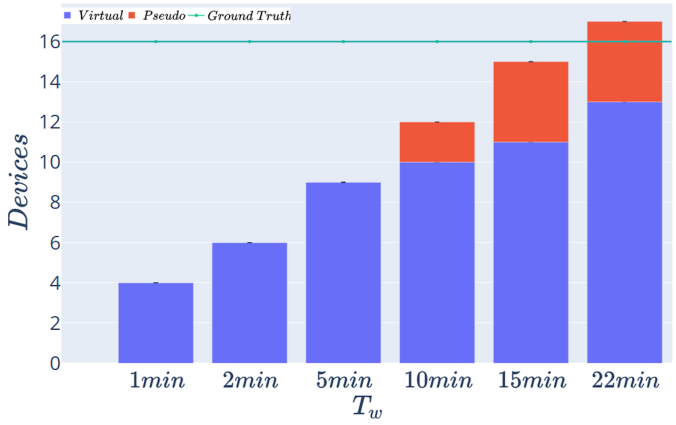

Figure 5.7: Semi-anechoic chamber tests: number of devices counted with the proposed algorithm varying the observation window.

short observation windows the algorithm under-counts the number of devices. It can also be observed that an observation window T^W of at least 10 minutes is needed to start distinguishing devices with pseudo-virtual MAC addresses from those with a virtual MAC addresses. The main reason is that in 10 minutes the algorithm is not able to identify repetitions of runs with the same MAC address. The best result obtained in this scenario is with T^W set to 22 minutes, reaching an accuracy of 97% using DBSCAN as core clustering algorithm, which means that this work has introduced an error of less than 0.5 devices over total of 16 test devices. This result represents an excellent basis to build solutions that can work in real scenarios. For this reason, it is a good idea to proceeded further with the development and testing of a solution for an Automatic Passenger Counting system to be used onboard of buses. Note that in this test the devices using real MAC addressed have not been included in the analysis but these have been excluded as soon as these have been detected.

Table 5.5 Test devices used in the semi-anechoic chamber test

BRAND	**# of devices**	**OSs**
Huawei	5	Android 5.1(1)/6(1)/8(2)/9(1)
Samsung	6	Android 4.2(1)/7(1)/8(1)/9(3)
ZTE	1	Android 8
Xiaomi	4	Android 6.1(1)/9(3)
Honor	1	Android 9.1
Apple	5	iOS 12.4.4(1)/13.3(4)
Motorola	1	Android 7

Figure 5.8: Setup of the anechoic chamber tests

5.6 Results onboard of city buses

This section illustrates the results obtained when using the devised algorithm to develop an Automatic Passenger Counting system. The following subsections present data acquisition setup, the analysis of the data and the counting performance.

Table 5.6 Details of the Datasets used

Dataset Name	Anechoic Chamber	Line 1 Brotzu	Line 1 Gioia	Line 30 Brigata Sassari	Line 30 Matteotti	**Totals**
Length [mins]	32	22	48	28	34	164
#Pkts	9707	15589	27107	22568	25693	100664
#MACs	300	1187	2032	1101	1695	6315
#Real MACs	31	363	524	285	278	1481
#Real Pkts	3598	6285	12235	12610	11955	46683
#Virtual MACs	269	823	1507	815	1416	4830
#Virtual Pkts	6109	9184	14504	9887	13579	53263
Ground Truth	16(v) + 8(r)	32	45	20	43	

5.6.1 Data acquisition setup

Three sniffers have been installed on-board of three different buses of the CTM S.p.A. local public transport service company that operated in the city of Cagliari, Italy. The buses were scheduled to work on a different service lines every day. This aspect allows to acquire a significant amount of data on the behavior of travelers when using urban transport services in the different areas of the city. In each bus under test, a sniffer was installed above the central door to better cover the whole bus area.

As for the power supply, different configurations have been tested. In the first trial, the sniffer's has been powered by the on-board services power line; thus, when

the bus engine turns off, the power lines is also switched off causing the sniffer to terminate the acquisition abruptly. This introduced a problem when the bus reaches the last stop where the engine is turned off while people continue to get on and off the bus. On the other hand, using the power lines under the battery is not a solution because the sniffer could drain the battery if the vehicle is not put into service for a long time. By studying the electrical system of the bus, I have identified a viable solution which is presented by connecting the sniffer power line to the power line of the ticket machine; indeed, this line is stabilized and it has a shutdown time which is delayed of 20 minutes after the bus engine is turned off. Accordingly, the sniffer could acquire the data of interest even at the last bus stop, when the bus engine is off, and at the same time, it turns off after 20 minutes once the bus has been turned off at the end of the service, preventing the battery from being discharged.

The acquisition took place from July 2020 to October 2020. Among the several data acquisitions that I collected through the sniffers installed on board the buses, I chose two specific bus service lines to be analyzed with tests performed in both directions. Specifically, I have selected line 1 (directions Brotzu and Gioia), which is one of the longest lines, and line 30 (directions BrigataSassari and Matteotti), which is the service line that connects the city with the second most populous city in the metropolitan area.

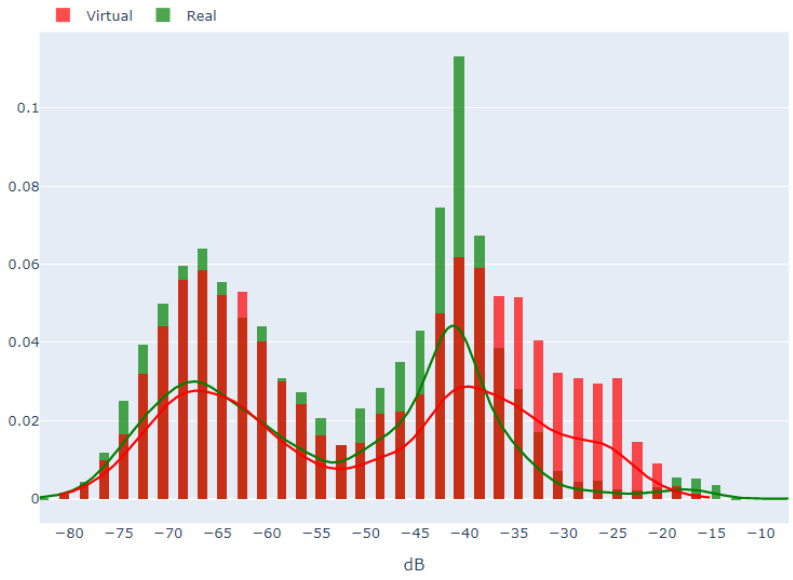

Figure 5.9: Power distribution

5.6.2 Dataset analysis

Table 5.6 provides the major information of the datasets analyzed in the following. In the table, there are also provided the information of the semi-anechoic dataset

for comparison purposed. To analyze and understand the efficiency of the algorithm was necessary to have the ground truth on how many person were on-board and got on and off at each stop. To this, I spent a few days by counting the people manually. In 5.6, I have also added key ground truth data: the number of devices in the semi-anechoic chamber which was constant over the observation window and total number of people in the bus which was not always in the bus as they entered and exited the bus in different stops during the observation period.

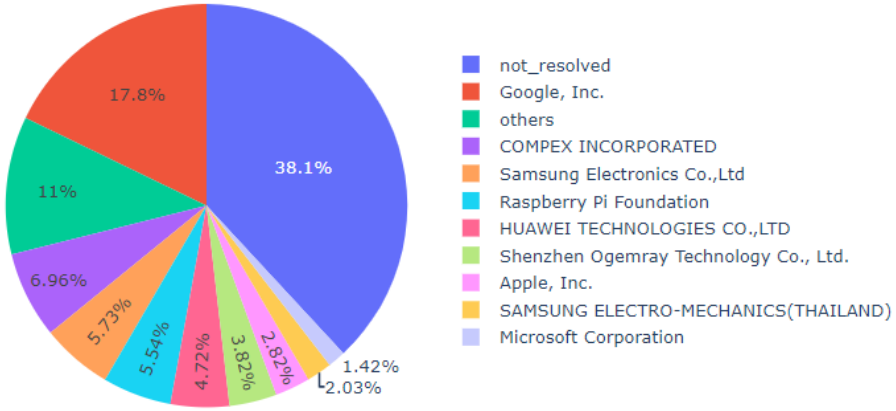

Figure 5.10: Vendors distribution

Figure 5.10 shows the distribution of the devices that have been observed among the various vendors. In "others" I have grouped 84 vendors that appeared during our tests with a presence lower than 1%. "not_resolved" represents the amount of traffic of which it was not possible to find the related vendor as some vendors perform randomization in the OUI field as well making impossible their identification. By analyzing the vendors identities it is easy to recognize that there are some that do not produce smartphones nor WiFi interfaces. This is the case of the category "Shenzhen Ogemray Technology Co., LTD", which represents an IoT solutions manufacturer that does not produce WiFi interfaces for smartphones. This means that the sniffer is acquiring probes sent by domestic devices inside the buildings close to the street where the bus is passing by. Indeed, this issue is particularly relevant when the bus stays at the bus stops as during this time-frame is more probable to capture stationary devices installed inside the buildings. For this reason, several processes to clean the data have been introduced, as explained in Section 5.4.

Splitting real and virtual MAC addresses and taking another look at the vendor distribution, the reality is more clear. All the virtual MAC addresses have Google as vendor id for 32.8% of time, whereas for the other 67.2% the id is not resolvable. No other vendor is explicit in frames sent with a virtual MAC address which increases the difficulty in distinguishing devices.

Figure 5.9 shows the power distribution for the captured frames, where it is possible to observe a double Gaussian-like distribution centered at two different

power levels. It is obvious that one represents the devices within the bus, whereas the others those that are outside. This clear distinction among the two groups of devices suggested us to introduce a power threshold to filter out the frames sent by devices that are aboard the bus from those that are not. Accordingly, I empirically determined this threshold by observing the power of frames generated by devices with known real MACs and varying the distance from the sniffer. I found that the threshold that maximized the probability to identify the devices that were on-board was -53dB. It is important to highlight that this works for the experiments specific setting and a model that could work in different setting needs to be devised.

5.6.3 Performance analysis

Figure 5.11 shows the number of passengers counted with the proposed algorithm on 4 sessions for the 4 different lines described above. The figure shows the results of the counting performed taking in account the whole acquisition for each line and computing the number of devices at every bus stop. The graphs shows the performance when using the three different clustering algorithms to analyze the impact of the different strategies. In particular, it can be seen that the features taken into consideration represent devices with constant density clusters; for this the DBSCAN clustering algorithm almost always provides more accurate estimates of the devices on-board of the bus. An important insight is the presence of some spikes in the estimation; this is due to the people that are waiting out of the bus near the stop and that could not be excluded by our algorithm. In this respect, the OPTICS and the HDBSCAN algorithms suffer more of this issue.

Because in situations of heavy congestion, but even more so in situations where it is possible that probe requests come from passengers waiting for other buses near the stops (e.g. squares or stations), they generate noise in the features space with sporadic and rarefied points that the OPTICS and HDBSCAN algorithms mistake for low-density clusters, i.e. more devices.

However, the issue could be easily resolved removing all the packets captured when the bus' doors are open. However, this cannot be implemented at the moment as the ground truth data does not include the information on the length of each bus stop.

Table 5.7 summarizes the results with the relative error (average and standard deviation). As already highlighted the DBSCAN algorithm provides better results, with an accuracy as high as 74.25%. Finally, Figure 5.12 shows the several devices traces and the pool of virtual MAC addresses which have changed over time. It is a graphical result of the algorithm, through which it is quite simple to count the number of unique devices that have appeared in the T^W, simply by counting the number of colored lines. This is an important result because it allows you to do a temporary tracking (limited in a few hours time) of the devices that use randomization. Furthermore, by applying the algorithm to the urban public transport scenario, it is possible to easily create Origin-Destination matrices and automatically derive the

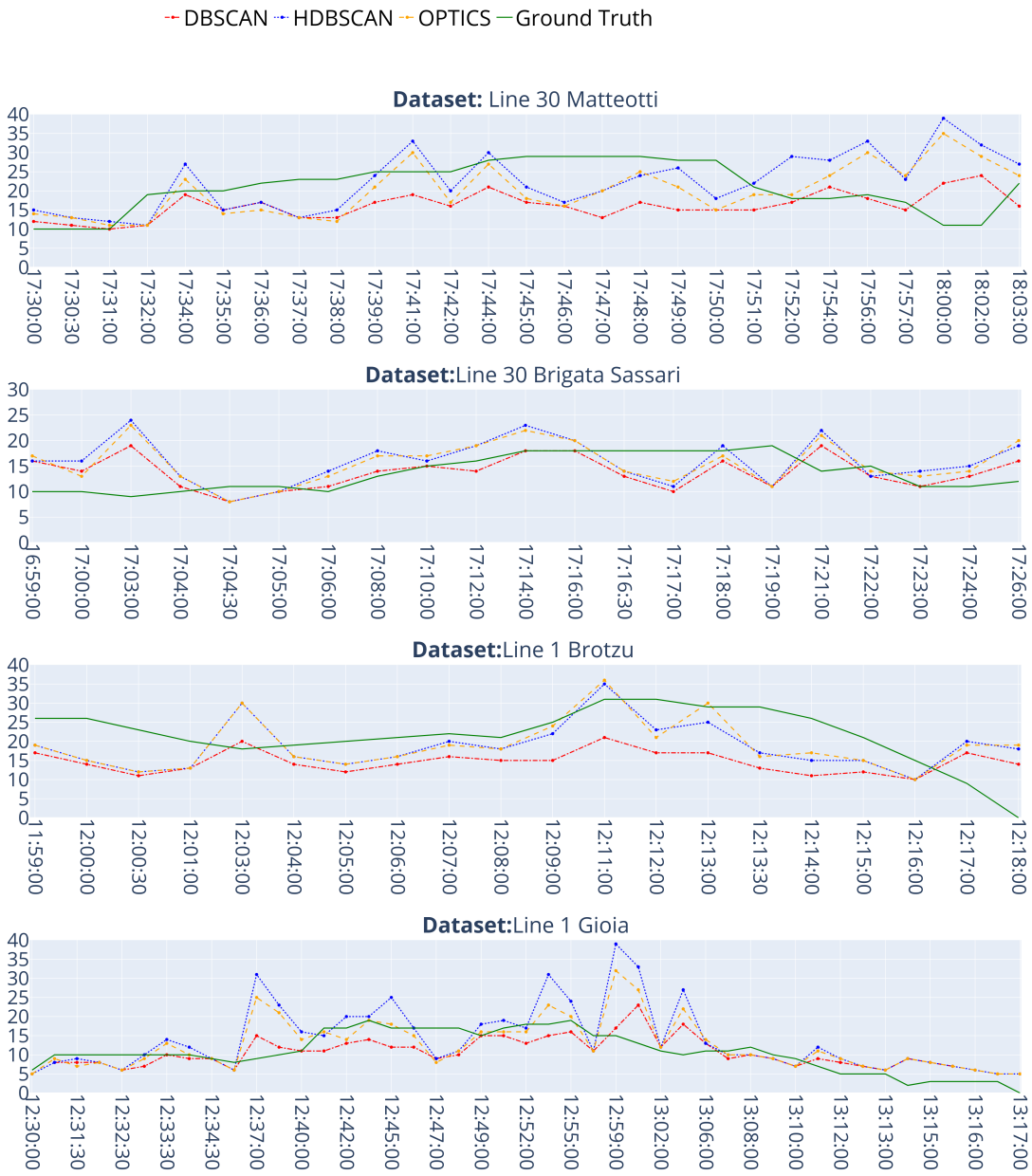

Figure 5.11: Number of devices estimated for each dataset in the bus scenario

Table 5.7 Estimation results errors

Scenario	Dataset	HDBSCAN	DBSCAN	OPTICS
Lab	Anechoic Chamber	3%	3%	3%
Real	Line 1 Brotzu	40%	31%	29%
	Line 1 Gioia	24%	27%	26%
	Line 30 B.Sassari	13%	27%	30%
	Line 30 Matteotti	29%	32%	32%
	Average	**26.5%**	**25.75%**	**29.25%**

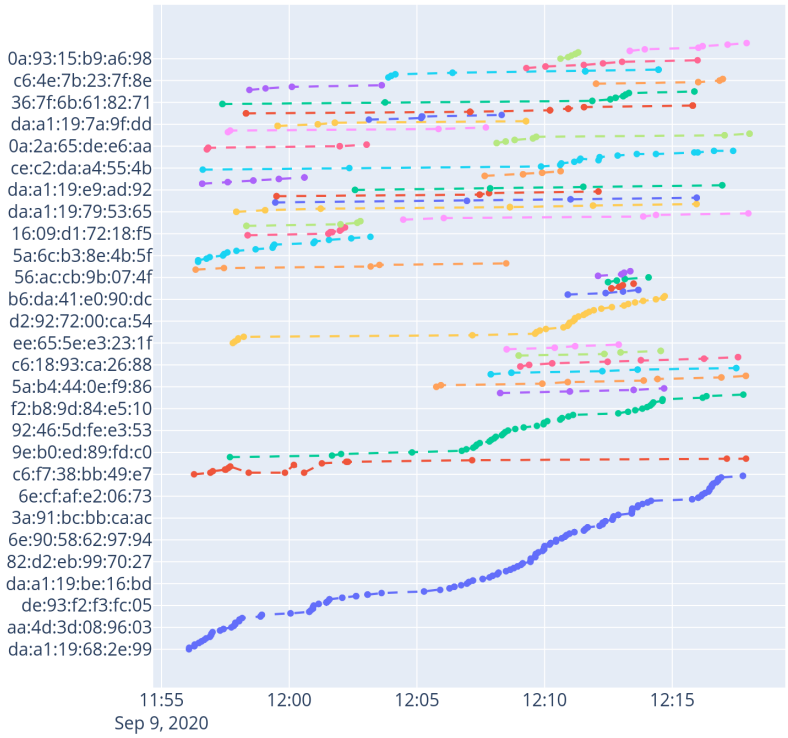

Figure 5.12: Device traces with pool of used MAC addresses (Dataset Brotzu)

demand of mobility in order to understand how citizens use public transport.

5.7 Conclusions

A novel de-randomization algorithm has been presented, which relies on clustering Probe Request frames by considering the content and the rate at which the frames are emitted. The algorithm has been tested in a controlled environment, where only the frames generated by the test devices have been captured bringing to an accuracy of almost 97% when the observation window is at least 22 min long. This result represents an excellent basis to build solutions that can work in real scenarios. For this reason, it has been proceeded further with the development and testing of a solution for an Automatic Passenger Counting system to be used onboard of buses. In this scenario the average accuracy has been as high as 75%.

Chapter 6

Open Data as objective data source for people mobility monitoring

The impact of the 2020 COVID-19 pandemic has had strong repercussions on all aspects of our life, sometimes even changing our habits. In this chapter we analyze a large dataset containing information on the traffic of the city of Cagliari, which located in the center of the Mediterranean Sea is one of the most popular tourist destinations, in this context it is shown how the pandemic has changed not only traffic volumes but also his model. In this work the state of city traffic is compared with the different levels of restriction imposed by the central government. The first lock-down led to a 76% reduction in traffic, while subsequently, although the measures were about the same, the reductions were less impactful. Thanks to the official tourist presence data, it was possible to identify the points most involved in tourist traffic and therefore pay more attention to those sensors. Basically, the analyzes show that the absolute volumes have naturally had a large reduction, but also the weekly and daily patterns have changed, although the latter have maintained greater consistency.

6.1 Introduction

Comparing cities to the beating heart of human life, the same could be done with city traffic and blood flow or roads like arteries. So referring to this duality, it is possible to think that traffic traceability models not only speak to us about citizens' mobility models, but also provide a good indication of how the city road infrastructure works in terms of efficiency. These are the reasons that lead researchers and administrators to monitor and analyze traffic flows, their volumes and models in order to have a clear and precise vision of how citizens live the city and how they react to anomalous situations.

The COVID-19 pandemic has been configured in some respects as a unique opportunity to understand how mobility patterns change in emergency health situations. Furthermore, thanks to the various contagion prevention measures, a unique context has been created to analyze the critical mobility models or those that keep the basic functions of the city alive. In fact, the everyday life lived for most of 2020 did not leave room for recreational activities or tourism, allowing only the really necessary trips.

Figure 6.1: Traffic measurement stations in the city.

In this chapter, a large set of data relating to urban mobility is analyzed, generated by 167 traffic sensors, organized in 98 measurement stations spread across the urban scenario of Cagliari and in continuous operation since 2016. Figure 6.1 shows their configuration on a map so that you can easily understand where they are located. Although the onset of the pandemic has aroused the interest of the world of research on urban mobility, in this chapter the studies are focused on tourism, analyzing times, volumes and models of tourist mobility starting from the **??** section.

The analysis is entirely based on the open data made available by the municipality [1]. However, although the sensors are maintained and kept operational by the municipal administration, a great deal of conditioning work was required of the raw data that otherwise would have been unusable due to problems relating to the transmission or processing of data before their transmission. The 6.3 section describes how the anomalies were detected, selecting a subset of the data deemed acceptable and the use of Machine Learning to carry out predictive models.

Although some results could be easily deduced, such as the reduction in traffic volume, some interesting ideas are provided in the 6.4.1 section for more refined reasoning about mobility during emergencies. As a first thing after the initial restriction which led to the 76% reduction in traffic, the subsequent restrictions did not have the same impact, and were therefore perceived differently by the population. Subsequently, it can be seen that the daily traffic pattern has remained roughly the same during the hours, while the weekly traffic pattern has changed drastically.

[1] `https://opendata.comune.cagliari.it/portale/it/st04_api_cloud.page`

6.2 Related work

The incidence of COVID-19 on everyday life and habits has been devastating, therefore the study of the same incidence has become a global goal for the academic sector. As might be expected, numerous studies have been carried out and as many phenomena have been highlighted all over the world. Clearly, in addition to the direct impact on citizens' health, there are also other phenomena directly generated by the pandemic. Among these in the following we will see some that have focused mainly on city mobility and on the variation of traffic flows.

In [ABC+20] the authors focus on noise pollution and report a significant reduction (-64%) of travel connected to the use of private vehicles in Rome. In their work, unlike ours, the data comes from the cars of the car sharing service [2] and not from IoT sensors.

The authors of [HHK20] compare the speed and length of travel on a stretch of European road, the E75. Comparing the traffic flows to and from the Slovak Republic before and after the implementation of the restrictive measures. This time the data source is the same one mentioned in this chapter, but with a substantial difference in the number of sensors that in the work related to this chapter were 98 while in [HHK20] only 2.

Social distancing represents another reason that leads to changes in mobility patterns. This conclusion was drawn in the work done in [De 20], in which it is discussed how social distancing has strong implications on how people live their daily lives, and therefore also on their movements. The influences given by these implications have been negative as people tend to travel less on public transport, sometimes avoiding meeting their relatives.

In [WWLL20] the authors analyzed the patterns of mobility at the national level, noting that the changes were driven mostly by citizens' awareness of the pandemic's consequences, rather than by the measures imposed by the central government. Thanks to their results we can see how, for the mobility models to reach stability it takes at least 14 days, this result was also found in the paper from which this chapter drew inspiration.

The reduction in traffic volumes, among other things, has led those who instead occupied the streets to carry out more illicit behaviors and follow questionable behaviors, such as increase in speed, increase in sudden acceleration / deceleration (+12%) and increase in use of phones while driving (+42%), these and other results are shown in the work in [KMSY20].

The virus has also brought about a very important change in air quality, due to restrictions on mobility, drastically reducing the emission of carbon dioxide into the atmosphere. Although air quality is not the subject of study in this chapter, it is worth mentioning the works carried out in the various states, for completeness I mention some such as that carried out in Brazil [DSF+20], India[SZA+20], Morocco

[2] `https://en.wikipedia.org/wiki/Floating_car_data`

[OBT+20], Kazakhstan [KBI+20], Spain [TCR+20, Bal20].

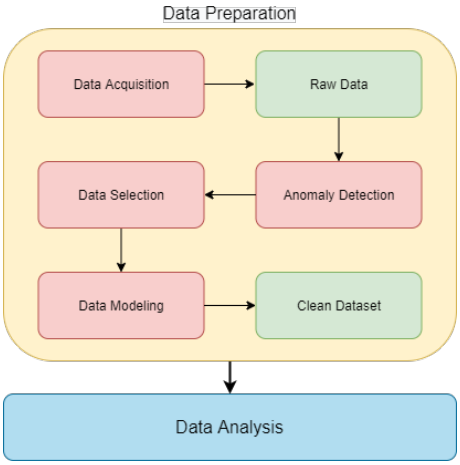

Figure 6.2: Data pipeline; from raw data to a clean dataset.

In [JY20] they demonstrate through the scientific method how by studying the population density and carrying out constant monitoring of air flows it is possible to reduce the rate of contagion.

Finally, in [CWW+20] the authors make an in-depth analysis of Wuhan traffic (which I recall is probably the first city to have been hit by the pandemic) and list some metrics to be monitored to manage the emergency.

6.3 Data Science Pipeline

In this section, mainly through the fig. 6.2 is shown the entire data science pipeline devised into two main phases, data preparation and subsequent data analysis, both of which are further explored in section 6.3 and 6.4 respectively.

6.3.1 Data acquisition

Since 2016 the municipality of the city of Cagliari has started to install inductive loops in the asphalt with the intention of monitoring vehicular traffic, Figure 6.3 shows the features of these sensors. The data are released by the municipality in an open way and are usable through REST API in an easy and fast way, in order to carry out the analyzes, the databases were queried to download the data from 1 January 2016 to 31 December 2020.

6.3.2 Anomaly detection

Unfortunately, from an initial preliminary analysis it was immediately clear that the data presented various anomalies. Therefore, before they could be used in data to

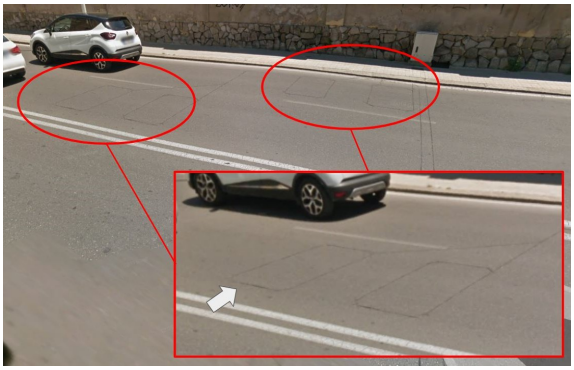

Figure 6.3: Inductive loops of traffic sensors installed in Cagliari.

derive information and evidence, they were cleaned with two levels of refinement. The first filter was done by setting a rigid and high enough threshold of the absolute value of the flow measured so that this was considered a valid data. Taking into account that the stations can have multiple measurement sensors (one per lane), therefore the threshold has remained constant as long as the number of lanes remains constant.

The most refined and reasoned filter was implemented with the Prophet library, created by Facebook [3] for forecasting time series data based on an additive model where non-linear trends are fit with custom seasonality, discovering outliers and remove them from original data

6.3.3 Data selection

The cleaned dataset has several missing values, not only because they are anomalous but also due to failures of the acquisition or transmission system. This situation has led to a careful choice of the stations that are still statistically significant, in this regard the stations that in 2020 sent at least 60% of the expected data were chosen.

6.3.4 Data Modeling

As usual in the context of Data Science, the last step of data pipeline is that of data modeling, a fundamental step for defining a correct strategy for imputing missing data, studying its seasonality and discovering the correct evidence located within the data. Here, too, the Prophet library comes in handy, so that is it possible view the data without anomalies or missing values.

[3] `https://facebook.github.io/prophet/`

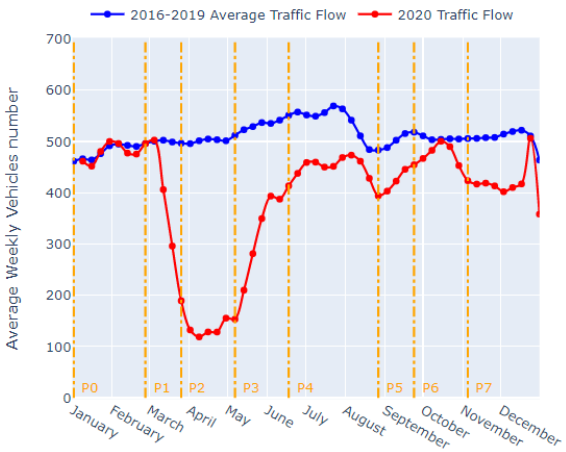

Figure 6.4: Comparison between 2020 traffic volumes and the average of the 2016-2019 volumes.

6.4 Data Analysis

6.4.1 City Traffic Analysis

This section shows how to derive traffic behavior in terms of volumes and characteristic trends. In order to get an idea of the total traffic volume under normal conditions, it is useful to see the data by looking at a pre-pandemic time frame.

The figure 6.4 shows the hourly average of vehicles during 2020 compared with the average of the previous three years. The eight orange lines divide 8 traffic variation areas. Table 6.1 describes them in detail. Furthermore, it is possible to associate these behaviors for example to the first first lockdown that affects the traffic between $P0$ and $P1$. After the first lockdown, we associate $P3$ with the rapid recovery of traffic, without however reaching the average of the previous years during the summer at $P4$, most likely due to the absence of tourists in the city.

The $P6$ period was characterized by the few restrictions imposed by the Government, we see how traffic returns for a short time to pre-pandemic levels. In this context, it must be borne in mind that Cagliari is a tourist city near the sea and that in October and November the tourist flow is always low. Finally, the drop associated with $P7$ clearly communicates the beginning of the second wave.

Figure 6.5 summarizes the traffic distribution patterns in relative terms. The values reported in the heat-map communicate the variations in traffic in relative terms between the values measured in 2020 and those of the previous three-year period. Green expresses a low difference (or high correlation) in the vehicular flow between the two periods, if instead the variations are more significant, they will be highlighted first with yellow then with red.

The greatest variations are recorded in the period associated with the first lockdown, i.e. between $P1$ and $P3$. In fact, there were strong reductions in volume in

Table 6.1 PERIODS DETAILS

	Average	Variation	From	To
P0	483.0	NaN	2020-01-02	2020-02-26
P1	277.0	-0.43	2020-02-27	2020-03-25
P2	145.0	-0.48	2020-03-26	2020-05-06
P3	368.0	1.54	2020-05-07	2020-06-17
P4	445.0	0.21	2020-06-18	2020-08-26
P5	444.0	-0.00	2020-08-27	2020-09-23
P6	463.0	0.04	2020-09-24	2020-11-04
P7	413.0	-0.11	2020-11-05	2020-12-31

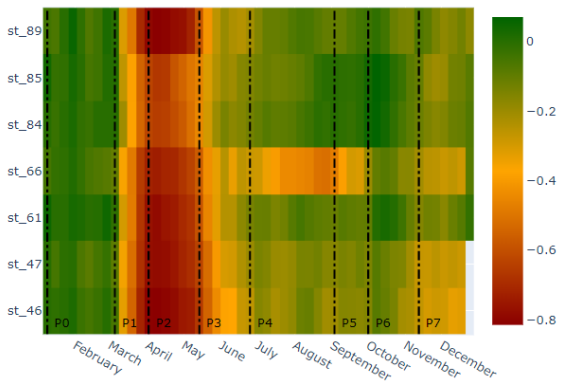

Figure 6.5: Variation of the 2020 traffic values as a fraction of the pre-pandemic volumes, avareged in 2016-2019.

the $P2$ period, with two other local highs positioned around the end of the year. The evidence that arouses the most interest in this visualization is how the traffic is distributed along the sensors. In fact, different stations have different traffic variations within the year, clearly denoting how traffic has moved to different areas of the city, as well as obviously having a general reduction in absolute terms.

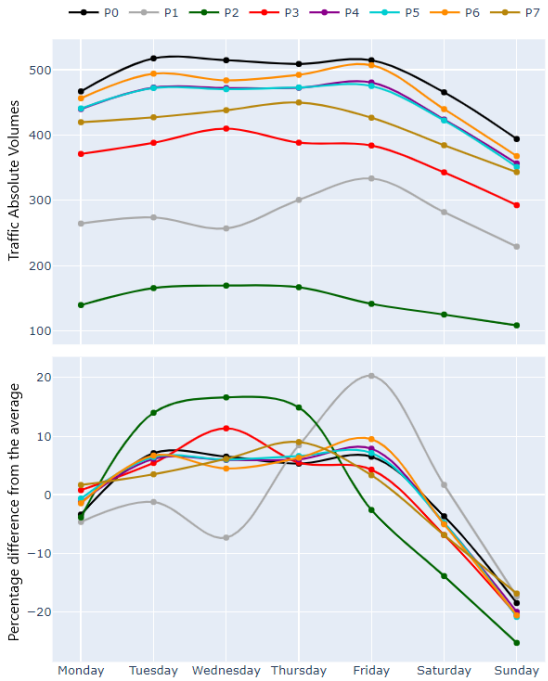

Figure 6.6: Weekly traffic distribution for each period, giving absolute values in the figure above and relative values in the figure below.

By analyzing the weekly traffic in each period and in particular the distribution in absolute and percentage terms, it is possible to communicate the results through Figure 6.6. What you see is how the mobility patterns change dramatically during the eight different phases.

Commenting in detail on each period, it can be said that in the $P0$ period, the one preceding the pandemic, the traffic is equally distributed on all days with the exception of the weekend when there are slight decreases, as is expected in an urban environment. The period just before the first $P1$ lockdown is characterized by a completely different weekly cycle, peaking on Friday. Indication of how habits have changed along with the absolute reduction in traffic (-76%). Throughout the time relating to the first $P2$ lockdown, traffic volumes remain stable, presenting the same anomaly on Friday as the previous period. From the period $P4$ to $P7$ there is a common trend without too much reduction in traffic, thus showing a small readjustment of the population to the routines of the pre-pandemic period. If we look at the daily traffic patterns in 6.7, we see how these remain constant throughout the pandemic period.

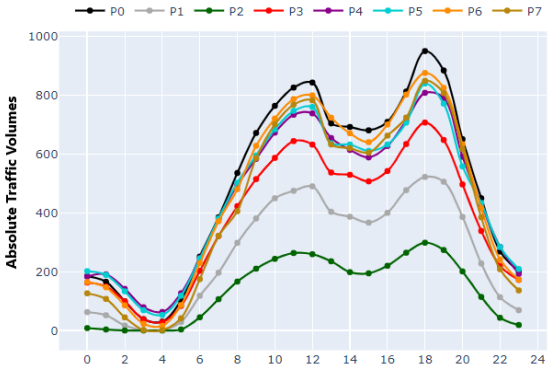

Figure 6.7: Daily traffic distribution for each of the eight pandemic periods.

6.4.2 Touristic Flows Analysis

This section analyzes in detail the impact of the COVID-19 pandemic on tourist flows. In order to carry out a statistical survey on the above impact, station 89 was chosen as it is located in a wide 4-lane road used as the main link between the city center and a long sandy beach, symbol of the city. According to our analyzes, the vehicular flow in 2020 is reduced by 20.69 %, a value in line with those recorded by the authorities at the regional level. Therefore by profiling the traffic in the different it is possible to make good deductions on the behavior of citizens and the different types of traffic. What has been evaluated so far has a general value, but it is possible to make other considerations of a more geographically precise nature by analyzing the traffic in a single sensor or a cluster of sensors. The figure 6.8 shows the distribution of traffic in a particular selection of sensors correlated to the curve of tourist presences in the city during the course of the year. This distribution follows equally the trend of the various government measures, in particular during the summer it is seen that two stations differ greatly.

Station 66 has a lower than average behavior in those months. Yet this was found to be due to road-works. For station 89, however, the situation is different because the traffic flowing in this sensor correlates with the tourist presence curve, confirming this station as a strong indicator of touristic traffic.

6.5 Conclusion and future work

The research work presented in this article analyzes the variations and impact in the mobility model that occurred in 2020 in the city of Cagliari, which is a particularly touristy coastal city. The entire analysis is based on the traffic data acquired by inductive loops sparse in the city. During the first lockdown period, traffic volumes dropped by up to 76% compared to average traffic values of the previous four years, and then recovered in subsequent periods. Despite the presence of further restrictions at the end of the year in the vicinity of the second wave of infections, the

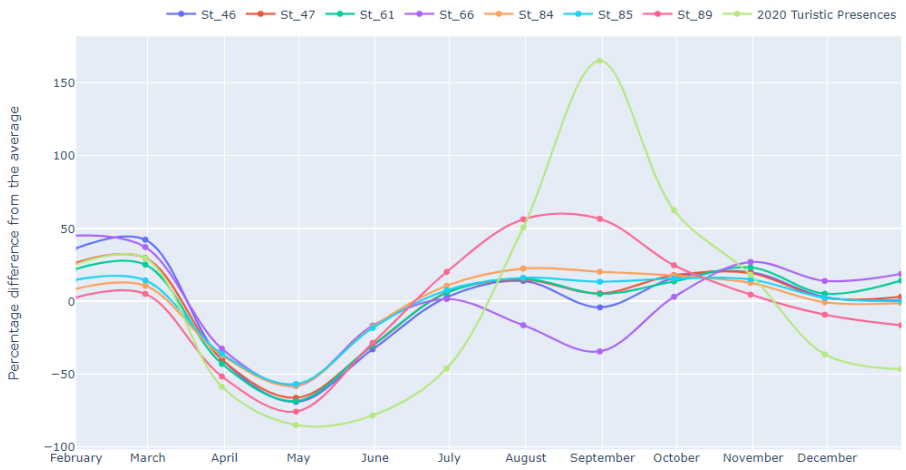

Figure 6.8: Comparison between 2020 traffic stations volumes and the touristic presences.

traffic did not react proportionally. The variation in traffic in the various stations was also analyzed, where different behaviors emerged, relating to different areas. Through this pilot study we could put real data to the test, unveiling a mix of expected results (volume drops) and less obvious results (change in traffic patters). We could also evaluate the specific effect that the pandemic had on tourism, given that traffic volumes did not return to normal during the summer, in spite of more relaxed restrictions.

The research work presented in this chapter is aimed at analyzing the variations in vehicular flow within the city of Cagliari, with particular attention to the effects that the pandemic has had on the volumes and models of urban and tourist mobility. The whole analysis is based on objective data from a network of 98 sensors distributed throughout the urban scenario, showing a reduction of 76% compared to the average traffic of the last 4 years. With a particular event coinciding with the second, a period in which government restrictions replicated themselves in the same way as those of the first period, however, the response of citizens was quite different since the traffic flows are not reduced in the same way, but in a less drastic. The variation in traffic in the various stations was also analyzed, where different behaviors emerged, relating to different areas. Through this pilot study we were able to test real data, revealing a mix of expected results (volume drops) and less obvious results (change in traffic patterns). We could also assess the specific effect the pandemic has had on tourism, as traffic volumes have not returned to normal during the summer, despite looser restrictions.

Chapter 7

Conclusions and future works

In this thesis it has been presented a way to understand people's mobility in Smart Cities. The solution developed in this work is based on the Wi-Fi probe request analysis, but without considering the MAC address of the devices.

In fact, after the introduction of the G.D.P.R. n. 2016/679 in EU the MAC address it is considered as a Personal Identification Information, this is the reason why, nowadays, almost all devices in the market have implemented MAC address randomization. Such process has been introduced by smartphone manufactures in order to protect users privacy during the smartphone's Access Point discovery, it requires that each probe request or probe request burst has a different MAC address. It is quit simple to understand that such mechanism has completely destroyed all the probe request based techniques and frameworks developed to study the people's mobility, transforming the probe request from source of data to "digital junk".

The aim of this study was to develop a complete cloud-based IoT system able to understand how devices generate Probe Request in order to detect how many devices are nearby the system and at the same time track them. The basic idea to do this is to generate fingerprint based on the Information Elements of the Probe Request frame, combined with the information of the growth speed of sequence number in a burst of probe request.

Another important point was to apply those technologies in some real-world scenarios like squares, university rooms or buses. The insights discovered in those application fields were interesting and promising a good research result, in particular for the public transportation field where understanding human mobility is crucial. This is the reason why, a corollary of this study, I believe that Wi-Fi will be still an important source of data, useful for mobility understanding even if the randomization process has introduced some complications. Indeed, through the result of this Thesis is still possible use Wi-Fi Probe Request for understanding human mobility while respecting privacy.

7.1 List of publications related to the Thesis

- M. Uras, R. Cossu, L. Atzori, "PmA: a solution for people mobility monitoring and analysis based on WiFi probes", 2019 4th International Conference on Smart and Sustainable Technologies (SpliTech), Split, 2019.

- M. Uras, R. Cossu, E. Ferrara, A. Liotta, L. Atzori, "PmA: a real-world system for people mobility monitoring and analysis based on Wi-Fi probes", Journal of Cleaner Production, volume 270, 2020.

- M. Uras, R. Cossu, E. Ferrara, O. Bagdasar, A. Liotta, L. Atzori, "Wifi probes sniffing: an artificial intelligence based approach for mac addresses derandomization", 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Virtual, 2020

- M. Uras, E. Ferrara, R. Cossu, A. Liotta, L. Atzori , "MAC address derandomization for WiFi device counting: combining temporal- and content-based fingerprints", Computer Networks - The International Journal of Computer and Telecommunications Networking, 2021 (Under Review)

- E. Ferrara, M. Uras, O. Bagdasar, A. Liotta, L. Atzori, "Mobility Analysis during the 2020 Pandemic in a Touristic city: the Case of Cagliari", IEEE IoT Vertical and Topical Summit for Tourism (IoTVTST), Virtual, 2020

List of Figures

List of Tables

Bibliography

[ABC+20] Francesco Aletta, Stefano Brinchi, Stefano Carrese, Andrea Gemma, Claudia Guattari, Livia Mannini, and Sergio Maria Patella. Analysing urban traffic volumes and mapping noise emissions in rome (italy) in the context of containment measures for the covid-19 disease. *Noise Mapping*, 7(1):114–122, 2020.

[ABKS99] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[ANL+18] Javier Andión, José M Navarro, Gregorio López, Manuel Álvarez-Campana, and Juan C Dueñas. Smart behavioral analytics over a low-cost iot wi-fi tracking real deployment. *Wireless Communications and Mobile Computing*, 2018, 2018.

[Aro77] Eugene A Aronson. Location errors in time of arrival (toa) and time difference of arrival (tdoa) systems. Technical report, Sandia Labs., Albuquerque, N. Mex.(USA), 1977.

[Bal20] José M. Baldasano. Covid-19 lockdown effects on air quality by no2 in the cities of barcelona and madrid (spain). *Science of The Total Environment*, 741:140353, 2020.

[BBGO08] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 116–127, 2008.

[BBQL13] Bram Bonné, Arno Barzan, Peter Quax, and Wim Lamotte. Wifipi: Involuntary tracking of visitors at mass events. In *2013 IEEE 14th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 1–6. IEEE, 2013.

[BBS17] Suvankar Barai, Debajyoti Biswas, and Buddhadeb Sau. Estimate distance measurement using nodemcu esp8266 based on rssi technique.

In *2017 IEEE Conference on Antenna Measurements & Applications (CAMA)*, pages 170–173. IEEE, 2017.

[BHP07] Genevieve Bartlett, John Heidemann, and Christos Papadopoulos. Understanding passive and active service discovery. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 57–70, 2007.

[BP+00] Paramvir Bahl, Venkata N Padmanabhan, et al. Radar: An in-building rf-based user location and tracking system. In *IEEE infocom*, volume 2, pages 775–784. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 2000.

[CDBvS18] Cristian Chilipirea, Ciprian Dobre, Mitra Baratchi, and Maarten van Steen. Identifying movements in noisy crowd analytics data. In *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, pages 161–166. IEEE, 2018.

[CMS13] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[CSMC04] Ka Wai Cheung, Hing-Cheung So, W-K Ma, and Yiu-Tong Chan. Least squares algorithms for time-of-arrival-based mobile location. *IEEE Transactions on Signal Processing*, 52(4):1121–1130, 2004.

[CSZ+15] Wu Chen, Jianhua Sun, Lu Zhang, Xiang Liu, and Liang Hong. An implementation of ieee 1588 protocol for ieee 802.11 wlan. *Wireless Networks*, 21, 08 2015.

[CWW+20] Yizhe Chen, Yichun Wang, Hui Wang, Zhili Hu, and Lin Hua. Controlling urban traffic-one of the useful methods to ensure safety in wuhan based on covid-19 outbreak. *Safety Science*, 131:104938, 2020.

[De 20] Jonas De Vos. The effect of covid-19 and subsequent social distancing on travel behavior. *Transportation Research Interdisciplinary Perspectives*, 5:100121, 2020.

[DLMS16] Adriano Di Luzio, Alessandro Mei, and Julinda 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. IEEE, 2016.

[DPČ19] Ante Dagelić, Toni Perković, and Mario 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. IEEE, 2019.

[DPS+16] Merkebe Getachew Demissie, Santi Phithakkitnukoon, Titipat Sukhvibul, Francisco Antunes, Rui Gomes, and Carlos Bento. Inferring passenger travel demand to improve urban mobility in developing countries using cell phone data: a case study of senegal. *IEEE Transactions on Intelligent Transportation Systems*, 17(9):2466–2478, 2016.

[DP9] 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.

[DSF+20] Guilherme Dantas, Bruno Siciliano, Bruno Boscaro França, Cleyton M. da Silva, and Graciela Arbilla. The impact of covid-19 partial lockdown on the air quality of the city of rio de janeiro, brazil. *Science of The Total Environment*, 729:139085, 2020.

[EKS+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, volume 96, pages 226–231, 1996.

[FLN+18] Enrico Ferrara, Antonio Liotta, Maryleen Ndubuaku, Laura Erhan, Daniel D Giusto, Miles Richardson, David Sheffield, and Kirsten McEwan. A demographic analysis of urban nature utilization. In *2018 10th Computer Science and Electronic Engineering (CEEC) (CEEC'18)*, Colchester, Essex, United Kingdom (Great Britain), September 2018.

[FMT+06a] Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoe, J Van Randwyk, and Douglas Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX Security Symposium*, volume 3, pages 16–89, 2006.

[FMT+06b] Jason Franklin, Damon Mccoy, Parisa Tabriz, Vicentiu Neagoe, Jamie Randwyk, and Douglas Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX Advanced Computing Systems Association Security Symposium*, volume 3, pages 16–89, 2006.

[Fre15] Julien Freudiger. How talkative is your mobile device?: an experimental study of wi-fi probe requests. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, page 8. ACM, 2015.

[Fri46] Harald T Friis. A note on a simple transmission formula. *Proceedings of the IRE*, 34(5):254–256, 1946.

[GC18] A. Guillén-Pérez and M. D. Cano Baños. A wifi-based method to count and locate pedestrians in urban traffic scenarios. In *2018 14th WiMob)*, pages 123–130, 2018.

[GLRC19a] P Galluzzi, Edoardo Longo, Alessandro Enrico Cesare Redondi, and Matteo Cesana. Occupancy estimation using low-cost wi-fi sniffers. *ArXiv*, abs/1905.06809, 2019.

[GLRC19b] Paolo Galluzzi, Edoardo Longo, Alessandro EC Redondi, and Matteo Cesana. Occupancy estimation using low-cost wi-fi sniffers. *arXiv preprint arXiv:1905.06809*, 2019.

[Goo13] Dan Goodin. No, this isn'ta scene from minority report. this trash can is stalking you. *Ars Technica*, 2013.

[Gru] E. Grumbach. iwlwi: mvm: support random mac address for scanning. Linux commit effd05ac479b.

[GWY+15] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y Yen, Runhe Huang, and Xingshe Zhou. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM Computing Surveys (CSUR)*, 48(1):7, 2015.

[HHK20] Veronika Harantová, Ambróz Hájnik, and Alica Kalašová. Comparison of the flow rate and speed of vehicles on a representative road section before and after the implementation of measures in connection with covid-19. *Sustainability*, 12(17), 2020.

[HPL18] He-Yen Hsieh, Setya Widyawan Prakosa, and Jenq-Shiou Leu. Towards the implementation of recurrent neural network schemes for wifi fingerprint-based indoor positioning. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, 2018.

[JY20] Wenrui Huang Jieya Yang. Effects of population density and traffic flow on covid-19 disasters in florida. *Advancements in civil engineering technology*, 4(2), 2020.

[KBCP11] Dmytro Karamshuk, Chiara Boldrini, Marco Conti, and Andrea Passarella. Human mobility models for opportunistic networks. *IEEE Communications Magazine*, 49(12):157–165, 2011.

[KBI+20] Aiymgul Kerimray, Nassiba Baimatova, Olga P. Ibragimova, Bauyrzhan Bukenov, Bulat Kenessov, Pavel Plotitsyn, and Ferhat

Karaca. Assessing air quality changes in large cities during covid-19 lockdowns: The impacts of traffic-free urban conditions in almaty, kazakhstan. *Science of The Total Environment*, 730:139179, 2020.

[KJ20] G Jaspher Willsie Kathrine and C Willson Joseph. Attacks, vulnerabilities, and their countermeasures in wireless sensor networks. In *Deep Learning Strategies for Security Enhancement in Wireless Sensor Networks*, pages 134–154. IGI Global, 2020.

[KJBK15] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 269–282. ACM, 2015.

[KMSY20] Christos Katrakazas, Eva Michelaraki, Marios Sekadakis, and George Yannis. A descriptive analysis of the effect of the covid-19 pandemic on driving behavior and road safety. *Transportation Research Interdisciplinary Perspectives*, 7:100186, 2020.

[Kna19] Stefan Knauth. Study and evaluation of selected rssi-based positioning algorithms. In *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*, pages 147–167. Elsevier, 2019.

[KO17] Abdullah Kurkcu and Kaan Ozbay. Estimating pedestrian densities, wait times, and flows with wi-fi and bluetooth sensors. *Transportation Research Record: Journal of the Transportation Research Board*, (2644):72–82, 2017.

[Kum20] TM Vinod Kumar. Smart living for smart cities. In *Smart Living for Smart Cities*, pages 3–70. Springer, 2020.

[LMM18] Asad Lesani and Luis Miranda-Moreno. Development and testing of a real-time wifi-bluetooth system for pedestrian network monitoring, classification, and data extrapolation. *IEEE Transactions on Intelligent Transportation Systems*, 20(4):1484–1496, 2018.

[LO17] Trung-Kien Le and Nobutaka Ono. Refinement of time-difference-of-arrival measurements via rank properties in two-dimensional space. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1971–1975. IEEE, 2017.

[MCRV16] Célestin Matte, Mathieu Cunche, Franck Rousseau, and Mathy Vanhoef. Defeating mac address randomization through timing attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 15–20, 2016.

[MMD+17] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik Rye, and Dane Brown. A study of mac address randomization in mobile devices and when it fails. *Proceedings on Privacy Enhancing Technologies*, 2017, 03 2017.

[MVFB10] Eladio Martin, Oriol Vinyals, Gerald Friedland, and Ruzena Bajcsy. Precise indoor localization using smart phones. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 787–790. ACM, 2010.

[Nas06] EN Nasibov. An alternative fuzzy-hierarchical approach to cluster analysis. In *Proceedings 7th International Conference on Application of Fuzzy Systems and Soft Computing, Germany*, pages 113–123, 2006.

[NPP+20] Michele Nitti, Francesca Pinna, Lucia Pintor, Virginia Pilloni, and Benedetto Barabino. iabacus: A wi-fi-based automatic bus passenger counting system. *Energies*, 13:1446, 03 2020.

[OBT+20] Anas Otmani, Abdelfettah Benchrif, Mounia Tahri, Moussa Bounakhla, El Mahjoub Chakir, Mohammed El Bouch, and M'hamed Krombi. Impact of covid-19 lockdown on pm10, so2 and no2 concentrations in salé city (morocco). *Science of The Total Environment*, 735:139541, 2020.

[P+99] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[PCB+17] Andreea-Cristina Petre, Cristian Chilipirea, Mitra Baratchi, Ciprian Dobre, and Maarten van Steen. Wifi tracking of pedestrian behavior. In *Smart Sensors Networks*, pages 309–337. Elsevier, 2017.

[PCG+16a] Francesco Potortì, Antonino Crivello, Michele Girolami, Emilia Traficante, and Paolo Barsocchi. Wi-fi probes as digital crumbs for crowd localisation. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2016.

[PCG+16b] 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.

[PCG+18] Francesco Potortì, Antonino Crivello, Michele Girolami, Paolo Barsocchi, and Emilia Traficante. Localising crowds through wi-fi probes. *Ad Hoc Networks*, 75:87–97, 2018.

[PMLC16] Marco Passafiume, Stefano Maddio, Matteo Lucarelli, and Alessandro Cidronali. An enhanced triangulation algorithm for a distributed rssi-doa positioning system. In *2016 European Radar Conference (EuRAD)*, pages 185–188. IEEE, 2016.

[PR10] Ilango Purushothaman and Sumit Roy. Fastscan: a handoff scheme for voice over ieee 802.11 wlans. *Wireless Networks*, 16(7):2049–2063, 2010.

[RC18] Alessandro EC Redondi and Matteo Cesana. Building up knowledge through passive wifi probes. *Computer Communications*, 117:1–12, 2018.

[RGPN20] Miguel Ribeiro, Bernardo Galvão, Catia Prandi, and Nuno Nunes. Passive wi-fi monitoring in public transport: A case study in the madeira island, 2020.

[RNNS20] Miguel Ribeiro, N. Nunes, Valentina Nisi, and J. Schöning. Passive wi-fi monitoring in the wild: a long-term study across multiple location typologies. *Personal and Ubiquitous Computing*, pages 1 – 15, 2020.

[RRBP+14] Antonio J Ruiz-Ruiz, Henrik Blunck, Thor S Prentow, Allan Stisen, and Mikkel B Kjærgaard. Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning. In *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 130–138. IEEE, 2014.

[SCJ19] F. Shi, K. Chetty, and S. Julier. Passive activity classification using just wifi probe response signals. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–6, April 2019.

[SN14] Supriya S. Sawwashere and Sonali U. Nimbhorkar. Survey of rts-cts attacks in wireless network. In *2014 Fourth International Conference on Communication Systems and Network Technologies*, pages 752–755, 2014.

[SNYK20] Naeimeh Soltanieh, Yaser Norouzi, Yang Yang, and Nemai Chandra Karmakar. A review of radio frequency fingerprinting techniques. *IEEE Journal of Radio Frequency Identification*, 4(3):222–233, 2020.

[SWM14] Lorenz Schauer, Martin Werner, and Philipp Marcus. Estimating crowd densities and pedestrian flows using wi-fi and bluetooth. In *MOBIQUITOUS '14 Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 171–177. IEEE, 2014.

[SWS+20] N. Suraweera, A. Winter, J. Sorensen, S. Li, M. Johnson, I. B. Collings, S. V. Hanly, W. Ni, and M. Hedley. Passive through-wall counting of people walking using wifi beamforming reports. *IEEE Systems Journal*, pages 1–7, 2020.

[SZA+20] Shubham Sharma, Mengyuan Zhang, Anshika, Jingsi Gao, Hongliang Zhang, and Sri Harsha Kota. Effect of restricted emissions during covid-19 on air quality in india. *Science of The Total Environment*, 728:138878, 2020.

[SZT08] Guowei Shen, Rudolf Zetik, and Reiner S Thoma. Performance comparison of toa and tdoa based location estimation algorithms in los environment. In *2008 5th Workshop on Positioning, Navigation and Communication*, pages 71–78. IEEE, 2008.

[TCR+20] Aurelio Tobías, Cristina Carnerero, Cristina Reche, Jordi Massagué, Marta Via, María Cruz Minguillón, Andrés Alastuey, and Xavier Querol. Changes in air quality during the lockdown in barcelona (spain) one month into the sars-cov-2 epidemic. *Science of The Total Environment*, 726:138540, 2020.

[TJMK18] Martin W Traunmueller, Nicholas Johnson, Awais Malik, and Constantine 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.

[Top10] D. Toppeta. The smart city vision: how innovation and ict can build smart, livable, sustainable cities. *Innov. Knowl. Found.*, page 1–9, 2010.

[UCA19a] M. Uras, R. Cossu, and L. Atzori. Pma: a solution for people mobility monitoring and analysis based on wifi probes. In *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–6, 2019.

[UCA19b] Marco Uras, Raimondo Cossu, and Luigi Atzori. Pma: a solution for people mobility monitoring and analysis based on wifi probes. In *2019 4th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–6. IEEE, 2019.

[UCF+20a] M. Uras, R. Cossu, E. Ferrara, O. Bagdasar, A. Liotta, and L. Atzori. Wifi probes sniffing: an artificial intelligence based approach for mac addresses de-randomization. In *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6, 2020.

[UCF+20b] Marco Uras, Raimondo Cossu, Enrico Ferrara, Ovidiu Bagdasar, Antonio Liotta, and Luigi Atzori. Wifi probes sniffing: an artificial intelligence based approach for mac addresses de-randomization. In *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 1–6. IEEE, 2020.

[UCF+20c] Marco Uras, Raimondo Cossu, Enrico Ferrara, Antonio Liotta, and Luigi Atzori. Pma: A real-world system for people mobility monitoring and analysis based on wi-fi probes. *Journal of Cleaner Production*, page 122084, 2020.

[UN18] UN. 68% of the World Population Projected to Live in Urban Areas by 2050, Says UN. `https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html`, 2018. [Online; accessed 20-Oct-2019].

[US07] Oktay Ureten and Nur Serinken. Wireless security through rf fingerprinting. *Canadian Journal of Electrical and Computer Engineering*, 32(1):27–33, 2007.

[VAB18] Saurabh Vaidya, Prashant Ambad, and Santosh Bhosle. Industry 4.0–a glimpse. *Procedia Manufacturing*, 20:233–238, 2018.

[VÇG+16] Edwin Vattapparamban, Bekir Sait Çiftler, Ismail Güvenç, Kemal Akkaya, and Abdullah 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. IEEE, 2016.

[VCS03] Vivek Vishnumurthy, Sangeeth Chandrakumar, and Emin Gun Sirer. Karma: A secure economic framework for peer-to-peer resource sharing. In *Workshop on Economics of Peer-to-peer Systems*, volume 35, 2003.

[VH10] Jiři Veselỳ and Petr Hubáček. The tdoa system topology optimization from signal source position error estimation point of view. *WSEAS Advances in Sensors, Signals and Materials*, pages 65–68, 2010.

[VMC+16] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms. In *Asia Conference on Computer and Communications Security '16*, 2016.

[w10] van hoef, m.: How mac address randomization works on windows 10 (2016). `http://www.mathyvanhoef.com/2016/`. Accessed: 27 Jan 2020.

[WCHZ18] Wei Wang, Jiayu Chen, Tianzhen Hong, and Na Zhu. Occupancy prediction through markov based feedback recurrent neural network (m-frnn) algorithm with wifi probe technology. *Building and Environment*, 138:160–170, 2018.

[WNDDZ06] Chris Wysopal, Lucas Nelson, Elfriede Dustin, and Dino Dai Zovi. *The art of software security testing: identifying software security flaws*. Pearson Education, 2006.

[WWLL20] Songhe Wang, Kangda Wei, Lei Lin, and Weizi Li. Spatial-temporal analysis of covid-19's impact on human mobility: the case of the united states, 2020.

[XSK+13] Zhuliang Xu, Kumbesan Sandrasegaran, Xiaoying Kong, Xinning Zhu, B Hu, J Zhao, and C Lin. Pedestrain monitoring system using wi-fi technology and rssi based localization. *International Journal of Wireless & Mobile Networks*, 2013.

[XZL+14] Wei Xi, Jizhong Zhao, Xiang-Yang Li, Kun Zhao, Shaojie Tang, Xue Liu, and Zhiping Jiang. Electronic frog eye: Counting crowd using wifi. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 361–369. IEEE, 2014.

[YBC05] Chin-Lung Yang, Saurabh Bagchi, and William J Chappell. Location tracking with directional antennas in wireless sensor networks. In *Microwave Symposium Digest, 2005 IEEE MTT-S International*, pages 131–134, 2005.