UNIVERSITY OF PADOVA

DEPARTMENT OF INFORMATION ENGINEERING
Ph.D. Course in Information Engineering
Information and Communication Science and Technologies Curriculum
XXXIV series

Wireless Communications and Sensing – Opportunities and Challenges for Environment Aware Networks

Ph.D. Candidate
Francesca Meneghello

Ph.D. Supervisor
Professor Michele Rossi

Ph.D. Coordinator
Professor Andrea Neviani

Academic Year
2020–2021

To *mamma* and *papà*,
for giving me solid roots
and strong wings, to stay, to fly.
I owe everything I am to you.

We must have perseverance and, above all, confidence in ourselves.
We must believe that we are gifted for something and that this thing must be attained.
Marie Skłodowska Curie

Abstract

The widespread deployment of wireless communication networks in both indoor and outdoor spaces opens the door to a variety of new opportunities and, at the same time, poses several technical challenges. In this scenario, *environment awareness* represents both a new appealing side-functionality offered by wireless systems, and a promising way to increase the quality of the services provided to the users.

As for the former, wireless devices are intrinsically capable of sensing the propagation environment. The estimation of the wireless channel parameters is continuously performed at the receiver end of the communication link to properly decode the radio signals collected by the antennas. This feature can be effectively exploited to gain knowledge of the surrounding space by detecting the presence of obstacles through the analysis of the multi-path propagation effect. The potential of this application resides in the possibility to realize indoor monitoring systems leveraging already deployed devices – such as Wi-Fi access points – without the need for additional and possibly expensive hardware. The scientific community is extensively studying this opportunity, developing algorithms that target different sensing applications such as human-presence detection and activity recognition among others. Despite the fact that several proofs of concept are available in the literature, robust algorithms that effectively work in changing environmental conditions are not yet available and require further investigations. The envisioned applications recently attracted the interest of the Institute of Electrical and Electronics Engineers (IEEE) that established a task force to develop a new version of the Wi-Fi standard, named IEEE 802.11bf. The project started in September 2020 and devices implementing this standard are expected to be commercialized by 2024. The IEEE 802.11bf will enable the joint provisioning of the communication and the sensing services to the users by the introduction of some modifications to the physical and medium access control layers of the protocol stack to fulfill the slightly different requirements of the two services.

Moving to the second aspect, information about the context where the wireless devices are operating enables reasoned management of the available network resources, guaranteeing an adequate quality of experience to the connected users. This is especially appealing when considering the emerging fifth generation – and beyond – systems, where the network will provide both connectivity and computation support for static and moving terminals, e.g., connected vehicles. The European Telecommunications Standards Institute (ETSI) standardization body is actively working on this paradigm change – named multi-access edge computing (MEC) – through the standardization of interoperable architectures and their integration with next generations mobile networks. In this context, the network entities need to properly handle the handover of the computation service provisioning to avoid critical discontinuity issues. The exploitation of environmental information in such network management processes provides a clear benefit from both a network and a user perspective, allowing the operators to optimize the energy consumption

while assuring a timely fulfillment of the customers' requests. Research in this field is still open and the scientific community is working on the identification of proper policies to decide when and where to perform the handover of the computation services to minimize the energy network consumption while guaranteeing adequate quality of service to the users.

This thesis makes substantial contributions to the development of new-generation wireless networks, since in the future environmental awareness will be one of the key enablers. The *interplay* between wireless communications and sensing is discussed by detailing the design and the implementation – together with the assessment of the performance – of brand-new algorithms for environment aware networks. Throughout the thesis, in-depth theoretical analysis is combined with advanced practical implementations and simulative evaluations. Machine and deep learning techniques are effectively exploited to extract relevant information from the huge amount of data required to enable the integration of communication and sensing services into future wireless networks. The strength of the detailed approaches is the combination of learning techniques with mathematical models that provide solid foundations for the integration of sensing and communication services. Both *communication-assisted sensing* and *sensing-assisted communication* applications are presented through practical uses cases targeting cutting-edge technologies. Next-generation Wi-Fi and cellular networks are considered as examples of respectively the former and the latter case, showing the potential of the envisioned environment aware paradigm in both indoor and outdoor scenarios.

Contents

Listing of figures

Listing of tables

Listing of acronyms

Symbols

3GPP Third Generation Partnership Project

5G fifth generation

6G sixth generation

A

AoA angle of arrival

AP access point

C

CFO carrier frequency offset

CFR channel frequency response

CNN convolutional neural network

COTS commercial-off-the-shelf

CSI channel state information

D

DL downlink

DNN deep neural network

E

EHT extremely high throughput

eMBB enhanced mobile broadband

eNB evolved node base

ETSI European Telecommunications Standards Institute

F

FMCW frequency-modulated continuous-wave

G

GNSS global navigation satellite system

GRU gated recurrent unit

GSM global system for mobile communications

H

HAR human activity recognition

HDBSCAN hierarchical density-based spatial clustering of applications with noise

HE high-efficiency

HT high throughput

I

ID identifier

IEEE Institute of Electrical and Electronics Engineers

IFFT inverse fast Fourier transform

IMT International Mobile Telecommunications

IMU inertial measurement unit

IoT Internet of things

IoV Internet of vehicles

ISI inter-stream interference

ITU International Telecommunications Union

IUI inter-user interference

L

LTE long term evolution of UMTS

LTF long training field

M

MAC medium access control

MC Markov chain

MCS modulation and coding scheme

MEC multi-access edge computing

MEH MEC host

MIMO multi-input, multi-output

MIP mixed integer programming

ML machine learning

mMTC massive machine type communications

mmWave millimeter wave

MPC model predictive control

MU-MIMO multi-user multi-input, multi-output

N

NDP null data packet

NG60 next generation 60 GHz

NLoS non line of sight

NN neural network

NR new radio

NR-U new radio-based access to unlicensed spectrum

O

OFDM orthogonal frequency-division multiplexing

OFDMA orthogonal frequency-division multiple access

P

PA phase ambiguity

PDD packet detection delay

PHY physical

PL path loss

PLL phase-locked loop

PPO phase-locked loop offset

PV photovoltaic panel

Q

QoE quality of experience

QoS quality of service

R

RFP radio fingerprinting

RNN recurrent neural network

RSS received signal strength

S

SBC single board computer

SC-FDMA single carrier - frequency division multiple access

SDR software-defined radio

SFO sampling frequency offset

SNR signal-to-noise ratio

STA station

SU-MIMO single-user multi-input, multi-output

SUMO simulation of urban mobility

SVD singular value decomposition

SVM support vector machine

T

TDoA time difference of arrival

ToA time of arrival

U

UE user equipment

UL uplink

UMTS universal mobile telecommunications system

URLLC ultra-reliable and low latency communications

USRP universal software radio peripheral

V

VHT very high throughput

VM virtual machine

W

WLAN wireless local area network

1 Introduction

The context is of paramount importance in our lives. As humans, we continuously capture side information from the environment and use the data to gain knowledge of the context. Moreover, we adapt our behavior to the social circumstances, and approach problems in different ways depending on the people the people we are surrounded by. This bidirectional *environment awareness* concept is gradually being integrated into wireless communication networks. On the one hand, the propagation patterns of the wireless transmissions can serve as a bridge to sense the environment, as obstacles in the scene cause characteristic signals' reflections. On the other hand, environmental knowledge can be exploited within the network-management processes to intelligently allocate resources to the connected users, saving energy and providing adequate quality of experience.

The former aspect – referred to as *communication-assisted sensing* – is gaining momentum given the high availability of signals of opportunity that can be exploited for sensing purposes in both indoor and outdoor spaces. In particular, the widespread adoption of the Wi-Fi technology – with more than 16 billion of Wi-Fi-enabled devices in use around the globe [1] – opens up new opportunities for device-free and unobtrusive indoor monitoring solutions.

As for the latter – indicated with *sensing-assisted communication* – cellular networks operators are already benefitting from side knowledge to adopt strategic policies in the service delivery and are expected to increasingly leverage such data with the advent of future generations networks. This is key to enabling new applications such as autonomous driving and augmented/virtual/mixed reality, that require reliable and timely network-service delivery.

Overall, several proofs of concept confirm the benefits that the communication and the sensing services grant to each other and brought standardization entities to move toward their joint implementation on future wireless networks. This is enabled by the development of new physical (PHY) and medium access control (MAC) layers protocols and the integration of artificial intelligence into the networks. In particular, machine learning (ML) techniques play an important role

in the statistical analysis of the huge amount of data collected for this purpose and the definition of data-driven sensing and communication services. The integration of learning strategies in next-generation networks will bring a significant paradigm shift, moving from reactive actions – in response to the occurrence of an event – to proactive operations – based on side knowledge gained from the data. This allows creating more general frameworks capable to adapt to changing environmental conditions, as is the case when working with wireless networks. Such an aspect is of paramount importance for the delivery of brand-new services that require the network to guarantee adequate quality of service – in terms of transmission delay and service continuity – to connected users moving within the network area. Learning frameworks can help in different functions, such as network resource management, environmental monitoring, mobility, and traffic prediction. They can be used to completely replace existing rule-based approaches or to augment the functionalities of already implemented algorithms. Depending on the specific application, ML techniques can be integrated at different network levels, starting from the end devices, the access points, the network infrastructure, up to the applications in the cloud of the network, enabling new opportunities for next generation networks [2].

The work presented in this thesis makes substantial contributions to the integration and exploitation of environmental awareness in wireless networks. Communication-assisted sensing in Wi-Fi networks and sensing-assisted communication for cellular networks, are considered in Part I and Part II respectively, focusing on practical use cases to show the potential of this foreseen holistic view of communication and sensing services. Two sensing applications are considered in Part I. At first, we design and implement a novel robust algorithm for environment- and person-independent human activity recognition through radio waves, departing from the approaches presented in the literature which still do not integrate such generalization capability. Next, we put forward the first physical layer fingerprinting algorithm for multi-user multi-input, multi-output (MIMO) devices, allowing the identification of the terminal from the information sent in clear text for the beam formation. These two aspects of sensing through radio waves are of strategic importance for next-generation Wi-Fi networks. As for the former, the Institute of Electrical and Electronics Engineers (IEEE) is currently working on the standardization of a new version of the standard – IEEE 802.11bf – for the integration of environmental and human sensing functionalities into commercial Wi-Fi devices. The latter is a key enabler for service provisioning in crowded networks where network administrators need to ensure the fair coexistence of devices implementing different standards – such as Wi-Fi and 3GPP new radio-based access to unlicensed spectrum (NR-U) – while using the same portion of the radio spectrum. Part II of the thesis starts with the design of a mobility prediction algorithm for vehicles moving within a city while being connected to a cellular network empowered with multi-access edge computing (MEC) functionalities. The framework is next used in combination with mathematical optimization methods for the definition of two network resource management strategies with the objective of guaranteeing computation service continuity to the users when changing the serving base station. This approach aims at minimizing the energy expenditure at the network side while being constrained on specific users' service requirements in terms of maximum allowed completion delay. Common to all the approaches presented in the thesis is the combination of mathematical analysis

with advanced ML techniques. This allows both the leveraging of the established models describing wireless communications and the exploitation of the huge amount of data that can be gathered from the environment to increase the quality of service – communication and sensing – offered to the connected users.

Before delving into the technical details, we give an overview of the two aspects tackled in Part I and Part II of the thesis, in Section 1.1 and Section 1.2 respectively. For the sake of completeness, we also briefly introduce Wi-Fi networks in Section 1.1.1 and cellular systems in Section 1.2.1.

1.1 Communication-assisted sensing in Wi-Fi networks

The inherent ability of the Wi-Fi devices to sense the propagation environment, has led researchers to investigate the possibility of using Wi-Fi access points (APs) as environmental sensors. In fact, Wi-Fi devices continuously estimate the parameters of the propagation environment, obtaining information on how radio waves propagate from the transmitter to the receiver end. This information, called channel frequency response (CFR) or channel state information (CSI), is used to properly decode the wirelessly exchanged data, compensating for channel impairments that otherwise would cause communication errors.* The main observation underpinning the use of Wi-Fi APs as environment sensors is that the CFR naturally encodes the presence of obstacles – e.g., walls, furniture, people – that are located within the propagation environments. Moreover, the CFR is affected by hardware imperfections of both the transmitter and the receiver Wi-Fi devices. Therefore, this information can serve as a bridge to sense the indoor environment where the communication is ongoing, revealing the presence of obstacles in the scene and identifying, possibly malicious, Wi-Fi devices transmitting in range. Such sensing capabilities allow developing monitoring systems by reusing already deployed Wi-Fi devices, without the need for additional and expensive ad-hoc sensors such as RADARs/LIDARs for environmental sensing or radio intruder detectors for wireless channel access control.

The first proof of concept showing that Wi-Fi transmissions – in the 2.4 GHz and 5 GHz bands – can be used as signals of opportunity to perform environmental sensing was presented in 2013 by Adib and Katabi [3]. The manuscript attracted the attention of several research groups that started working on this newly opened research direction. A clear benefit of this approach is that the sensing is performed without the acquisition of images of the scene – as cameras do – thus reducing privacy issues. Different algorithms were proposed over the years, addressing various aspects of environmental sensing including subject identification and localization, activity and gesture recognition, and even estimation of biological signals at distance, such as the respiration rate or the heartbeat [4]. However, the effective implementation of such sensing solutions poses some limitations associated with the achievable sensing resolution on below 6 GHz bands and the robustness to changing environmental conditions. To tackle the first issue – low sensing resolution – some recent works propose to exploit IEEE 802.11ad devices operating on the 60 GHz band, obtaining high resolution solutions for different tasks, such as person localization [5] and

*In the rest of the thesis, the terms CSI and CFR will be used interchangeably.

hand gestures detection [6], at the cost of lower operational range. As for the second challenge (robustness to changing environment), in Chapter 2 of this thesis we present how to address it – for the specific task of human activity recognition – through a learning-based algorithm that, once trained, can be used to infer the activity in a completely different scenario, i.e., in a different day, room and with different persons than those in the training examples. This contribution substantially departs from the literature in the field where algorithms cannot generalize to different situations without being retrained on new data.

The exploitation of the CFR as a means to perform wireless channel access monitoring is instead investigated in Chapter 3 of the thesis. We leverage the transmitter hardware inaccuracies encoded in the CFR to devise a learning-based PHY layer fingerprinting algorithm of Wi-Fi devices. In this case, the target of the sensing process is the identification of the Wi-Fi device that is transmitting data. PHY layer fingerprinting algorithms are expected to serve as a basis for reliable authentication techniques, reducing the latency experienced when using cryptography-based approaches. Existing solutions for radio fingerprinting (RFP) rely on software-defined radio (SDR) devices that monitor the radio channel and extract characteristic features from the ongoing transmissions. However, these devices require expert knowledge and ad-hoc deployments. We instead propose to leverage existing Wi-Fi devices already present in most residential and workspaces, introducing a new and high potential strategy to perform RFP. More than that, our work represents the first effort toward the implementation of a RFP for devices using the multi-user multi-input, multi-output (MU-MIMO) transmission mode. Being MU-MIMO one of the key enablers for next-generation wireless networks, this work is expected to have a high impact on the scientific and industrial community.

The performance of the two developed *communication-assisted sensing* algorithms is assessed through commercial-off-the-shelf (COTS) Wi-Fi devices implementing the IEEE 802.11ac standard. The implementation of the experimental testbeds posed several challenges that come from the fact that Wi-Fi manufacturers do not provide the CSI out of the shelf. The CFR is considered useless for the final user that only requires connectivity support. This leads researchers to apply reverse engineering approaches to effectively extract the CFR from specific commercial Wi-Fi devices. For several years the scientific community relied on two software working with IEEE 802.11n devices. Next, in December 2020, a new powerful tool was released allowing one to obtain the CFR from IEEE 802.11ac devices [7]. The work presented in Chapter 2 relies on this tool. As for the work in Chapter 3 we instead applied reverse engineering by ourselves to obtain the CFR from feedback information provided by a Wi-Fi receiver to a Wi-Fi transmitter for enabling MIMO communications.

1.1.1 Preliminaries on Wi-Fi networks

In 1997, following a seven-year development process, the IEEE released the first standard of the 802.11 series. The document specified PHY and MAC layers characteristics for wireless local area networks (WLANs) operating on the 2.4 GHz band of the *unlicensed* radio spectrum. Next, in 1999, a group of telecommunication companies founded the Wi-Fi alliance to ensure interoperability

among devices implementing the IEEE 802.11 standard [8]. Specifically, the Wi-Fi alliance owns the *Wi-Fi* trademark and provides certification for devices that pass specific interoperability tests. While at its beginning the Wi-Fi technology was not able to compete with the Ethernet one in terms of data transmission rate (2 Mb/s versus 100 Mb/s), over the years new standard releases made Wi-Fi a very successful and widespread communication technology, offloading more than the 60% of the Internet traffic in 2021 [9]. With the IEEE 802.11a/b/g standards (1999-2003), data rates of 54 Mb/s were achieved thanks to the released of the new 5 GHz spectrum band and the use of the orthogonal frequency-division multiplexing (OFDM) modulation over 20 MHz channels with sub-channels of 312.5 KHz. OFDM entails the parallel transmission of groups of bits through the different sub-channels thus increasing the network throughput. In 2009, the IEEE 802.11n high throughput (HT) release – certified as Wi-Fi 4 – introduced the MIMO technology that exploits arrays of antennas to transmit multiple data streams to the user in a parallel fashion. The MIMO technology has been refined in 2013 with the IEEE 802.11ac very high throughput (VHT) release – certified as Wi-Fi 5 – with the introduction of downlink (DL) MU-MIMO that enables the AP to simultaneously transmit data streams to multiple users. These technological innovations came together with the expansion of the OFDM modulation channel width to 40 MHz (IEEE 802.11n) and 160 MHz (IEEE 802.11ac). Overall, devices implementing these two standards provide data rates of up to 600 Mb/s and 7 Gb/s respectively.

Nowadays, the majority of the Wi-Fi devices implement the IEEE 802.11n/ac standards. However, as a natural consequence of the digitalization of the society, current Wi-Fi networks will soon no longer be able to support the increasing demands for connectivity. The introduction of the Internet of things (IoT) paradigm brought to the market a variety of connected devices for the more diverse applications. Moreover, new emerging services are more network resources hungry than already supported ones. For instance, the delivery of high-quality 4K and 8K videos – characterized by uncompressed data rates of up to 20 Gb/s – requires the availability of high throughput links that, together with low-latency, is also a key enabler for augmented, virtual and mixed reality applications. All these factors will rapidly lead to the saturation of the available license-exempt radio spectrum, thus causing the reduction of the quality of service offered to Wi-Fi users. To prevent this from happening, the research community is designing ameliorations to the IEEE 802.11 standards to efficiently exploit the spectrum resources nowadays available for unlicensed use:

- *Low frequencies*, i.e., in the $2.4-7.125$ GHz range. In addition to the already available 2.4 GHz and 5 GHz bands, a new portion of the radio spectrum has been recently made available. Specifically, in April 2021, the Federal Communication Commission allowed unlicensed devices to exploit frequencies from 5.925 GHz to 7.125 GHz [10]. The European Commission adopted an analogous decision in June 2021 with the release of 500 MHz at similar frequencies ($5.945 - 6.425$ MHz) [11]. These new portions of the spectrum are commonly referred to as the 6 GHz band.

- *High frequencies*, i.e., in the $57.24-70.20$ GHz range. This is the so-called 60 GHz band and is characterized by wider bandwidth availability. However, wireless communication at these

frequencies is more challenging due to the higher propagation loss.

Several IEEE task groups are actively working on the standardization of the Wi-Fi networks of the future [12]. As part of this process, two amendments – namely IEEE 802.11ax and IEEE 802.11ay – were released in May and July 2021 respectively, and another one – the IEEE 802.11be – is in its way for standardization (expected in 2024). The IEEE 802.11ax high-efficiency (HE) standard – Wi-Fi 6 – works in the below 7.125 GHz bands and provides solutions to improve the network efficiency in dense deployments without bringing high improvements to the overall throughput with respect to its predecessor [13]. In IEEE 802.11ax MU-MIMO is enabled also in the uplink (UL) direction and Wi-Fi 6 devices use the orthogonal frequency-division multiple access (OFDMA) transmission scheme with 78.125 MHz sub-channels. While in OFDM all the channels are reserved for one user at a time, the OFDMA scheme allows allocating groups of sub-channels to different users that, in this way, use the channel simultaneously. The throughput improvement aspect is being addressed by the IEEE task group working on the 802.11be extremely high throughput (EHT) release – Wi-Fi 7 – with the promise to reach a data throughput of at least 30 Gb/s using the below 7.125 GHz transmission bands [14], [15]. The utilization of frequencies in the 60 GHz band is instead regulated by the IEEE 802.11ay next generation 60 GHz (NG60) standard – WiGig – that allows achieving data rates of up to 100 Gb/s at a cost of a lower transmission range [16]. Besides the specific standard, a common challenge for the envisioned future high density connectivity scenarios is the fair coexistence between Wi-Fi and other communication technologies operating on the unlicensed 5 GHz, 6 GHz and 60 GHz bands such as the 3GPP NR-U extension for fifth generation (5G) cellular systems [17], [18].

However, the future of Wi-Fi networks is not only concerned with better connectivity solutions. Recently, the scientific community identified a parallel promising research direction, referred to as *Wi-Fi sensing*. The shared objective of the arising works in this area is to turn Wi-Fi devices into environmental sensors opening a plethora of new applications. For example, in a residential scenario, Wi-Fi sensing can enable solutions for elderly care, to promptly detect critical situations or, for assisted living, to support physically impaired people in their daily living activities. The main idea behind Wi-Fi sensing solutions is that the changes in the propagation environment reflect on the frequency response of the Wi-Fi channel. These changes therefore also encode the presence and the movement of a person and Wi-Fi environment sensing is about learning and tracking them. The Wi-Fi CFR is continuously estimated at the receiver device to properly decode the transmitted information and, in turn, can be exploited to learn any changes that occur in the physical environment, mapping them onto actionable insight about the number of people and their activities. This new opportunity attracted the attention of the IEEE working group that, in September 2020, established a task group to standardize the Wi-Fi sensing technology. The new version of the standard – named IEEE 802.11bf – is expected to define modifications to the PHY and MAC layers of the protocol stack to empower Wi-Fi devices with sensing capabilities [19]. The modifications are needed as the two functions have different requirements in terms of physical channel access. Specifically, the communication functionality requires spectrum resources only when transmitting/receiving data, while sensing can be performed accurately only providing the device with regular access to the wireless channel, i.e., by sounding the channel

at regular time intervals. The standardization process is crucial for the widespread implementation of Wi-Fi sensing applications in the variety of devices available in the market. As for now, researchers are constrained to handcraft ad-hoc procedures that allow extracting the CFR, being the source of environmental information. Such procedures require performing some firmware and driver modifications to the Wi-Fi devices thus making difficult their out-of-the-shelf implementation. With the adoption of the new IEEE standard, manufacturers will be required to produce standard-compliant Wi-Fi chips that are ready to support sensing functionalities. This will enable the integration of already assessed sensing algorithms into newer commercial devices, providing joint support for communication and environmental monitoring services. As a side benefit, the standardization process will also provide the researchers with more straightforward methodologies to obtain the CFR from which to develop new sensing applications that, in turn, can be implemented in next-generation COTS devices.

1.2 Sensing-assisted communication for cellular networks

In a mobile network scenario, environmental information (context) refers to the data collected from both the physical scene and the communication network. For instance, context includes terminals' locations, transmission and mobility patterns, network energy consumption and neighboring cells' load, among others. These pieces of information enable the implementation of environment aware management strategies at different functional and control blocks. For example, based on the context, network resources can be *proactively* allocated to the users, thus reducing the service discontinuity risk [20]. In this thesis, we focus on the statistical analysis and exploitation of users' mobility within the network. User's position and mobility patterns can serve for several purposes [21], [22]. In [23], users' locations and content requests are used to manage a network of unmanned aerial vehicles (UAVs), identifying the optimal UAVs locations, the contents placement and the beamforming patterns to provide the users with appropriate quality of service (QoS) and quality of experience (QoE). Moreover, the prediction of future users' positions and network metrics allows implementing proactive management strategies. Location-based prediction of interference sources can be used within channel access protocols, while longer-term mobility predictions can be exploited to re-route processes at the network layer [21]. Another example is provided in [24], where short-term location estimates are utilized to perform beamforming for 5G enabled vehicles and high-speed trains. Mobility-aware caching strategies at the base stations and at the devices are instead proposed in [25], [26]. Finally, some works in the literature consider mobility predictions to manage computing services in MEC enabled networks [27], [28], with the objective of ensuring the continuity of edge computing tasks as the users change their point of attachment to the network. This is very relevant for next-generation cellular networks, where the infrastructure will not only provide communication functionalities, but also the support for intensive computing tasks for moving users. The migration of a computing task from a location to another – to follow the user's movements – is particularly challenging, as it requires to transfer the needed data to the new site. If not properly handled, this may lead to the introduction of additional delays or, even worse, to service interruptions. Hence, algorithms that proactively act

to prevent service discontinuity are gaining momentum, especially within safety-critical applications, such as telemedicine and autonomous driving, where network failures may pose a threat to the users' health condition and safety. Thus, environment aware strategies are expected to be widely adopted in the near future within the process of algorithms' design and implementation.

In Part II of this thesis we consider a vehicular scenario where users are 5G enabled vehicles constantly connected to the cellular network and requiring support to offload computing intensive tasks. In Chapter 4 we devise a learning based framework to predict the vehicles' next point of attachment based on the context information acquired from MIMO transmissions. Our work stems from the intuition that mobility in vehicular networks is a highly correlated process, and such correlation can be captured by advanced neural network designs to anticipate the evolved node base (eNB) to which the user will connect to after the handover from the serving one. The mobility estimates are then exploited in the design of two network computing resource management strategies, presented in Chapter 5 and Chapter 6 respectively. The common objective of the presented algorithms is to minimize the network energy expenditure while ensuring serving continuity and adequate QoS/QoE to the users. The proposed strategies contribute to the advancement of the research in this field, by tackling the problem from a mathematical perspective and providing effective solutions for efficient network resources management.

The performance of the proposed *sensing-assisted communication* algorithms is assessed using emulated vehicles' mobility traces obtained through the simulation of urban mobility (SUMO) package for the city of Cologne [29]. The emulation reflects the habits of city dwellers, encoding typical daily traffic patterns associated with working activities. The cellular network consists of eNBs covering the entire area of interest and deployed ensuring an inter-distance of about 400 meters, in accordance with the envisioned topology for 5G networks.

1.2.1 Preliminaries on cellular networks

The starting point of cellular networks development can be identified with the introduction of the first commercial communication systems in the '80s of the past century. The systems exploited analog technologies to transmit over the *licensed* portions of the radio spectrum, e.g., the 800 MHz and 900 MHz bands. Different independent systems came to the market in different countries, bringing to the formation of a heterogeneous environment. The evolution of digital technologies allowed the definition of a new set of standards, identified as second generation (2G) systems. In addition to the voice transmission, 2G cellular networks provided the users with the short message service (SMS) and the possibility to exchange e-mails. Among the 2G systems, the global system for mobile communications (GSM) one was developed in a spirit of cooperation between the European companies, thanks to the foundation of the European Telecommunications Standards Institute (ETSI). In this way, GSM became an interoperable, widely accepted and robust standard. The successful cooperation strategy continued with the foundation of the Third Generation Partnership Project (3GPP) at the end of 1998. Nowadays, 3GPP unites seven telecommunications standard development organizations – from Europe, America, China, Japan, Korea and India – with the objective to develop interoperable standards. The project allowed offering for the first

time the roaming service, ensuring the communication service support to users that are outside the coverage area of their regional operator. Next, universal mobile telecommunications system (UMTS) was the first standard developed within the 3GPP project and marked a step forward to the development of third generation (3G) systems in the 21^{st} century, providing the users with wireless access to the Internet at a rate of 2 Mb/s. Depending on the country, 3G systems operate in different licensed bands in the $800 - 2,100$ MHz range. The fourth generation of cellular networks came with the 3GPP Releases 8-9 (2008-2010) that standardized the long term evolution of UMTS (LTE) systems. LTE networks provides theoretical data rates of 75 Mb/s in the uplink and 300 Mb/s in the downlink by respectively using OFDM and single carrier - frequency division multiple access (SC-FDMA) modulations, and leveraging the MIMO technology. Next, in 2013, the 3GPP Release 10 made the LTE radio-access technology fully compliant with the International Mobile Telecommunications (IMT)-Advanced global standard initiative promoted by the International Telecommunications Union (ITU) [30], [31]. LTE-Advanced allows achieving theoretical peak data rates of 1.5 Gb/s and 3 Gb/s in the uplink and downlink directions respectively, enabling a variety of new applications with high QoS and QoE requirements, like video streaming and online video conferencing.

LTE-Advanced is nowadays the most adopted standard for cellular networks. However, as for the Wi-Fi technology, the rapid digitalization of the society comes together with an increasing demand for faster and more reliable access to the Internet and cellular connectivity in general. The specific requirements for the fifth generation of cellular networks have been detailed by ITU in [32]. Specifically, ITU identified three main connectivity scenarios that are expected to be supported by 5G networks. At first, 5G systems will address the increasing request of multimedia contents and data through enhanced mobile broadband (eMBB) applications. This service requires both considerable network capacity to serve users in high density areas, and mobility support for users' moving at up to 500 km/s. Ultra-reliable and low latency communications (URLLC) is the second aspect that 5G networks need to consider, providing support for, e.g., remote medical surgery, self-driving cars and industry 4.0 applications. Finally, the rapid increase of IoT solutions requires 5G networks to enable massive machine type communications (mMTC), offering connectivity to a large number of devices transmitting small data packets. Overall, 5G networks are expected to reach peak data rates of 20 Gb/s and latencies as small as 1 ms. These guidelines lead to the development of a new radio access technology named 3GPP new radio (NR), introduced in 2019 within the 3GPP Release 15 [33] and approved by ITU in 2021 [34]. 3GPP NR introduces several innovations such as the exploitation of new portions of the radio spectrum in the $30 - 300$ GHz frequency range – the so-called millimeter wave (mmWave) band – and the integration of the massive MU-MIMO technology through the deployment of antenna arrays with several transmitter/receiver elements (more than 100). Following the standardization of Release 16 in 2021 [35], 3GPP is now working on further enhancements through Release 17 – with expected standardization in 2022 – and Release 18 that will introduce 5G-Advanced [36], [37]. In parallel, the research community started envisioning the sixth generation (6G) of cellular networks that is expected to enable widespread availability of augmented, mixed and virtual reality applications, e-health and industry 4.0 services, with support for high mobility and density [38]. The vision

for next-generation cellular networks is to move from inflexible architectures constrained on the hardware to softwarized and virtualized architectures with the possibility to reprogram the different hardware components. The virtualization of the network components is key toward sharing the same physical infrastructure among different network operators thus optimizing the exploitation of the resources based on the real-time traffic requirements. This will enable the realization of autonomous and self-optimizing networks able to serve devices with different requirements, starting from laptops up to power-constrained IoT terminals [39]. In this context, the O-RAN Alliance – established at the end of 2018 – is working toward the development of intelligent, open, virtualized and fully interoperable mobile networks [40]. The O-RAN Alliance effort is supported by more than 160 companies around the globe, including 24 network operators, and represents a prominent collaborative initiative for the realization of the technical standards regulating future wireless networks.

The future of cellular networks is not only identified with faster and more reliable wireless connections. The present architectures fail to guarantee adequate QoE to the final user when considering the above mentioned applications that are computing intensive and require support for real-time functionalities. Such applications cannot be executed on-board on the end-devices due to memory and energy limitations, neither on the network cloud facilities as the delay constraint would be missed. For this reason, 5G and beyond networks are designed to bring the computing support near to the user – i.e., at the edges of the cellular network – by empowering the eNB sites with computing facilities accessible by the users. The new paradigm is referred to as MEC and is expected to provide the needed support for applications requiring fast and reliable data processing. This computing de-location approach allows a user to offload intensive computing jobs to the closer MEC host (MEH), thus considerably reducing the communication delays compared to the usage of cloud services. The computation service is delivered through the instantiation of a virtual machine (VM) or a containers on the MEH, empowered with the sufficient memory and computing resources to handle the user's request. This poses several challenges associated with the mobility of the users within the network that should be addressed with appropriate handover procedures for the computation service. Enforced by the high potential of the innovation, ETSI is working on the standardization of interoperable MEC architectures [41], together with their integration with 5G – and beyond – mobile networks [42].

1.3 Thesis organization

In the next chapters, we will deepen our discussion on environment awareness in wireless networks by detailing the design and implementation of novel algorithms for communication-assisted sensing and sensing-assisted communication.

We address the first aspect in Part I, focusing on indoor Wi-Fi networks. Specifically, in Chapter 2 we present the work in [163] about human activity recognition through IEEE 802.11ac devices. Chapter 3 refers to [168] where we detail the functioning of a Wi-Fi physical layer fingerprinting algorithm that runs on off the shelf devices.

The other way around is tacked in Part II targeting cellular networks that serve 5G-enabled vehicles. Chapter 4 is about the design of a learning-based mobility prediction algorithm, referring to [169]. In Chapter 5 we describe a mobility-aware computing services migration algorithm as presented in [164]. A scheduler for optimal computing tasks allocation on the network facilities is presented in Chapter 6 considering the work in [162].

Chapter 7 concludes the thesis with final remarks and suggestions for future researches.

Part I

Communication-assisted sensing in Wi-Fi networks

2

Environment- and person- independent human activity recognition through Wi-Fi channel state information

2.1 Introduction

With this chapter we start our journey into *communication-assisted sensing* algorithms by presenting and validating SHARP (*sensing human activities through Wi-Fi radio propagation*). SHARP is a device-free human activity recognition (HAR) system for the automatic detection and classification of human activities in indoor spaces which relies upon the analysis of scattered Wi-Fi signals. SHARP achieves unobtrusive human sensing by leveraging the channel frequency response (CFR) gathered by a commercial-off-the-shelf (COTS) IEEE 802.11ac router. The router is used as a sensor to obtain information on moving targets. The Doppler information at the receiving end of the Wi-Fi link is extracted and utilized to concoct an environment- and person-independent learning based algorithm that reliably detects human activities. As stated in Section 1.1, the main intuition underpinning the proposed algorithm is that the presence of (moving) obstacles in the propagation environment reflects on the shape of the Wi-Fi CFR by introducing characteristic variations that can be captured by advanced machine learning (ML) strategies.

HAR systems are key elements of emerging applications for smart buildings. Among others, they can be profitably used to optimize the buildings' energy consumption, implement alert systems, and provide platforms for smart entertainment, etc. Also, they can be utilized to complement IT solutions for elderly care, to promptly detect critical situations or, for assisted living, to support physically impaired people in their daily activities [43].

We stress that, although similar sensing technology has been proposed as part of prior work,

e.g., [3], [4], [44]–[46], existing algorithms do not generalize across unseen environments and people, for which they were not trained. The proposed system sharply departs from these previous solutions, and improves upon them, in that it reliably works across different indoor environments and people, without having to be re-trained each time the physical wireless propagation setup, or the monitored person, change. To make this possible, SHARP features *an original phase cleaning method*, which permits the extraction of *micro-Doppler* traces from the CFR data. Without this procedure, the phase offsets that affect any Wi-Fi transmission would be so strong to hide finer phase variations and, in turn, would prevent the extraction of the micro-Doppler information. From the micro-Doppler traces is is then possible to remove the components that are due to static objects and only retain the relevant information that is connected with the human movement of interest, leading to the desired generalization capability.

In the last decade, since the pioneering work of Adib and Katabi [3], wireless signals of opportunity have been extensively researched as a means to perform device-free localization and activity recognition tasks, including pattern and gesture recognition [44], and even estimating biological signals at distance, such as the respiration rate or the heartbeat [45]. In this respect, Wi-Fi signals are particularly appealing, due to the high availability of Wi-Fi enabled devices in most residential and working spaces [46]. While active sensing techniques allow localizing and following a user as she/he carries along a Wi-Fi-connected device, passive sensing approaches obtain such information by monitoring the changes in the Wi-Fi channel, where users act as intermediate scatterers that modify the channel frequency response (CFR) [4].

Here, we are concerned with the recognition of dynamic and coarse-grained activities performed by humans in indoor spaces, such as jumping, running, walking, etc. using COTS IEEE 802.11ac routers. Human movement data is extracted from the channel state information (CSI), a frequency estimate of the radio channel describing how the signal changes as it propagates from the transmitter to the receiver. Note that channel estimation is continuously performed by Wi-Fi routers for communication purposes, and specific tools allow gathering such information from COTS devices, exploiting them for environmental sensing. So far, many approaches have been presented, e.g., [4], [46]. However, they fail to reliably recognize human activities across different scenarios, days or with different persons, without being re-trained with the new data. That is, they *fall short in generalizing across persons and environments*.

To improve upon prior work, and offer a usable system across different setups, we design SHARP, an original framework to extract micro-Doppler traces of human motion activity from IEEE 802.11 routers, combining them with a learning-based algorithm that acts as a feature extractor and activity classifier. The extraction of the micro-Doppler information is enabled by a novel *phase sanitization technique*, that effectively copes with the phase offsets affecting the channel CSI estimates. The proposed phase sanitization approach is a major step forward in the HAR via Wi-Fi channels: as we show in Section 2.7.2, prior sanitization algorithms fail to provide a sufficiently accurate phase signal and are thus ineffective to our purpose. Once available, the Doppler shift reveals the velocities of the scattering points during the transmission events and is not affected by static objects (furniture, walls, etc.), allowing one to gauge dynamic HAR features. As quantified in Section 2.7, we found it to be a robust representation for environment- and

person-independent HAR.

The present work sharply departs from the literature on HAR with Wi-Fi devices, as follows.

- We propose an original method for CSI phase correction (also referred to as *sanitization*). The algorithm is independently used at each receiving antenna without the need for a reference stream. Hence, the space diversity provided by the receiving antennas can be fully exploited for sensing purposes.

- We present and validate `SHARP`, an environment- and person-independent learning based framework for HAR, which leverages the Doppler effect caused by human motion. The information collected at the four monitoring antennas is combined to identify the activity, regardless of the user's position in the monitored area.

- We assess `SHARP` robustness to changing environmental conditions through CSI data collected with commercial off-the-shelf IEEE 802.11ac devices operating on an 80 MHz frequency band. It reaches an average accuracy higher than 95% in the identification of four activities – namely walking, running, jumping and sitting – in the most challenging scenario, i.e., when the person, the day and the indoor environment change with respect to those in the training set. `SHARP` is validated against prior work – `DeepSense` [47], `EI` [48] and `MatNet-eCSI` [49] – achieving superior generalization capabilities across subjects and environments.

The rest of the chapter is organized as follows. The related work is reviewed in Section 2.2. In Section 2.4 and Section 2.5, we respectively detail the processing on the CSI data and the learning architecture for HAR, which represent the `SHARP` building blocks. The experimental setup is presented in Section 2.6, while the performance of the proposed approach is evaluated in Section 2.7. Conclusions are drawn in Section 2.8.

2.2 Related work

2.2.1 CSI based human activity recognition

HAR through CSI data from COTS Wi-Fi devices was first studied in, e.g., E-eyes [50], CARM [51]. More recently, several articles showed the effectiveness of machine learning techniques in building algorithms that distinguish human activities based on CSI features [47], [52]–[54]. However, these works do not focus on the robustness to environmental changes and on the generalization capability to previously unseen environments and subjects, which are key enablers for the successful development of Wi-Fi-based sensing systems [4]. Only a few works try to address these weaknesses. In [48], the authors use a neural network based approach to extract environment-independent features from the CSI amplitude to recognize human movements. The performance of their algorithm is promising, but remains below 80% in the best scenario. In [55], transfer learning is shown to be effective to adapt the Wi-Fi-based HAR algorithm to different persons

and days for the same environment. The algorithm presented in [56] leverages generative adversarial networks to generalize on new persons, while in [49] the matching network one-shot learning approach [57] is proposed to bridge the gap between previously seen environments and new ones. A recent work [58] addresses the problem of location and subject independent HAR through a learning architecture consisting of three deep neural networks. The algorithm is trained on the CSI amplitude collected by monitor routers placed in different positions inside a room, and is tested in the same room by changing the location of a single router. The approach presents a significant performance degradation when evaluated on other datasets (going from 99% to 80% of accuracy).

To the best of our knowledge, no work in the literature proposes a system that generalizes well on unseen environments, days and subjects without any re-training step and using COTS devices. In the present work, we propose an effective solution to this problem, and we consider as benchmarks for comparison, three recent approaches from the literature, i.e., `DeepSense` [47], `EI` [48] and `MatNet-eCSI` [49].

2.2.2 Exploitation of the CFR phase

Some works, e.g, [59]–[61], consider a reference antenna and use the phase differences at the remaining ones to perform sensing. By construction, such phase difference is not affected by systematic offsets (same value across the antennas), but it is unable to cope with offsets that are antenna-specific. Exploiting a similar idea, in [62], the authors correct the rotation errors using a reference signal, obtained by connecting with a cable one of the available monitoring antennas with the transmitter. However, the need for a reference reduces the spatial diversity that can be exploited for sensing purposes at the monitor station. The possibility to remove the unwanted phase offsets without exploiting any reference is less addressed in the literature. In [63], the authors propose to mitigate the errors by considering the average signal over a number of time instants, while in [64] a two-step approach is presented to estimate and remove some of the antenna-independent offsets. A different strategy is adopted in [49], where the authors estimate the systematic phase shift at each CFR acquisition. With their approach they remove the static component, that is used as a reference for the algorithm, and retain the dynamic one, using it as the input for the subsequent HAR framework.

In this work, we correct the phase offsets by devising an optimization approach that finds the optimal CFR parameters from the raw channel data. For each CFR sample, we extract the contributions of the different radio paths and exploit the component related to the strongest path as a reference to correct the signal phase. This allows the removal of the non-systematic offsets that affect the signal. As discussed above, this would be infeasible if considering as a reference a signal extracted form a different physical measurement (either in the time or in the spatial dimension).

2.2.3 Doppler based applications

A few works in the literature exploit the Doppler shift computed from Wi-Fi CSI data. In **Qian2017_2**, [65], [66], the Doppler information is used to track the movement of a user inside an environ-

ment. Two different monitoring devices, placed in strategic positions, are used to obtain effective Doppler shift estimates. In [67] the authors leverage the Doppler effect for breathing detection and people counting. The sensing system consists of one Wi-Fi access point and two universal software radio peripherals (USRPs) monitoring the target channel and a reference one that needs to remain stable. A sensing system consisting of a single monitoring station is presented in [68], to recognize three different bodyweight exercises. In [68], the transmitter and the monitor stations must be placed in specific positions with respect to the user, so that the activity is performed perpendicularly to the radio link. Moreover, the detection algorithm is based on metrics that are manually extracted from the raw CSI data and the Doppler trace. This makes the proposed approach highly activity-dependent, as the features are specifically designed to separate the three considered physical exercises. HAR through Doppler information has been recently addressed in [69]. The latter algorithm uses phase sanitization based on a reference antenna, followed by a support vector machine (SVM) classifier. This strategy allows obtaining a good recognition accuracy (96%) on test data collected in the same environment where the algorithm is trained, proving the effectiveness of the Doppler information for HAR purposes. Also, the system performance is improved considering additional data from a wearable inertial measurement unit (IMU), increasing the accuracy to nearly 100%. However, no performance assessment is carried out for scenarios that are not seen at training time. SHARP outperforms [69] by exploiting an advanced phase sanitization approach and a more powerful learning architecture. As a result, it reaches accuracies close to 100% by only exploiting CSI data (no IMU). Moreover, SHARP generalizes to unseen environments, by correctly recognizing the activity in at least 95.99% of the cases.

Our intuition, behind the design of SHARP, is that the Doppler shift naturally lends itself to the separation/removal of the radio reflections coming from static objects or structures (e.g., room walls) from those generated by moving targets (e.g., humans). If available, it can be used as a reference signal domain to perform environment independent recognition tasks, assuming that the environment is mostly static. Our work descends from this line of reasoning and our novelty resides in the fact that we were, for the first time, able to extract the Doppler signal from 802.11ac COTS devices. After that, such Doppler signal must be used in combination with dedicated processing and learning architectures, which are the second main contribution of our present work. In the remainder, the extraction of the Doppler is presented in detail, along with its use with inception based neural networks.

2.3 Wi-Fi OFDM transmission model

Next, we detail the main blocks of a Wi-Fi transmission chain, deriving the expression for the frequency response of the Wi-Fi channel that is the core of Part I of the thesis.

2.3.1 Transmitted signal model

Wi-Fi systems adopt orthogonal frequency-division multiplexing (OFDM), by transmitting the user information over K partially overlapping and orthogonal sub-channels, with K even. Data

over each sub-channel is modulated with a conventional digital modulation scheme (such as QPSK, 16QAM). The input bits are grouped and mapped onto source data symbols, which are complex numbers representing the modulation constellation points. Such constellation points are further grouped into K elements each, i.e., the OFDM symbols. OFDM symbols are then fed to an inverse fast Fourier transform (IFFT) block that transforms the data prior to transmitting it over the wireless channel, in parallel, over K sub-channels with carriers spaced apart by $\Delta f = 1/T$ Hz (T is the OFDM symbol time).

The duration of an OFDM symbol is $\bar{T} = T + T_{\text{CP}}$, where T_{CP} is the duration of the cyclic prefix, added to mitigate inter-symbol interference. Specifically, for IEEE 802.11ac the transmission bandwidth is 80 MHz, the samples are clocked out at 80 Msps, the number of sub-channels is $K = 256$, $T = 1/\Delta f = 3.2$ μs (i.e., $\Delta f = 312.5$ kHz), $T_{\text{CP}} = 0.8$ μs, and, in turn, $\bar{T} = 4$ μs.

Let $\mathbf{a}_m = [a_{m,-K/2}, \ldots, a_{m,K/2-1}]$ be m-th OFDM symbol, where $a_{m,k}$ is the k-th OFDM sample. After digital to analog conversion, the baseband OFDM signal for the m-th symbol is

$$x_m(t) = \sum_{k=-K/2}^{K/2-1} a_{m,k} e^{j2\pi kt/T}, \tag{2.1}$$

where $k \in \{-K/2, \ldots, K/2-1\}$ is the sub-channel index. Considering M subsequent blocks, the baseband signal is

$$x(t) = \sum_{m=0}^{M-1} x_m(t)\xi(t - m\bar{T}), \tag{2.2}$$

with

$$\xi(t) = \begin{cases} 1 \text{ if } t \in [-T_{\text{CP}} - TK/2, TK/2] \\ 0 \text{ otherwise} \end{cases}, \tag{2.3}$$

and the signal transmitted over the Wi-Fi channel is obtained by upconverting $x(t)$ to the carrier frequency f_c,

$$s_{\text{tx}}(t) = e^{j2\pi f_c t} x(t). \tag{2.4}$$

2.3.2 Received signal model

At each receiver antenna, P signal copies are collected, due to the scatterers that the signal $s_{\text{tx}}(t)$ encounters (multi-path propagation). Each path p is characterized by an attenuation $A_p(t)$ and a delay $\tau_p(t)$. Neglecting the additive white Gaussian noise, the received signal $s_{\text{rx}}(t)$ is written as

$$\begin{aligned} s_{\text{rx}}(t) &= \sum_{p=0}^{P-1} A_p(t) s_{\text{tx}}(t - \tau_p(t)) \\ &= e^{j2\pi f_c t} \sum_{p=0}^{P-1} A_p(t) e^{-j2\pi f_c \tau_p(t)} x(t - \tau_p(t)), \end{aligned} \tag{2.5}$$

and its baseband representation $y(t)$ is expressed as,

$$y(t) = s_{\text{rx}}(t)e^{-j2\pi f_c t}. \tag{2.6}$$

A rectangular window $[m\bar{T}, m\bar{T}+T]$ is used at the receiver to collect and decode the information carried by an OFDM symbol at a time. Without loss of generality, we assume $m = 0$ and hence we omit such index from the following equations. The transmitted symbol a_k is recovered by computing the Fourier transform of the signal in the received window:

$$\begin{aligned}
\hat{a}_k &= \int_{\bar{T}}^{\bar{T}+T} y(t)e^{-j2\pi kt/T}dt \\
&= \sum_{p=0}^{P-1} A_p e^{-j2\pi f_c \tau_p} \sum_{b=-K/2}^{K/2} a_b e^{-j2\pi b\tau_p/T} \int_{\bar{T}}^{\bar{T}+T} e^{j2\pi(b-k)t/T}dt \\
&= a_k T \sum_{p=0}^{P-1} A_p e^{-j2\pi(f_c+k/T)\tau_p},
\end{aligned} \tag{2.7}$$

where we consider A_p and τ_p constant over an integration interval T. The sum in the last line of Eq. (2.7) corresponds to the frequency response of the Wi-Fi channel,

$$H_k = \sum_{p=0}^{P-1} A_p e^{-j2\pi(f_c+k/T)\tau_p}, \tag{2.8}$$

that is estimated based on the known preamble symbols. In Eq. (2.7), we consider that the path attenuation and delay remain constant over each window, i.e., $A_p(t) = A_p$ and $\tau_p(t) = \tau_p$. Also, exchanging the order of integration and summation is legitimate as we deal with finite quantities, and we used $\int_{\bar{T}}^{\bar{T}+T} e^{j2\pi(b-k)t/T}dt = 0$ if $k \neq b$.

2.3.3 Phase offsets in the Wi-Fi channel estimates

Hardware artifacts make the CFR gathered from Wi-Fi devices slightly deviate from the model in Eq. (2.8). These artifacts introduce offsets (rotation errors) in the phase information, among which the most significant are [62], [70]:

- *carrier frequency offset (CFO)*, due to the difference between the carrier frequency of the transmitted signal and the one measured at the receiver. The CFO is only partially compensated for at the receiver [64].

- *sampling frequency offset (SFO)*, due to the imperfect synchronization of the clocks between transmitter and receiver.

- *packet detection delay (PDD)*, due to the time required to recover the transmitted modulated symbols from the received signal [64].

- *phase-locked loop offset (PPO)*, due to the phase-locked loop (PLL), the entity responsible for randomly generating the initial phase at the transmitter.
- *phase ambiguity (PA)*, due to the phase difference (multiples of π) between the antennas, that in static conditions should remain constant.

Considering these contributions, the complete expression for the phase of the p-th path in the received signal is

$$\bar{\theta}_{p,k} = -2\pi(f_c + k/T)\tau_p + \theta_{\text{CFO}} - 2\pi k(\tau_{\text{SFO}} + \tau_{\text{PDD}})/T + \theta_{\text{PPO}} + \theta_{\text{PA}}. \tag{2.9}$$

Note that, while the CFO, SFO and PDD contributions take the same value across different antennas, the initial PLL phase (PPO) and PA are antenna specific [71], [72].

2.4 CSI data processing

At the OFDM receiver, the channel parameters (amplitudes and phases) are continuously estimated for all the sub-channels. This information is computed for each received packet based on known preamble symbols, and is collected by the CSI, a large (environment-dependent) complex matrix describing the CFR for each sub-channel along every receiving antenna. Specifically, for each pair of transmit and receive antennas, the CFR is a set of complex numbers $A_k e^{j\theta_k}$ specifying the attenuation A_k and the phase shift θ_k for each sub-channel $k \in \{-K/2, \ldots, K/2-1\}$. Considering Eq. (2.8), the CFR estimated on packet n is written as

$$\begin{aligned} H_k(n) &= A_k(n)e^{j\theta_k(n)} \\ &= \sum_{p=0}^{P-1} A_p(n)e^{-j2\pi(f_c+k/T)\tau_p(n)}. \end{aligned} \tag{2.10}$$

In our evaluation, the CFR matrix is estimated by a monitor Wi-Fi device running the Nexmon CSI tool [73], as detailed in Section 2.6. We remark that the collected CFR slightly deviates from the theoretical model in Eq. (2.10) due to hardware artifacts (see Section 2.3.3), which introduce an undesired phase offset $\theta_{\text{offs},k}$, i.e.,

$$\bar{H}_k(n) = H_k(n)e^{\theta_{\text{offs},k}}. \tag{2.11}$$

Next, we present the steps that we implemented to clean the CFR matrix that is extracted by the Nexmon CSI tool. Note that a new CFR matrix is retrieved for each received packet. Thus, the interval between subsequent acquisitions of the CFR is variable but, for the sake of exposition, in the following analysis we assume that a new sample is available every T_c seconds, with T_c fixed. Note that, in a real Wi-Fi network, the traffic exchanged between already deployed access points and user devices (e.g., streaming services), can be captured and exploited for HAR recognition purposes. When this is not possible, two Wi-Fi access points are required to recreate the setup,

using a simple network controller to transmit packets at regular intervals and a second monitor device to infer the CFR from these.

2.4.1 Phase sanitization

As detailed in Section 2.3.3, the undesired phase offset $\theta_{\text{offs},k}$ in Eq. (2.11) contains different contributions. Some of them, namely, the CFO, PPO and the PA, although changing in time, have the same value across the sub-channels in each receiving antenna. The SFO and the PDD are instead sub-channel dependent. As for Eq. (2.9), the offset $\theta_{\text{offs},k}$ experienced at one receiving antenna in sub-channel k can be expressed as

$$\theta_{\text{offs},k} = -2\pi k(\tau_{\text{SFO}} + \tau_{\text{PDD}})/T + \theta_{\text{CFO}} + \theta_{\text{PPO}} + \theta_{\text{PA}}. \tag{2.12}$$

The main idea of our approach for phase sanitization is that the contribution of each path in Eq. (2.11) is affected by the same phase shift $\theta_{\text{offs},k}$. Hence, if we were able to separate the different paths, we could use (any) one of them as a reference to remove the phase offset from the CFR. In our method, the strongest path will be used to this end, as this is the path whose parameters are more reliably estimated at the receiver. The details are given shortly below. (Note that in the following analysis the time index n is omitted in the interest of readability.)

Let $\mathbf{h}$ be the K-dimensional vector collecting the CFR information for the K sub-channels,

$$\mathbf{h} = [\bar{H}_{-K/2}, \ldots, \bar{H}_0, \ldots \bar{H}_{K/2-1}]^T. \tag{2.13}$$

To separate the P multi-path contributions, we define a grid of P' possible paths with $P' > P$ and solve a minimization problem to select the P components out of the P' ones that contribute to the CFR. We consider the following decomposition

$$\mathbf{h} = \mathbf{T}\mathbf{r}, \tag{2.14}$$

where $\mathbf{T} = [\mathbf{T}_{-K/2} \ldots \mathbf{T}_{K/2-1}]^T$ is an $(K \times P')$-dimensional matrix collecting the contributions to the CFR that depend on the sub-channel index k, while $\mathbf{r}$ is a P'-dimensional column vector representing the sub-channel independent terms. The P'-dimensional row vectors $\mathbf{T}_k$ are defined though a dictionary of candidate total delays $\tau_{p,\text{tot}} = \tau_p + \tau_{\text{SFO}} + \tau_{\text{PDD}}$, with $p \in \{0, \ldots, P'-1\}$ as

$$\mathbf{T}_k = \begin{bmatrix} e^{-j2\pi k\tau_{0,\text{tot}}/T} & \ldots & e^{-j2\pi k\tau_{P'-1,\text{tot}}/T} \end{bmatrix}, \tag{2.15}$$

The column vector $\mathbf{r}$ can instead be modeled as

$$\mathbf{r} = e^{j(\theta_{\text{CFO}}+\theta_{\text{PPO}}+\theta_{\text{PA}})} \begin{bmatrix} A_0 e^{-j2\pi f_c \tau_0} \\ \vdots \\ A_{P'-1} e^{-j2\pi f_c \tau_{P'-1}} \end{bmatrix}, \tag{2.16}$$

and is obtained by solving the following minimization problem

$$\mathbf{P1}: \quad \mathbf{r} = \underset{\tilde{\mathbf{r}}}{\operatorname{argmin}} \ ||\mathbf{h} - \mathbf{T}\tilde{\mathbf{r}}||_2^2 + \lambda \, ||\tilde{\mathbf{r}}||_1 \,, \tag{2.17}$$

where $\lambda > 0$ is the weighing parameter for the ℓ_1 regularization term in the Lasso regression. As $\mathbf{r}$ reflects the sparse channel impulse response, **P1** is a compressive sensing reconstruction problem and can be solved through quadratic optimization using, e.g., [74]. The non-zero entries in the solution $\mathbf{r}$ reveal the presence of a path p with corresponding total delay $\tau_{p,\text{tot}}$.

Next, using the decomposition in Eq. (2.14), vector $\mathbf{r}$, that we found solving problem **P1**, and Eq. (2.15), we are able to separate the contributions of the different paths in the CFR for each sub-channel k by applying the following Hadamard product

$$\mathbf{X}_k = \mathbf{T}_k^T \circ \mathbf{r}. \tag{2.18}$$

$\mathbf{X}_k$ can be rewritten by replacing the terms in Eq. (2.18), as

$$\mathbf{X}_k = e^{-j2\pi k(\tau_{\text{SFO}}+\tau_{\text{PDD}})/T} e^{j(\theta_{\text{CFO}}+\theta_{\text{PPO}}+\theta_{\text{PA}})} \begin{bmatrix} A_0 e^{-j2\pi(f_c+k/T)\tau_0} \\ \vdots \\ A_{P'-1} e^{-j2\pi(f_c+k/T)\tau_{P'-1}} \end{bmatrix}. \tag{2.19}$$

At this point, we define $p^* \in \{0, \ldots, P'-1\}$ as the position where $\mathbf{r}$ has the highest amplitude, which, in turn, is associated with the strongest path. Let X_{k,p^*} be the p^*-th entry of $\mathbf{X}_k$. By multiplying $\mathbf{X}_k$ by the complex conjugate of X_{k,p^*}, we remove the phase components that are constant across all the paths, including the offset, obtaining

$$\bar{\mathbf{X}}_k = A_{p^*} e^{j2\pi(f_c+k/T)\tau_{p^*}} \begin{bmatrix} A_0 e^{-j2\pi(f_c+k/T)\tau_0} \\ \vdots \\ A_{P'-1} e^{-j2\pi(f_c+k/T)\tau_{P'-1}} \end{bmatrix}. \tag{2.20}$$

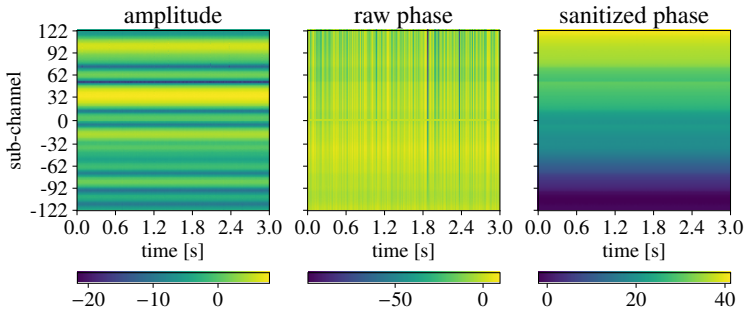
Hence, summing up the elements of $\bar{\mathbf{X}}_k$ (associated with the paths from $p = 0$ to $p = P'-1$), we attain a CFR estimate for each of the sub-channels, where the phase offset is mitigated, as follows,

$$\begin{aligned} \hat{H}_k &= \hat{A}_k(n) e^{j\hat{\theta}_k(n)} \\ &= A_{p^*} \sum_{p=0}^{P'-1} A_p e^{-j2\pi(f_c+k/T)\hat{\tau}_p} \\ &\simeq A_{p^*} e^{j2\pi(f_c+k/T)\tau_{p^*}} H_k, \end{aligned} \tag{2.21}$$

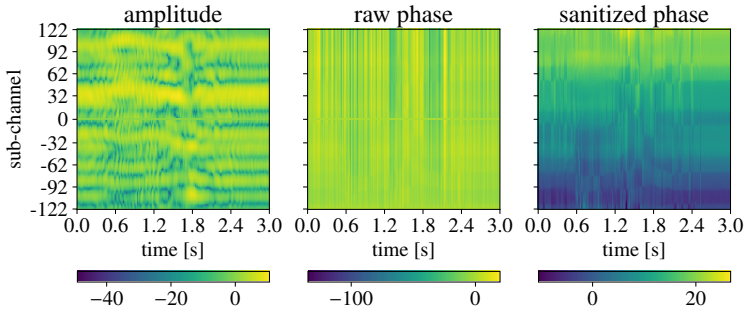
where $\hat{\tau}_p = \tau_p - \tau_{p^*}$. Note that $\hat{H}_k$ of Eq. (2.21) represents the CFR estimate for sub-channel k, where the phase offset has been removed and the contribution of each path is modulated according to the amplitude and phase of path p^*.

2.4.2 Doppler trace computation

Figure 2.1 shows how the amplitude and the phase change during a time interval of 3 seconds in two different scenarios, i.e., an empty room (on the left) and a room with a moving person (right plots). The presence of a human induces changes in the extracted channel parameters, which can be exploited by HAR algorithms. However, the CFR is affected by the whole indoor multi-path propagation environment and hence also accounts for the reflections from static objects. The reflected signals combine differently at each sub-channel, and the delays induce sub-channel specific phase shifts (see Eq. (2.10)). Such a behavior is *environment-specific* and is clearly identifiable in the amplitude plots of Figure2.1 (see the horizontal patterns in the figure). This fact does not allow developing robust algorithms for HAR that generalize across different environments, as the CFR is strongly affected by the room configuration itself (static objects, including walls, and the room shape). Also, even considering the same indoor space, slight changes in the position of the objects therein have a non-negligible effect on the measured CFR, thus making the HAR task more challenging. As an example, Figure 2.2 shows the CFR amplitude collected in two different days within the same empty room.



(a) Amplitude (in dB scale), raw and sanitized phases (unwrapped) of CSI data for an empty room.



(b) Amplitude (in dB scale), raw and sanitized phases (unwrapped) of CSI data with a person running.

Figure 2.1: Amplitude (in dB scale), raw and sanitized phases (unwrapped) of CSI data for an empty room and with a person running. Each trace is three seconds long and shows the behavior on each of the monitored sub-channels (y-axis). Note that CSI is not available on the three central sub-channels, see Section 2.6 for details.

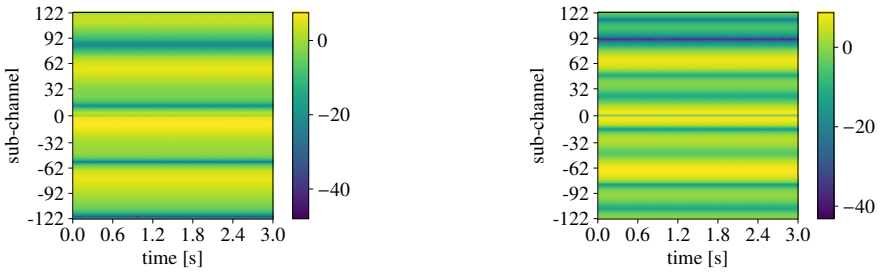


Figure 2.2: CSI amplitude (dB scale) collected in the same empty room but in two different days. Environmental variations from one day to another change the multi-path channel response.

Due to these facts, in our framework we exploit the Doppler effect to obtain effective features for environment-independent HAR. The Doppler effect corresponds to a shift in the signal phase measured at the receiver as the geometry of the multi-path propagation changes during a transmission event. The movements of the scatterers cause variations in the time taken for the signal to reach the receiver through each of the propagation paths. This reflects in a phase shift in the received OFDM signal and, in turn, in the CFR samples defined in Eq. (2.10). Specifically, considering path p in $H_k(n)$, its associated delay, $\tau_p(n)$, can be expressed as the sum of two contributions. Let ℓ_p be the path length related to the initial position of the scattering point and $\Delta_p(n)$ be the delta caused by the movement of the point during the transmission period nT_c. We have

$$\tau_p(n) = \frac{\ell_p + \Delta_p(n)}{c}, \tag{2.22}$$

with

$$\Delta_p(n) = -\int_0^{nT_c} v_p(x) \cos \alpha_p(x) dx, \tag{2.23}$$

where v_p indicates the speed of scatterer p, while $\cos \alpha_p$ results from the combination of sinusoidal functions related to the angles of motion of the scatterer, and the angles of arrival and departure of the signal.

The activity-related movements of a human cause complex variations in the phase, as each body part acts as a scatterer moving at a specific velocity $v_p(x)$. This is revealed by the Doppler vector computed from the CFR samples through a short-time Fourier transform over N subsequent channel estimates, i.e., during a *channel observation window*. These N estimates are respectively acquired by extracting the CFR samples at the Wi-Fi monitor for N subsequent packets, collected with sampling period T_c. The value of N is selected so that the attenuation, the velocities, and the angles can be considered constant during the i-th observation window $[iNT_c, (i+1)NT_c]$, with $i \geq 0$. This allows us to rewrite $\Delta_p(n)$ (Eq. (2.23)) as

$$\Delta_p(n) = -v_p \cos \alpha_p n T_c. \tag{2.24}$$

Note that, the sanitized phase expressed by Eq. (2.21) depends on $\hat{\tau}_p = \tau_p - \tau_{p^*}$, where τ_{p^*} is the propagation delay associated with the strongest path. Our objective corresponds to estimating the Doppler shift $v_p \cos \alpha_p$ (see Eq. (2.24)). Fortunately, we can reasonably assume that the strongest path, p^*, refers to a static component, thus $\Delta_{p^*}(n) = 0$ and $\tau_{p^*} = \ell_{p^*}/c$, which is constant. It follows that the term depending on p^* in Eq. (2.21) does not alter the dynamic component of the signal, which is entirely contained in H_k and, in turn, reveals the subject's movements. It descends that we are entitled to use Eq. (2.21) to track the dynamic component in the Doppler domain. We do so by defining the $K \times N$ dimensional CFR matrix $\mathbf{H}_i$ associated with the i-th observation window, with entry (k, n) referring to the CFR estimate from Eq. (2.21) obtained for OFDM sub-channel k and packet n within the current window i, formally,

$$\mathbf{H}_i = \begin{bmatrix} \hat{H}_{-K/2}(iN) & \dots & \hat{H}_{-K/2}((i+1)N-1) \\ \vdots & & \\ \hat{H}_{K/2-1}(iN) & \dots & \hat{H}_{K/2-1}((i+1)N-1) \end{bmatrix}. \tag{2.25}$$

A parallel can be built between matrix Eq. (2.25) and the signal obtained by a frequency-modulated continuous-wave (FMCW) radar, where the *sub-channel* axis (row index) encodes the *fast-time* component, while the *time* axis (column index, i.e., subsequent time estimates) serves as the *slow-time* component [75]. This analogy guides us on the transformation that is to be applied to the matrix to extract the desired Doppler information. Specifically, each element of the N_D-dimensional Doppler vector $\mathbf{D}_i = [d_i(-N_D/2), \dots, d_i(N_D/2-1)]^T$ is obtained as

$$d_i(u) = \sum_{k=-K/2}^{K/2-1} |\mathscr{F}\{\mathbf{H}_i\}(k,u)|^2, \tag{2.26}$$

where $u \in \{-N_D/2, \dots, N_D/2-1\}$ is the Doppler index and $\mathscr{F}\{\cdot\}$ indicates the Fourier transform operation:

$$\begin{aligned} \mathscr{F}\{\mathbf{H}_i\}(k,u) &= \sum_{n=0}^{N_D-1} \hat{H}_k(n) W_k(n) e^{-j2\pi nu/N_D} \\ &\simeq A_{p^*} \sum_{p=0}^{P'-1} A_p e^{-j2\pi(f_c + k/T)\hat{\ell}_p/c} \sum_{n=0}^{N_D-1} W_k(n) e^{j2\pi n(f_c \hat{v}_p \cos\hat{\alpha}_p T_c/c - u/N_D)}. \end{aligned} \tag{2.27}$$

In 2.27, W_k is the Hanning function and the term $e^{j2\pi(k\hat{v}_p \cos\hat{\alpha}_p n T_c/T)/c}$ is negligible and omitted in the final expression. Note also that, to increase the velocity resolution, the signal can be zero-padded out to N_D samples before applying the Fourier transform.

By summing over the sub-channel axis k, in Eq. (2.26), we preclude the possibility of retrieving ℓ_p taking the maximum of the Fourier transform over k. However, by definition, ℓ_p is constant during a transmission slot and only depends on the position of the corresponding scattering point inside the room. Here, we are interested in capturing the path variations ($\Delta_p(n)$) caused by a moving subject, thus the constant value ℓ_p is irrelevant to our HAR task.

The non-zero entries u in the Doppler vector reveal the presence of a scatterer with associated velocity

$$v_p \cos \alpha_p = \frac{uc}{f_c T_c N_D}, \tag{2.28}$$

where c is the speed of light. Note that, we do not need to estimate v_p and $\cos \alpha_p$ separately. In fact, we are interested in finding a representative feature for the activity-related movements, i.e., we aim at revealing the dynamic component $\Delta_p(n)$ (Eq. (2.24)) of the paths in the environment. It follows that the quantity in Eq. (2.28) is a good proxy to such dynamic components and is a good input feature for SHARP. We recall that this feature is obtained considering time windows consisting of N subsequent channel estimates each. In the following, we refer to Doppler trace, or Doppler spectrogram, as the matrix obtained by stacking a number of subsequent Doppler vectors, computed for consecutive observation windows.

As an example, in Figure 2.3 we plot the Doppler spectrograms as a person performs each of the four considered activities inside a room, compared with an empty room case. The spectrograms show the evolution of the quantity in Eq. (2.28) over three-seconds long observation windows and the colormap refers to the normalized amplitudes in dB. As expected, the Doppler trace related to the empty room only presents non-negligible power at the zero-velocity bin, revealing no movement in the environment. Instead, in the presence of a moving person the power is spread across different bins, reflecting the human-related multi-path changes.

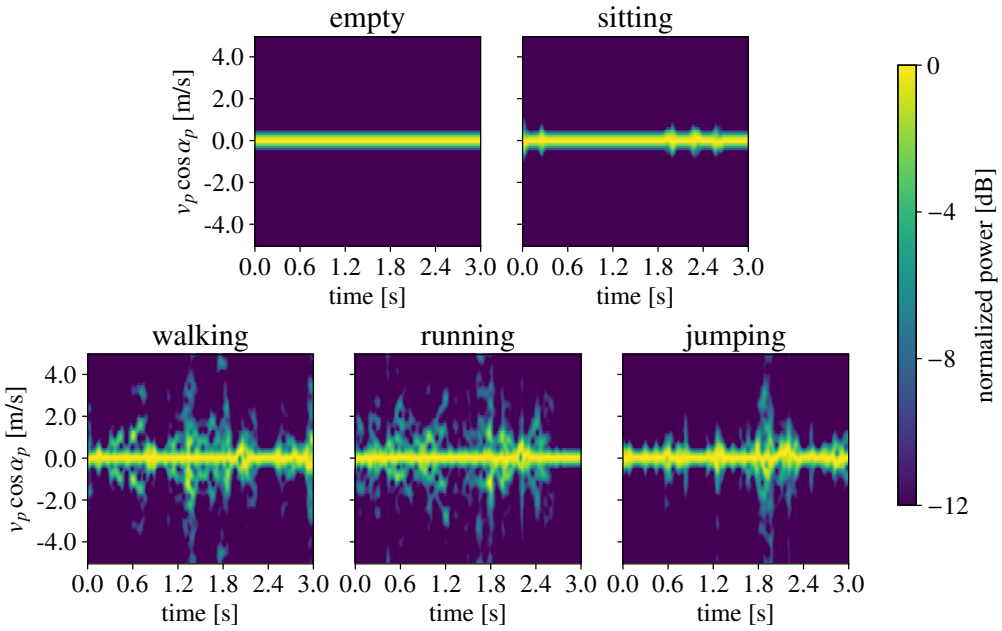


Figure 2.3: Doppler spectrogram expressed as $v_p \cos \alpha_p$ for five different setups (from left to right: empty room, sitting, walking, running and jumping). The quantity can be positive or negative depending on the way the scattering points move, changing the geometry of the multi-path propagation channel.

We stress that the main idea behind SHARP is to perform environment and person independent activity recognition. To this end, we exploit the proprieties of the Doppler spectrogram discussed above combined with the learning architecture detailed in the next section. The framework is trained on a single scenario (i.e., same environment and person) and its robustness is assessed considering different test cases. Note that the framework is used at testing time without performing any retraining, to assess its performance when the day, the person and/or the environment change with respect to those used at training time.

2.5 Learning architecture for HAR

SHARP is conceived to fully exploit the data gathered from a commercial Wi-Fi access point, i.e., using amplitude and phase information from the CFR, and combining this data for the available N_{ant} receiving antennas. Activity recognition is performed using N_w subsequent channel estimates at a time, which amounts to monitoring the channel for $N_w T_c$ seconds. Note that this can be implemented in a sliding window fashion, thus updating the activity label at every new CSI sample.

The HAR algorithm consists of two steps.

1. First, we compute the Doppler traces from the data collected at all the receiving antennas and we use them to obtain activity estimates though a neural-network based algorithm (Section 2.5.1). Specifically, the machine learning based prediction chain that we build is *independently applied* to the data stream coming from each of the available antennas.

2. As a result of the previous step, we obtain N_{ant} independent predictions, one per antenna, which are combined, in a second step, through a decision fusion method that leads to the final activity estimate (Section 2.5.2).

We remark that an alternative approach that combines the data from the antennas (at the input of the above decision chains), would also be possible. Such combined input data (the Doppler traces coming from the antennas) would then be fed to a neural network to classify the activity. We experimentally verified that this is not a viable method to obtain an environment-invariant classifier, as the classification result depends on the antenna ordering and, in turn, on the person's location inside the room.

2.5.1 Activity classifier for the single antenna system

We framed the problem as a multi-class classification task with five classes (four user activities plus empty room), tackling it through a learning-based approach. The designed neural-network classifier takes as input the Doppler trace from a single antenna element, i.e., an $N_w \times N_D$ dimensional matrix obtained by stacking N_w subsequent Doppler vectors, and returns the label of the activity being performed. The classification architecture for the single-antenna system is shown in Figure 2.4.

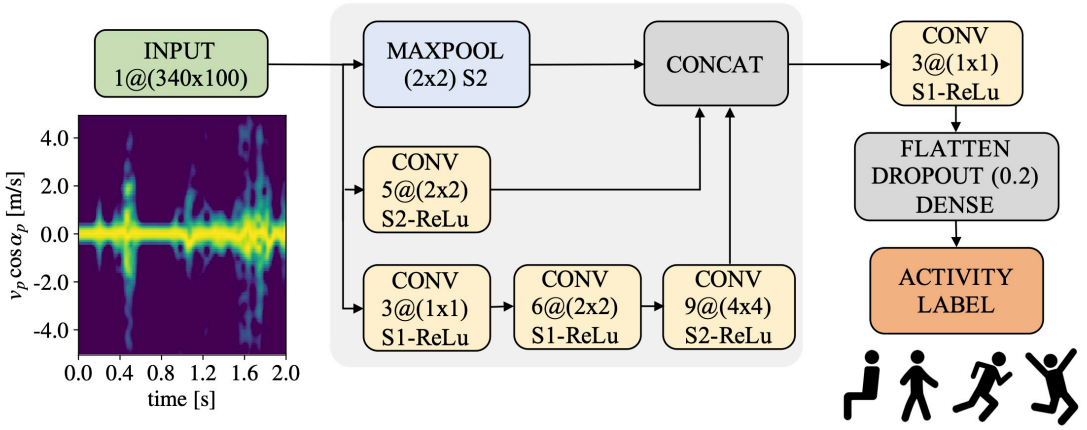


Figure 2.4: SHARP classification architecture for a single antenna system. The input of the network consists of 340 (about two seconds) long Doppler traces with 100 velocity bins. [CONV maps@(kernel)] blocks are convolutional filters with stride S#, followed by ReLu activation functions.

To start with, the input is fed to a simplified Inception module with dimensional reduction. This block extracts significant features from the Doppler trace at several scales by using layers with different kernel sizes in a parallel fashion. The structure is inspired by the *reduction block* proposed for the Inception-v4 neural network in [76], and consists of three branches combining max-pooling (MAXPOOL) and convolutional (CONV) layers. Overall, the proposed neural network has $128,535$ parameters. In the interest of obtaining a lightweight model, we do not use the full 17 blocks Inception-v4 network which consists of 43 million parameters, but leads to negligible performance improvements. The $N_w/2 \times N_D/2$ dimensional feature maps obtained at the output of each branch of the Inception module are concatenated and passed to the following convolutional filter with 1×1 kernel, used to reduce the number of feature maps from 15 to 3. Hence, the output of such filter is flattened and passed to a fully connected (dense) layer with five output neurons, one for each activity class. The Dropout technique is used as a regularization strategy, randomly zeroing 20% of the elements in the flatted vector preceding the dense layer. The Inception module was selected among other candidate learning architectures as the subject's movements introduce both small and large scale variations in the Doppler spectrogram (see, e.g., Figure 2.3). As humans, we are able to detect and visually extract patterns at different scales, but this is not straightforward for a neural network, where each kernel captures features at a specific scale, depending on its size. The Inception module tries to mimic such human capability by simultaneously processing the input through kernels having different sizes. As a consequence, in our case, the combination of the features extracted by an Inception module is expected to better represent the input than using a single kernel.

The classifier is trained in a supervised manner on data collected from a single indoor environment, using the cross-entropy loss function. The Doppler traces collected from the different antennas are used without any distinction among them, i.e., they are all added to a unique train-

ing set, without keeping track of the antenna that generated them. Using the trained architecture of Figure 2.4, each input trace is associated with the activity having the highest score in the output five-dimensional vector, referred to as *activity vector*.

2.5.2 Decision fusion for the multiple antenna system

At runtime, the trained classification engine from the single antenna system is independently applied to the Doppler stream gathered by each antenna. This returns N_{ant} independent classification outcomes that are combined as we now explain. In detail, for each antenna we obtain a five-dimensional *activity vector* (the classifier output) and an *activity label*, corresponding to the largest element in the activity vector. When at least $N_{\text{ant}} - 1$ activity labels agree on a certain activity, there is a clear winner, and the Doppler trace is associated with that activity. Otherwise, an overall *decision vector* is computed by summing, element-wise, the N_{ant} activity vectors. The trace is then associated with the activity having the highest score in this decision vector.

2.6 Experimental setup

In this section we present the experimental setup designed to train and validate SHARP. We first introduce the CSI extraction method (Section 2.6.1), and then present the measurement scenarios and campaigns selected for building the dataset (Section 2.6.2).

2.6.1 Nexmon extraction tool

Although most commercial Wi-Fi chipsets can potentially generate CSI data, few manufacturers make this data available to developers and researchers, especially for modern chipsets. Hence, the majority of the Wi-Fi-based HAR works in the literature have used outdated chipsets, for which some CSI extraction tools have been developed over the years: widely used ones are [77], [78], that target network interface cards implementing the IEEE 802.11n protocol. Recently, as part of the Nexmon project [73], [79], [80], it has been released a firmware patch allowing the extraction of CSI from specific Broadcom/Cypress Wi-Fi chipsets. Here, we use the Nexmon CSI extraction tool presented in [7] to obtain CSI data from an Asus RT-AC86U IEEE 802.11ac Wi-Fi router. The extraction tool is compatible with the very-high-throughput mode, defined by IEEE 802.11ac, working with a total bandwidth of 80 MHz. Each CSI sample results in complex-valued channel information from 242 data sub-channels for each transmit-receive antennas pair. In our experiments, with one transmitter antenna and four at the monitoring device, each CSI sample corresponds to four vectors of 242 complex values. Although the total number of sub-channels at 80 MHz is 256, each antenna vector has 242 components as the CFR is only provided for data sub-channels, namely sub-channels whose indexes are $\{-122, \ldots, -2\}$ and $\{2, \ldots, 122\}$, i.e., no CFR value is provided for the control sub-channels. Moreover, the values returned by the tool on the sub-channels from -63 to 122 need an inversion on the sign, probably due to hardware artifacts.

In the next sections, the Asus router that estimates the CFR is referred to as *monitor* device. The name reflects the status of the wireless interface of the router, that should be set in monitor mode to capture the packets sent over-the-air by other access points and/or stations transmitting in its proximity.

2.6.2 Dataset acquisition and organization

In order to analyze the effects of different room geometries and static obstacles, we collected CSI samples in three different environments, a bedroom (Figure 2.5-a), a living room (Figure 2.5-b) and a University laboratory (Figure 2.6), where one person moves within the area. Specifically, we obtain data from three volunteers (a male, `P1`, and two females, `P2`, `P3`) while they are walking or running around, jumping in place, or sitting somewhere in the room.

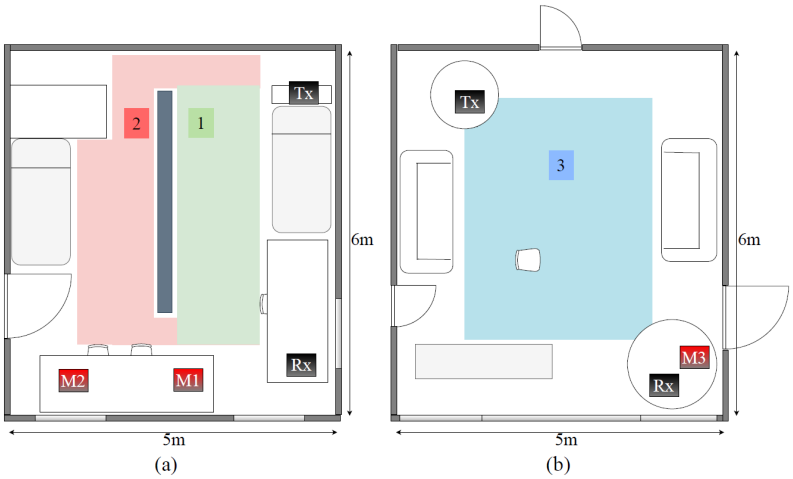


Figure 2.5: Monitored environment for (a) sets $S1$ - $S5$, bedroom with a bookcase on the middle; (b) set $S6$, living room. `Tx` and `Rx` denote the transmitter and the receiver, respectively. `Mj`, `j` $\in \{1, 2, 3\}$, denotes the monitor station.

We recall that, during a Wi-Fi communication, the CSI samples are computed at the receiver side based on known packet preambles and used to estimate the channel conditions. CSI samples can also be estimated by any other monitor node that has access to the wireless channel, exploiting so called "signals of opportunity". In our setup, the CSI samples are collected by a monitor node, implemented on an Asus router equipped with $N_{\text{ant}} = 4$ antennas and running the Nexmon firmware (see Section 2.6.1). To generate the signals of opportunity, we set up a Wi-Fi transmission link using two Netgear X4S AC2600 Wi-Fi routers, equipped with Qualcomm Atheros chipsets. The packets are transmitted through a single antenna, by using a fixed modulation and coding scheme (namely, MSC 4), by disenabling frame aggregation, and by setting the source rate to 173 packets per second. Since the Nexmon tool is configured for reading CSI samples on data packets only (i.e., by neglecting the acknowledgement frames sent by the receiver), a new channel estimate is generated every $T_c \simeq 6 \times 10^{-3}$ s.

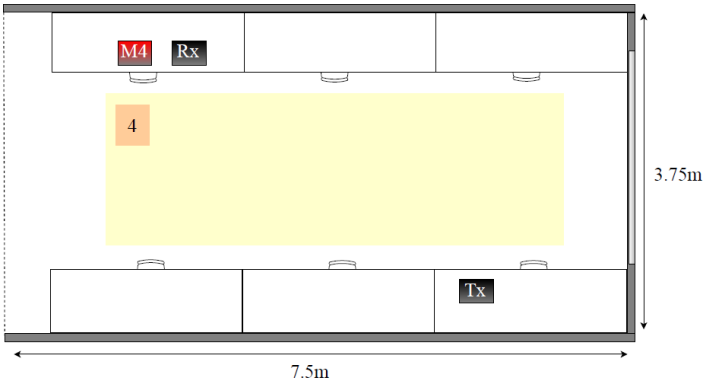


Figure 2.6: Monitored environment for set *S7*. The desks are fully of computers and monitors. `Tx` and `Rx` denote the transmitter and the receiver, respectively. `M4` denotes the monitor station.

Figure 2.5 and Figure 2.6 display the positions of transmitter, receiver and monitor routers in the different environments. The black boxes represent the transmitter (`Tx`) and the receiver (`Rx`), while the red boxes labeled with `Mj` (with `j` $\in \{1, \ldots, 4\}$) indicate the position of the monitor station in different measurement sets. The activities are performed in the areas identified by a color and a number. Note that in the bedroom (Figure 2.5-a), the direct path between `Tx` and `M2` is occluded by the bookcase in the middle of the room (grey rectangle in the figure).

We performed several measurement campaigns and we grouped them into seven sets, *Sj* with $j \in \{1, ..., 7\}$ each corresponding to a different triplet of environment-day-person. For each set, Table 2.1 provides the position of the monitoring station (`Mj`), the area where the activities take place (identified by the color), and the person performing them (`Pi`). We also include an indication of whether there is a direct path between the transmitter and the monitor stations. The configuration for sets *S*1-*S*2 is the same apart from the day of measurement. Set *S*1 is used to train the model, while set *S*2 is used to test the generalization over different days. In *S*3 we monitor the same environment as the previous sets, but in a different day and with a different person performing the activities. For sets *S*4 and *S*5, the person is required to move in both area 1 and 2 of the bedroom depicted in Figure 2.5-a. In these configurations, the direct path between the transmitter and the monitor is disturbed by the bookcase. Set *S*6-*S*7 are collected in two different days and environments. *S*7 represents the more challenging situation, in which no element (room, day, person) is in common with the scenario used for the training set.

Each measurement campaign involves 120 seconds of data for each activity, plus an additional trace of 120 seconds of data collected when the room is empty. Activities are repeated continuously by the volunteers during the trace acquisition time. The campaigns have been performed in different days through several months (April-December, 2020), resulting in a considerable time diversity. Overall, we collected nearly 120 minutes of CSI data, consisting of the CFRs estimated at all the four antennas of the monitor station.

set	monitor position, `Mj`	person, `Pi`	direct path	type
*S*1	`M1`	`P1`	yes	train-test
*S*2	`M1`	`P1`	yes	test
*S*3	`M1`	`P2`	yes	test
*S*4	`M2`	`P1`	no	test
*S*5	`M2`	`P2`	no	test
*S*6	`M3`	`P1`	yes	test
*S*7	`M4`	`P3`	yes	test

Table 2.1: Measurement conditions. For each set *Sj* we specify the position of the monitor station (`Mj`) (see Figure 2.5 and Figure 2.6), the person (`Pi`) performing the activity, and the presence of a direct path between the transmitter and the monitor. The last column indicates whether the set is used for training the proposed HAR algorithm or testing its performance.

2.7 Experimental results

SHARP has been tested on the scenarios detailed in Table 2.1. The adopted communication and processing parameters are summarized in Table 2.2 and are presented in Section 2.7.1. The performance is discussed in Section 2.7.3.

communication and processing parameters	
monitored channel	IEEE 802.11ac ch no. 42
OFDM sample duration, T	3.2×10^{-6} s
modulation and coding scheme	MCS 4
channel estimates interval, T_c	$\simeq 6 \times 10^{-3}$ s
no. OFDM sub-channels, M	256 (245 used)
no. ch. estimates for Doppler computation, N	31
no. bins in a Doppler vector, N_D	100
no. Doppler vectors per window, N_w	340
no. monitoring antennas, N_{ant}	4

Table 2.2: Summary of parameters related to the communication setup, and the Doppler traces computation.

2.7.1 Pre-processing steps

We considered some pre-processing operations for extracting the features of interest from the dataset. For each CSI sample, we divided the CFR values by the mean amplitude over the 242 monitored sub-channels to remove unwanted amplifications. Then, the phase sanitization algorithm presented in Section 2.4.1 was applied, fixing $\lambda = 10^{-1}$. We reconstructed the CFR values on the three central sub-channels together with the other 242 using Eq. (2.20), thus obtaining a CFR complex-valued vector (amplitudes and phases) of 245 components. The Doppler vectors have been computed considering $N = 31$ subsequent sanitized CFR samples. The velocity resolution was increased by zero-padding the signal out to $N_D = 100$ points before applying the Fourier transform. A threshold was used on the resulting Doppler vectors to remove noisy contributions with power smaller than 12 dB. Finally, the Doppler trace acting as input feature for the

HAR system was built by stacking $N_w = 340$ consecutive Doppler vectors. The Doppler vectors were generated for each CSI acquisition by using a sliding window mechanism. Therefore, the complete Doppler trace lasted roughly $340 \cdot T_c = 2$ s of measurements.

2.7.2 On the selection of the phase sanitization approach

The removal of the phase offsets is necessary to obtain meaningful Doppler traces. The unwanted offsets, and in particular the CFO, disrupt the Doppler shift information contained in the CFR data, making its extraction impossible. Prior to developing our own phase sanitization method (see Section 2.4.1), we investigated existing phase sanitization techniques from the literature. The comparison between these strategies is not straightforward due to two main reasons. First, the target phase (i.e., the one without offsets) can be hardly obtained through a mathematical model, as the multi-path propagation is an environment-specific process that depends on many factors. One may think of evaluating the effectiveness of a phase sanitization algorithm by developing a simulator of the Wi-Fi channel, including the phase offsets introduced by the hardware imperfections. However, even with such simulator a fair comparison among the different strategies would be difficult, as they use different reference systems for the sanitized output signal. Specifically, the extensively adopted conjugate-multiplication between antennas (see, e.g., [44], [59]–[61]), uses one of the antennas as the reference for the processed signal. The method in [49], instead, refers the signal to its static component, while our proposed approach considers the strongest channel path as the reference.

As our main objective is to develop an environment-independent algorithm for HAR, we compare these phase sanitization approaches based on the quality of their features, i.e., by analyzing their respective Doppler spectrograms. Figure 2.7 shows the Doppler spectrogram computed from a 6 second long CFR trace and its sanitized versions using the just mentioned techniques. In the considered time window, a person sits for the first 3 seconds and then starts running inside the room. The "original signal" is clearly too noisy to be useful. The Doppler trace obtained with our sanitization algorithm (see Section 2.4.1) is referred to as "ref. main path", while the "ref. one antenna" and "ref. static component" respectively refer to the approaches in [44] and [49]. The use of one antenna as the reference allows removing most of the phase offset contributions in the CFR, but it also removes some relevant information related to the environmental changes. This is due to the conjugate multiplication operation, which can lead to destructive interference between the signal to be cleaned and the reference. On the other hand, when the reference is the static component, the processed signal contains some noisy contributions which are probably associated with energy leakage between the static and the dynamic part. This prevents the correct estimation of the reference used for phase sanitization. We experimentally verified that these previous algorithms do not lead to good HAR performance. These drawbacks led us to the development of our approach, detailed in the earlier Section 2.4.1.

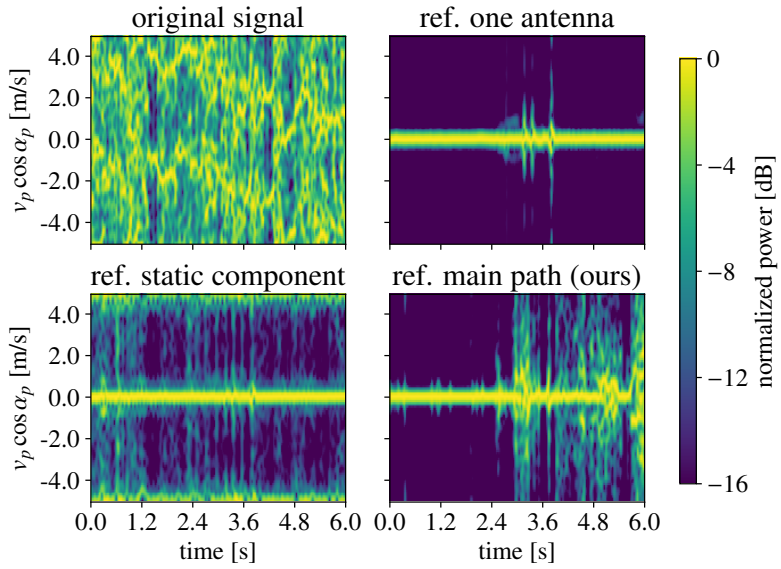


Figure 2.7: Doppler spectrograms computed from a six seconds long CFR signal, where the activity performed are sitting ($[0, 3]$ s) and running ($[3, 6]$ *s*). The subfigures show the Doppler extracted from the original signal (containing offsets) and those obtained with three phase sanitization techniques, labeled according to the reference (ref.) used.

2.7.3 HAR algorithm training and performance assessment

The HAR algorithm presented in Section 2.5 has been trained by using the features extracted on set $S1$. Specifically, 60% of the data in $S1$ makes up the training set, while the remaining 40% is evenly split between the validation and the test sets. The other measurement sets, i.e., $S2$-$S7$, are only considered in the test phase. The classification accuracy and the F1-score obtained on each of the test sets are reported in Table 2.3. Note that, for $S1$, the evaluation is performed on the test set only, i.e., on new data never seen during training. Overall, the mean recognition accuracy is higher than 95%, reaching almost 100% when the environment and location of the monitor node ($M1$) remain the same of the training data, regardless of day of measurement and the person performing the activity ($P1$ or $P2$). We still obtain a good accuracy, around 97%, when the direct path between the transmitter and the monitor stations is blocked (sets $S4$ and $S5$). In this situation, the running activity is sometimes confused with the walking one, as revealed by the F1-score associated with these classes. However, this is a reasonable limit of our approach, as the activities are recognized by considering the velocity of different parts of the human body (i.e., the scattering sources), which can be very similar when people walk or run in indoor spaces. We also believe that a small confusion between the running and walking activities is admissible in almost all the fields of application of a Wi-Fi sensing system. For example, in a residential scenario, it is more relevant to correctly distinguish between static and dynamic activities, rather than providing details about the type of movement.

		empty	sitting	walking	running	jumping	**mean**
$S1$	**accuracy (%)**	100	100	100	100	97.93	99.59
	F1-score	1	1	0.997	0.996	0.990	0.996
$S2$	**accuracy (%)**	100	100	100	99.28	99.64	99.78
	F1-score	1	0.998	0.999	0.996	0.998	0.998
$S3$	**accuracy (%)**	100	100	96.64	100	100	99.33
	F1-score	1	1	0.983	0.984	1	0.993
$S4$	**accuracy (%)**	100	100	99.84	93.27	94.63	97.55
	F1-score	1	1	0.965	0.941	0.972	0.976
$S5$	**accuracy (%)**	100	100	98.58	84.93	99.45	96.59
	F1-score	1	1	0.923	0.911	0.997	0.966
$S6$	**accuracy (%)**	99.76	100	100	100	99.20	99.79
	F1-score	0.999	0.999	0.999	0.997	0.996	0.998
$S7$	**accuracy (%)**	100	100	81.24	98.71	100	95.99
	F1-score	1	1	0.89	0.909	1	0.960

Table 2.3: SHARP performance. The accuracy and the F1-score are reported for each of the five classes, along with their average value.

For testing the generality of our algorithm under different environments, we consider the measurement sets *S*6 and *S*7. When the person is the same of the training data, the average HAR accuracy approaches 100% while it decreases to 95.99% when both the person and the room change. Again, the lower accuracy is achieved for the walking activity, which is wrongly classified as the running one, as displayed in the normalized confusion matrix in Figure 2.8.

2.7.4 HAR algorithms performance comparison

In Figs. 2.9a-2.9b, SHARP is compared against three HAR systems from the literature: DeepSense [47], EI [48] and MatNet-eCSI [49]. DeepSense and EI are used in place of the single antenna classifier described in Section 2.5.1 (i.e., before the decision fusion step) and rely on the sole CSI amplitude. MatNet-eCSI, instead, considers phase and amplitude information, and takes as input their combination at all the antennas. The three approaches use different neural network architectures. In the interest of a fair comparison, we trained them with the same portion of the set *S*1 used to train SHARP. Note that MatNet-eCSI would require two training steps: the first performed on the training scenario, whereas the second on the test scenario, to fine tune the classification architecture to the new setup. For a fair comparison, in our evaluation we only considered the first step (using *S*1 data), i.e., without benefiting from additional data collected in the test scenarios. Specifically, we respectively used the training and the validation portions of *S*1 as the *reference* and the *target data* for the MatNet-eCSI neural network. For the EI algorithm, we considered the three different days containing the measurements for the training set *S*1 as the different domains for its adversarial learning framework. Then, for DeepSense and EI, the decision fusion approach of Section 2.5.2 is applied at runtime to combine the output of the classifiers

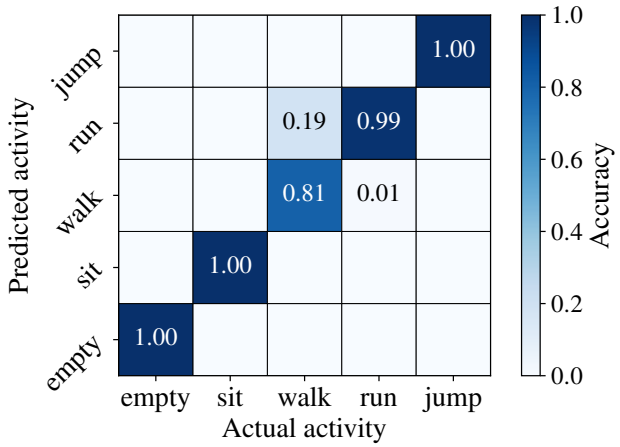
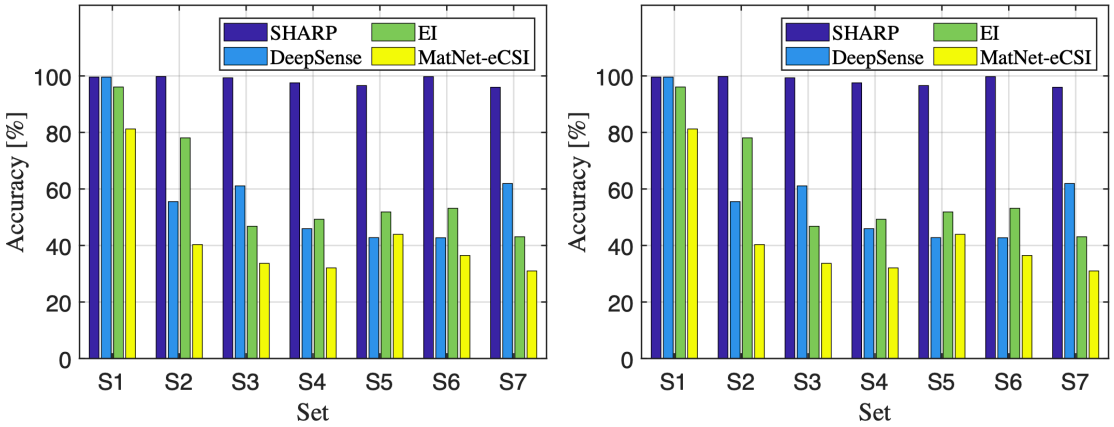


Figure 2.8: SHARP normalized confusion matrix for test set *S7*. Environment, day and person change with respect to the training.

operating on the single antenna data and obtain the final activity label. As MatNet-eCSI already takes the combination of features from all the antennas as input to its neural network classifier, the decision fusion is not performed for this scheme.



(a) Average accuracy of SHARP compared against the three considered approaches from the literature.

(b) Average F1-score of SHARP compared against the three considered approaches from the literature.

Figure 2.9: SHARP performance in the worst case scenario, i.e., *S7* and comparison against three HAR algorithms from the literature – namely DeepSense [47], EI [48], and MatNet-eCSI [49] – for all the configurations in Table 2.1.

From Figs. 2.9a-2.9b, we observe that the performance of all the considered competing algorithms is in line with that of SHARP when tested on set *S1*, while it substantially degrades in all the other cases, even when data are collected in the same environment used for training, but on a different day (e.g., set *S2*). These results demonstrate that the sole amplitude information (DeepSense and EI) or its combination with the phase (MatNet-eCSI) are insufficient for the

classifiers to generalize well across different scenarios. Indeed, as shown and discussed in Section 2.4, the CFR amplitude and its phase are environment-dependent and are affected by small scene variations over days, such as changes in the positioning of the monitor node, room obstacles, and objects. Conversely, SHARP generalizes across different days, environments and persons, by reliably distinguishing between static and dynamic situations. This is achieved by: i) exploiting the Doppler shift as the input feature for HAR, and ii) using the Inception module as the core of the learning-based classifier, to automatically extract coarse- and fine-grained details from the Doppler spectrograms.

2.8 Concluding remarks

In this chapter, SHARP, a novel low-cost system for HAR in indoor spaces has been presented and experimentally validated. This system analyzes Wi-Fi signals scattered into the environment and runs on COTS Wi-Fi routers, which are usually available in indoor spaces for communication purposes, thus making the deployment and maintenance of an ad-hoc sensing infrastructure unnecessary. The use of Wi-Fi routers as environmental sensors is enabled by the possibility of reliably estimating the wireless channel frequency response while receiving background data traffic.

In contrast with previous solutions, we implemented a *robust* HAR system that does not require complex and periodic calibrations, i.e., once trained, it can be used at run-time for a completely unseen setup (new environment and individuals). SHARP revolves around the idea of processing the CSI data gathered by a monitor access point to quantify the Doppler effect due to the presence of moving objects over time. Indeed, the Doppler shift describes the velocity of the scattering points (i.e., the human body parts), whose temporal trace depends on the specific activity and is neither affected by the environment geometry, nor by static objects. A learning based algorithm, working on the Doppler trace, has been proposed to distinguish among different activities.

Four coarse-grained human activities, i.e., sitting, walking, running and jumping are considered, together with the empty space recognition. The robustness of SHARP has been challenged considering seven combinations of environments, access point locations, individuals and days. In each scenario, several measurement campaigns have been conducted, using a four-antennas IEEE 802.11ac router by Asus, operating on a frequency band of 80 MHz. Experimental results show that our system achieves a very high accuracy in all the considered scenarios. In the most challenging situation, i.e., when the person and the environment change with respect to those used at training time, the average accuracy is 96%. Moreover, the system outperforms state-of-art methods based on the analysis of the CSI amplitude and phase. Small errors are observed only when working with unseen persons and when distinguishing between walking and running. These errors somehow represent the fundamental limits of the approach, because they are due to the behavioral differences among different persons, or to the movement similarities in walking and running activities.

3

Radio fingerprinting of MU-MIMO devices through Wi-Fi channel state information

3.1 Introduction

As introduced in Section 1.1, the Wi-Fi CFR is not only affected by obstacles in the physical propagation environment. The hardware inaccuracies of both the transmitter and the receiver devices introduce small-scale variations in the CFR estimated at the receiver side. In this chapter, we proceed with our discussion on *communication-assisted sensing* applications by presenting `DeepCSI`, the first physical layer fingerprinting algorithm for multi-user multi-input, multi-output (MU-MIMO) devices. `DeepCSI` leverages the small-scale variations introduced in the CFR by the hardware inaccuracies to perform physical layer fingerprinting of Wi-Fi terminals through off-the-shelf devices such as laptops, without the need for more expensive software-defined radio (SDR) devices.

In recent years, radio fingerprinting (RFP) has attracted significant attention as reliable and effective spectrum-level authentication technique [81]–[87]. RFP consists in the computation of a unique "fingerprint" of the device directly at the waveform level, by exploiting naturally-occurring circuitry imperfections [88]. Although RFP for physical (PHY) Wi-Fi authentication has been explored, existing approaches require SDR devices to extract RFP features. This may ultimately prevent widespread adoption, since SDRs require expert knowledge and are usually more expensive than off-the-shelf devices. Moreover, existing work has tackled Wi-Fi fingerprinting up to the legacy 802.11 a/g/b standards, which do not support multi-input, multi-output (MIMO) techniques. However, newer Wi-Fi releases such as 802.11ac/ax and the upcoming 802.11be will heavily rely on MU-MIMO techniques to deliver significantly higher throughput than previous standards [89]–[91]. Thus, it is still unknown whether existing RFP techniques can be applied in

the significantly more complex scenario of MU-MIMO, where inter-user interference (IUI) and inter-stream interference (ISI) can significantly decrease the quality of the fingerprint itself.

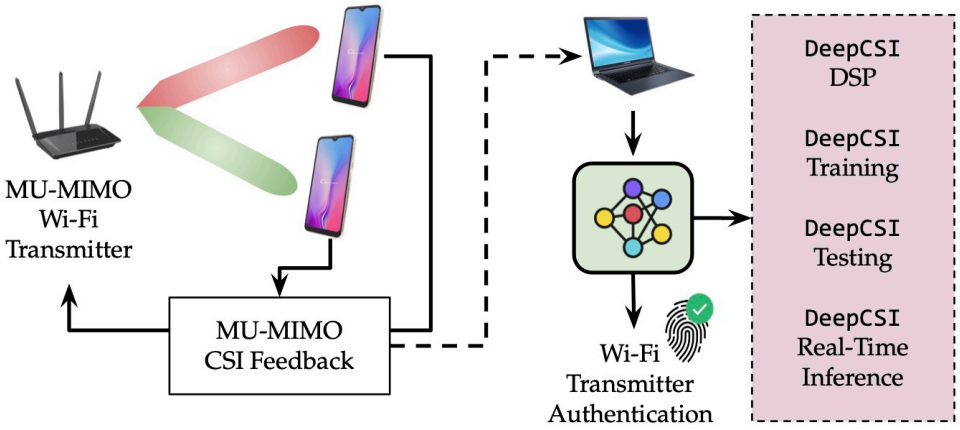


Figure 3.1: Main operations of `DeepCSI`.

To fill the current research gap, in this chapter we propose `DeepCSI`, a brand-new technique for RFP of Wi-Fi devices which is summarized in Figure 3.1. The core intuition behind `DeepCSI` is that the circuitry imperfections in the transmitter's radio interface will percolate onto the MU-MIMO CSI feedback sent by the receiver to the transmitter to perform beamforming. By demodulating this PHY-level feedback and performing deep learning techniques on a processed version of the feedback, an observer can fingerprint the transmitter without the need of SDR capabilities. The core concern of performing RFP without direct CSI access is that it is unknown whether the imperfections will actually percolate onto the beamforming feedback matrix. Furthermore, as pointed our earlier, the effect of IUI and ISI may further obscure the fingerprint, which is not present in non-MIMO transmissions. Finally, it is crucial to evaluate our PHY fingerprinting technique as a function of different channels and different transmitter-receiver positions, since these can significantly undermine the fingerprint [87].

To summarize, in this chapter

- we propose `DeepCSI`, a novel framework that leverages standard-compliant beamforming feedback matrices to authenticate MU-MIMO Wi-Fi devices. Conversely from prior work, `DeepCSI` does not require SDR technologies and can be run on any low-cost Wi-Fi device. The core intuition behind `DeepCSI` is that imperfections in the transmitter's radio circuitry will also be present in the beamforming feedback matrix, and thus explicit CSI computation through SDR is not needed. `DeepCSI` uses deep learning of the CSI vectors to learn the device-unique imperfections located in the CSI;

- we extensively evaluate the performance of `DeepCSI` through a massive data collection campaign performed *in the wild* with off-the-shelf equipment, where 10 Wi-Fi radios emit MU-MIMO signals to multiple receivers located at different positions (and thus, with different beam patterns). Experimental results indicate that `DeepCSI` is able to correctly identify

the transmitter with an accuracy above 97%. Furthermore, we show that our technique achieves accuracy of at least 46% when tested with a different channel condition, which becomes 73% when training on mixed position data. We evaluate the impact of the feedback quantization error on the performance – where quantization is applied for transmission efficiency reasons as per the Wi-Fi standards [13], [92] – observing an accuracy increase of up to 63% when changing the feedback PHY parameters. Finally, we show that `DeepCSI` achieves at least 17% more accuracy than methods based on CSI phase cleaning, since the latter partially remove the imperfections due to the hardware circuitry.

3.2 Background, related work, and challenges

Thanks to their capability of identifying transmitters without the need of computation-hungry cryptography techniques, RFP techniques have received a significant amount of attention from the research community [82]–[84], [86], [93]–[95]. While early work has demonstrated the feasibility of RFP, it has focused on the extraction of complex hand-tailored features, which do not scale well with the device population, or work in ad hoc propagation settings only. Among the earliest papers in the field, Brik *et al.* [94] show that 99% accuracy can be achieved through offset-based features on a device population of 130 devices. However, experiments were performed in an anechoic chamber, which is an almost ideal propagation environment. Among the first works on Wi-Fi-specific RFP, Vo *et al.* [86] propose RFP techniques that extract features from the scrambling seed, the level of frequency offset and transients between symbols. However, the models achieve accuracy up to 50% on 100 devices. The authors in [82], instead, demonstrated that up to 54 ZigBee devices can be fingerprinted with about 95% accuracy through PSK transients. More recently, Zheng *et al.* [81] studied and evaluated in a testbed of 33 devices a model-based approach to summarize imperfections in the modulation, timing, frequency and power amplifier noise. It is not clear, however, whether the approach in [81] generalizes to different channel environments.

In stark contrast with early work, recent RFP papers have leveraged deep learning techniques to fingerprint wireless devices [85], [96]–[99]. A key advantage of deep learning techniques is that they are able to perform feature extraction and classification at the same time, thus avoiding manual extraction of device-distinguishing features. For example, Das *et al.* [99] and Merchant *et al.* [98] deep neural networks (DNNs) achieve more than 90% accuracy with a population of 7 ZigBee devices and 30 LoRa devices. To further increase accuracy, [85], [97] proposed the introduction of artificial impairments at the transmitter's side. However, without compensation, this approach inevitably increases the bit error rate (BER). The usage of complex-valued convolutional neural networks (CNNs) has been explored by Gopalakrishnan *et al.* [100], while in [96] and [101] the authors propose the usage of finite impulse response (FIR) filters to compensate for the adverse action of the wireless channel on the fingerprinting accuracy. The key limitation of existing work is that it is entirely based on SDRs, which is very specialized, expensive equipment that is not widely available in common Wi-Fi networks. Moreover, to the best of our knowledge, no prior work has tackled the issue of assessing whether RFP is feasible in MU-MIMO Wi-Fi networks. In this chapter, we address both issues at once by presenting `DeepCSI`, a framework that (i) can be

run on any off-the-shelf Wi-Fi-compliant device, and (ii) can accurately fingerprint MU-MIMO devices.

3.2.1 Challenges of MU-MIMO fingerprinting

Performing RFP of devices operating in downlink (DL) MU-MIMO mode is significantly more challenging than RFP of devices operating with omnidirectional antennas. First, transmissions are inevitably impaired by imperfect beamforming weights that do not accurately compensate the wireless channel. Moreover, (i) inter-stream interference (ISI) occurs between streams transmitted to the same receiver; (ii) inter-user interference (IUI) affects streams directed to different receivers. The time-varying behavior of both ISI and IUI complicates the identification of the device-specific imperfections. Moreover, it has been shown in prior work that the RFP process may be adversely impacted by the presence of the wireless channel [85], [87]. This reasoning led us to design a different approach for extracting effective radio fingerprints. Specifically, we use the beamforming feedback matrix $\tilde{\mathbf{V}}$ described in Section 3.3.2. The matrix $\tilde{\mathbf{V}}$ is estimated based on the very high throughput (VHT)-long training fields (LTFs) of the null data packet (NDP) that is sent in broadcast mode without being beamformed. Moreover, the VHT-LTFs are sent over the different antennas in subsequent time slots of 4 μs each. Therefore, the NDP and, in turn, $\tilde{\mathbf{V}}$, *are not affected by IUI nor by ISI*. However, since the feedback matrix is *quantized* before transmission, quantization errors are inevitable when reconstructing the matrix. In Section 3.5, we analyze the effect of the quantization error and investigate the generalization capability of our RFP approach to multiple channels and beamformee positions.

3.3 The `DeepCSI` framework

Henceforth, we will adopt the following notation for mathematical expressions. The superscripts T and † respectively denote the transpose and the Hermitian of a matrix, i.e., the complex conjugate transpose. By $\angle\mathbf{C}$, we refer to the matrix whose elements are the phases of the corresponding elements in the complex-valued matrix $\mathbf{C}$. $\text{diag}(c_1, \ldots, c_j)$ indicates the diagonal matrix with elements $(c_1, \ldots, c_j)$ on the main diagonal. The (c_1, c_2) entry of matrix $\mathbf{C}$ is denoted by $[\mathbf{C}]_{c_1,c_2}$. Finally, $\mathbf{I}_c$ denotes a $c \times c$ identity matrix while $\mathbf{I}_{c\times d}$ is a $c \times d$ matrix with ones on the main diagonal and zeros elsewhere.

3.3.1 Preliminaries on MU-MIMO in Wi-Fi

In the following, we will consider Wi-Fi devices operating with the IEEE 802.11ac/ax standards [13], [92]. These devices use *beamforming* techniques to focus the power toward the intended receiver, thus increasing the signal-to-noise ratio (SNR). The beamforming may also compensate the effect of the wireless channel from the transmitter (*beamformer*) to the receiver (*beamformee*). Specifically, when both devices in the communication link are equipped with antenna arrays (MIMO system), each pair of transmitter and receiver antennas forms a physical channel that can be exploited

for wireless communication. This allows shaping multiple beams, referred to as *spatial streams*, to transmit different signals to the beamformee, in a parallel fashion. To this end, the signals are combined at each transmitter antenna through steering weights, $\mathbf{W}$, derived from the CFR matrix $\mathbf{H}$. The CFR needs to be estimated for every OFDM sub-channel over each pair of transmitter (TX) and receiver (RX) antennas, thus obtaining a $K \times M \times N$ matrix, where M and N are respectively the number of transmitter and receiver antennas. In Figure 3.2 we report an example of beamforming for a 3×2 MIMO system. At the beamformee side, the original signals are retrieved from their combination exploiting the fact that, ideally, $[\mathbf{H}]_{\bar{\ell},\bar{i}}[\mathbf{W}]_{\ell,i} = 0$ when $\bar{\ell} \neq \ell$ or $\bar{i} \neq i$.

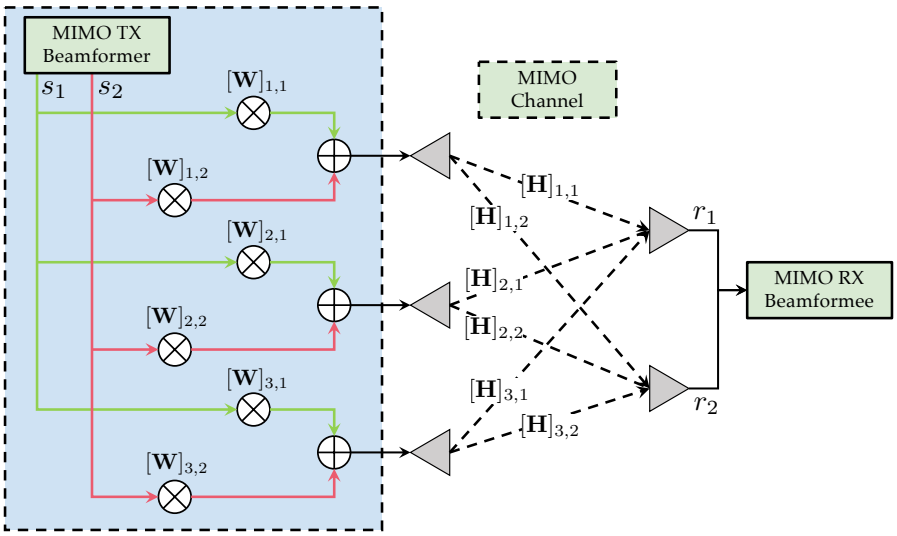


Figure 3.2: Example of beamforming for a 3×2 MIMO system. The grey triangles represent the Wi-Fi antennas. s_1, s_2 and r_1, r_2 stand for the transmitted and received signals respectively. $\mathbf{W}$ is the steering matrix containing the weights to shape the beams while $\mathbf{H}$ is the CFR matrix.

Considering the Wi-Fi OFDM transmission model detailed in Section 2.3, we can write the expression for the CFR $\mathbf{H}$ describing the propagation channels between the M transmitter and the N receiver antennas. Specifically, the (k, m, n) element of $\mathbf{H}$ is

$$[\mathbf{H}]_{k,m,n} = \sum_{p=0}^{P-1} A_{m,n,p} e^{-j2\pi(f_c + k/T)\tau_{m,n,p}}, \tag{3.1}$$

with $m \in \{0, \ldots, M-1\}$ and $n \in \{0, \ldots, N-1\}$. By knowing $\mathbf{H}$, the beamformer can generate the steering matrix $\mathbf{W}$ to maximize the power sent toward the beamformee or simultaneously send parallel data streams to multiple beamformees. These communication modes are respectively referred to as single-user multi-input, multi-output (SU-MIMO) and MU-MIMO. While IEEE 802.11n only supports SU-MIMO mode, in IEEE 802.11ac and above MU-MIMO can be enabled in the DL direction, i.e., at the access point (AP) side [92], while in 802.11ax MU-MIMO can be also enabled in uplink (UL) mode [13]. DL MU-MIMO, together with wider bandwidths and new modulation and coding schemes (MCSs), allows achieving the so called VHT enhancement.

3.3.2 Compressed beamforming feedback

In IEEE 802.11ac/ax, DL MU-MIMO is enabled by the *pre-coding* and the *channel sounding* procedures [92]. Pre-coding linearly combines the signals to be simultaneously transmitted to the different beamformees. This procedure shapes the beams focusing the power in the correct directions. The combination weights are antenna-specific and are computed based on channel sounding performed through a NDP, transmitted without beamforming. After receiving the NDP, each beamformee estimates $\mathbf{H}$ based on a VHT-LTF for each spatial stream. Next, the beamformee feeds back the matrix to the beamformer in the form of a *compressed beamforming feedback*, which is computed for each sub-channel k as follows.

Let $\mathbf{H}_k$ be the $M \times N$ sub-matrix of $\mathbf{H}$ containing the CFR samples (see Eq. (3.1)) related to sub-channel k. $\mathbf{H}_k$ is first decomposed via singular value decomposition (SVD), obtaining

$$\mathbf{H}_k^T = \mathbf{U}_k \mathbf{S}_k \mathbf{Z}_k^\dagger, \tag{3.2}$$

where $\mathbf{U}_k$ and $\mathbf{Z}_k$ are, respectively, $N \times N$ and $M \times M$ unitary matrices, while $\mathbf{S}_k$ is an $N \times M$ diagonal matrix collecting the singular values. Next, the first $N_{\text{SS}} \leq N$ columns of $\mathbf{Z}_k$ are extracted to form the complex-valued beamforming matrix $\mathbf{V}_k$ that is used by the beamformer to compute the pre-coding weights for the N_{SS} spatial streams directed to the beamformee. Note that the beamformee can be served with at maximum $N_{\text{SS}} = N$ spatial streams (see Chapter 13 of [102]). Thus, the beamformee is required to send back $\mathbf{V}_k$ to the beamformer. To do that efficiently, instead of sending the complete matrix, the beamformee derives and transmits its *compressed representation*. Specifically, the feedback consists of a number of angles obtained by converting $\mathbf{V}_k$ into polar coordinates. The transformation is based on the iterative procedure described in Algorithm 3.1,

Algorithm 3.1 $\mathbf{V}_k$ matrix decomposition

Input: $\mathbf{V}_k$
Output: $\mathbf{D}_{k,i}$ and $\mathbf{G}_{k,\ell,i}$ for $i \in \{1, \ldots, \min(N_{\text{SS}}, M-1)\}$, $\ell \in \{i+1, \ldots, M\}$

$\tilde{\mathbf{D}}_k = \text{diag}(e^{j\angle[\mathbf{V}_k]_{M,1}}, \ldots, e^{j\angle[\mathbf{V}_k]_{M,N_{\text{SS}}}})$
$\Omega_k = \mathbf{V}_k \tilde{\mathbf{D}}_k^\dagger$
for $i \leftarrow 1$ to $\min(N_{\text{SS}}, M-1)$
  $\phi_{k,\ell,i} = \angle\,[\Omega_k]_{\ell,i}$ with $\ell = i, \ldots, M-1$
  compute $\mathbf{D}_{k,i}$ through Eq. (3.3)
  $\Omega_k \leftarrow \mathbf{D}_{k,i}^\dagger \Omega_k$
  for $\ell \leftarrow i+1$ to M
    $\psi_{k,\ell,i} = \arccos\left(\frac{[\Omega_k]_{i,i}}{\sqrt{[\Omega_k]_{i,i}^2 + [\Omega_k]_{\ell,i}^2}}\right)$
    compute $\mathbf{G}_{k,\ell,i}$ through Eq. (3.4)
    $\Omega_k \leftarrow \mathbf{G}_{k,\ell,i} \Omega_k$
  end for
end for

where $\mathbf{D}_{k,i}$ and $\mathbf{G}_{k,\ell,i}$ are defined as

$$\mathbf{D}_{k,i} = \begin{bmatrix} \mathbf{I}_{i-1} & 0 & & \dots & 0 \\ 0 & e^{j\phi_{k,i,i}} & 0 & \dots & \\ & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & e^{j\phi_{k,M-1,i}} & 0 \\ 0 & \dots & & 0 & 1 \end{bmatrix}, \tag{3.3}$$

$$\mathbf{G}_{k,\ell,i} = \begin{bmatrix} \mathbf{I}_{i-1} & 0 & & \dots & 0 \\ 0 & \cos\psi_{k,\ell,i} & 0 & \sin\psi_{k,\ell,i} & \\ & 0 & \mathbf{I}_{\ell-i-1} & 0 & \vdots \\ \vdots & -\sin\psi_{k,\ell,i} & 0 & \cos\psi_{k,\ell,i} & 0 \\ 0 & \dots & & 0 & \mathbf{I}_{M-\ell} \end{bmatrix}. \tag{3.4}$$

The obtained matrices allows rewriting $\mathbf{V}_k$ as

$$\mathbf{V}_k = \tilde{\mathbf{V}}_k \tilde{\mathbf{D}}_k, \tag{3.5}$$

with

$$\tilde{\mathbf{V}}_k = \prod_{i=1}^{\min(N_{SS},M-1)} \left(\mathbf{D}_{k,i} \prod_{l=i+1}^{M} \mathbf{G}_{k,l,i}^T \right) \mathbf{I}_{M \times N_{SS}}, \tag{3.6}$$

where the products represent matrix multiplications. Note that, by construction, the last row of the complex-valued $\tilde{\mathbf{V}}_k$ matrix, i.e., the feedback for the M-th transmitter antenna, consists of non-negative real numbers. Next, the $K \times M \times N_{SS}$ beamforming matrix $\tilde{\mathbf{V}}$ is obtained by stacking the $\tilde{\mathbf{V}}_k$ matrices for $k \in \{-K/2, \dots, K/2-1\}$. Thanks to this transformation, the beamformee is only required to transmit the ϕ and ψ angles from which the $\tilde{\mathbf{V}}_k$ matrices can be reconstructed. The beamforming performance is equivalent at the beamformee when using $\mathbf{V}_k$ or $\tilde{\mathbf{V}}_k$ to construct the steering matrix $\mathbf{W}$ and, in turn, the feedback for $\tilde{\mathbf{D}}_k$ is not sent [102]. The angles are quantized for transmission using $b_\phi \in \{7, 9\}$ bits for ϕ and $b_\psi = b_\phi - 2$ bits for ψ. Next, the quantized values are packed into the VHT compressed beamforming frame and transmitted without encryption, thus allowing any device that can access the wireless channel to capture the information sent by the beamformee to the beamformer. The b_ϕ and b_ψ values can be read in the VHT MIMO control field of the frame, together with other information including the number of columns (N_{SS}) and rows (M) in the beamforming matrix and the channel bandwidth. At the beamformer, the ϕ and ψ angles are retrieved from their quantized versions $q_\phi = \{0, \dots, 2^{b_\phi}-1\}$ and $q_\psi = \{0, \dots, 2^{b_\psi}-1\}$ using

$$\phi = \pi \left(\frac{1}{2^{b_\phi}} + \frac{q_\phi}{2^{b_\phi - 1}} \right), \tag{3.7}$$

$$\psi = \pi \left(\frac{1}{2^{b_\psi+2}} + \frac{q_\psi}{2^{b_\psi+1}} \right). \tag{3.8}$$

3.3.3 `DeepCSI` workflow and learning architecture

Figure 3.3 summarizes how `DeepCSI` leverages the sounding protocol mechanism described in Section 3.3.2 to obtain a fingerprint of the IEEE 802.11ac/ax AP (beamformer). The sounding is triggered by the beamformer before sending data in DL MU-MIMO mode to the beamformees, and concludes with the transmission of the feedback angles. `DeepCSI` exploits the fact that the angles can be easily collected by any Wi-Fi compliant device by setting the Wi-Fi interface in monitor mode and using a network analyzer toolkit, e.g., Wireshark [103], to capture the packet containing the feedback. Notice that `DeepCSI` does not require the monitor device to be authenticated with the target AP. Once obtained the feedback angles, `DeepCSI` reconstructs $\tilde{\mathbf{V}}$ through Eq. (3.6).

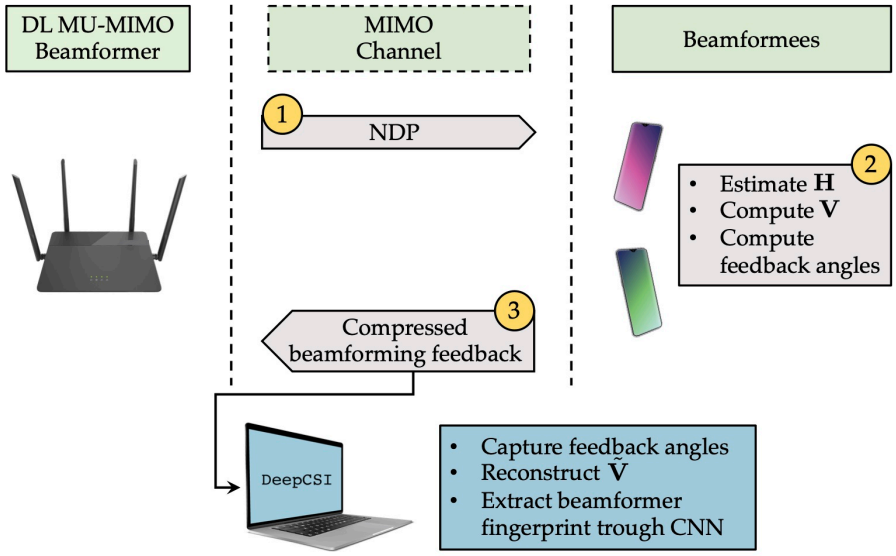


Figure 3.3: `DeepCSI` workflow. The compressed beamforming feedback computed by each beamformee as the final step of the sounding protocol is exploited by `DeepCSI` to obtain a fingerprint of the beamformer.

Next, the beamforming feedback matrix is used as input for the DNN classifier depicted in Figure 3.4 to extract the radio fingerprint of the beamformer. Notice that once trained, the DNN can be deployed and utilized in real time to authenticate the devices at the spectrum level. The elements of the feedback matrix are fed to the DNN as follows. The I/Q components of the beamforming feedback are stacked into an $N_{\text{row}} \times N_{\text{col}} \times N_{\text{ch}}$ matrix, where $N_{\text{col}} \leq K$ identifies the number of selected OFDM sub-channels, $N_{\text{row}} \leq N_{\text{SS}}$ and $N_{\text{ch}} < 2M$ refer to the columns and rows of $\tilde{\mathbf{V}}$ used for fingerprinting and the 2-factor is for the I/Q components. Note that the feedback for the last transmitting antenna consists of the sole I information as, by construction, the last row of each $\tilde{\mathbf{V}}_k$ (Eq. (3.6)) is composed of non-negative real values [92]. The learning architecture is inspired from [104] and consists of a series of N_{conv} convolutional layers followed by `selu` activation function [105], and by a max-pooling layer. The output of the previous block (in blue and green in Figure 3.4) is flattened and forward through N_{dense} dense layers with `selu` activation function. A final dense layer with `softmax` activation is used for classification. Alpha-dropout

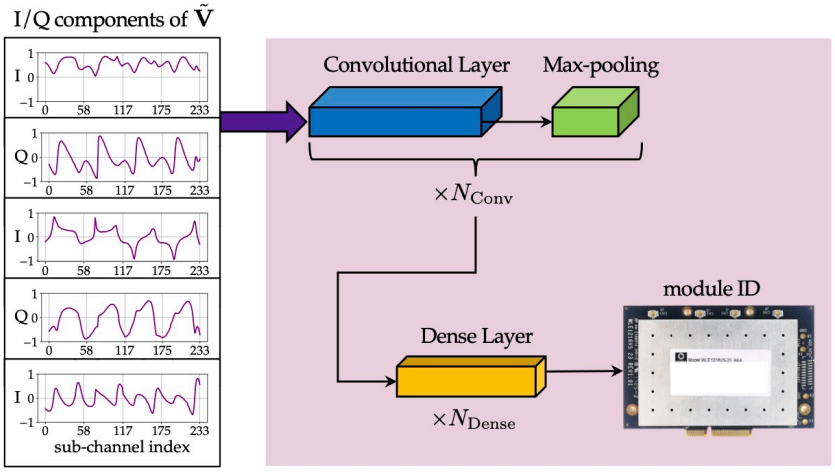


Figure 3.4: `DeepCSI` learning algorithm. The I and Q components of $\tilde{\mathbf{V}}$ serves as input for a neural network classifier that computes the beamformer fingerprint and returns, as output, the estimated Wi-Fi module ID.

layers are interposed between the dense layers. We performed hyper-parameter evaluation in Section 3.5, and established through experiments that a good set of hyper parameters is $N_{\text{conv}} = 5$ with 64 filters each, and $N_{\text{dense}} = 2$ dense layers with 128 and 64 neurons each. This architecture yields a DNN containing 159,050 trainable parameters, which is relatively small compared to state-of-the-art DNNs. The `DeepCSI` learning algorithm is trained in an offline fashion by back propagating the cross-entropy loss between the module identifier (ID) predicted by the classifier and the actual one.

3.4 Experimental setup

We evaluate the effectiveness of `DeepCSI` using off-the-shelf devices and through extensive experimental evaluation. To this end, we set up a Wi-Fi network consisting of one AP and two stations (STAs). The AP was implemented through a Gateworks GW6200 single board computer (SBC) equipped with a Compex WLE1216v5-23 IEEE 802.11ac module, as shown in Figure 3.5. The STAs were deployed at different positions as depicted in Figure 3.6 to generate different beam patterns and different SNR regimes.

Two Netgear Nighthawk X4S AC2600 routers, with 2 out of 4 antennas enabled, acted as STAs (beamformees). At the AP, $M = 3$ antennas were used to sound the channel for DL MU-MIMO transmission mode and the STAs were served with $N_{\text{SS}} = 2$ spatial streams each. Note that implementation specific constraints prevent the use of $M = 4$ for DL MU-MIMO. For the data transmission between the AP and the STAs, we used channel 42, i.e., $f_c = 5.21$ GHz with 80 MHz bandwidth. The number of OFDM sub-channels sounded is $K = 234$ as the mechanism does not consider the 14 control sub-channels and the 8 pilot ones. The AP uses the quantization parameters $b_\phi = 9$ and $b_\psi = 7$ for ϕ and ψ feedback angles, respectively. No other sources of interference

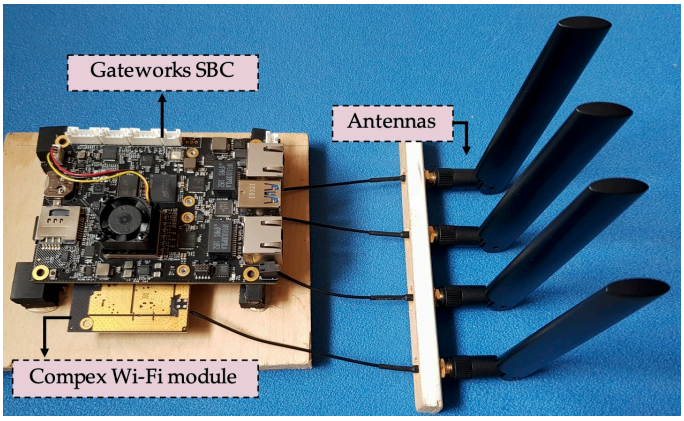


Figure 3.5: DL MU-MIMO enabled Wi-Fi AP (beamformer). The Compex WLE1216v5-23 Wi-Fi module was mounted on a Gateworks GW6200 SBC platform. Four antennas were connected to the Wi-Fi module.

are present in the considered indoor space. We generated UDP traffic in the DL direction to induce the AP to trigger the channel sounding mechanism, and collected the angles (ϕ, ψ) that were sent back by the beamformees using the Wireshark network analyzed toolkit [103] running on an off-the-shelf laptop equipped with an IEEE 802.11ac Wi-Fi card. This allows retrieving the $\tilde{\mathbf{V}}$ beamforming feedback matrices associated with each sounding operation, and computing the beamformer fingerprint as detailed in Section 3.3.3.

3.4.1 Dataset structure

The dataset consists of the beamforming feedback angles associated with $N_{\text{modules}} = 10$ different Compex Wi-Fi modules, which are the target of the proposed fingerprinting mechanism. It is collected in an indoor environment where the three entities constituting the experimental Wi-Fi network are placed as shown in Figure 3.6 and no obstacles are present between the AP and the STAs. At the AP, the SBC, the antennas and the coaxial cables remain the same across all the considered network setups, by only changing the Compex Wi-Fi module. This ensures that the fingerprint procedure only relies on the hardware imperfections of the Wi-Fi module. We collected 9 different measurements for each Compex module by changing the positions of the STAs. Specifically, the beamformees are first placed in front of the beamformer, i.e., with an angle of arrival (AoA) for the direct path of nearly zero degrees, and next moved of multiples of 10 cm respectively to the left and to the right with respect to their initial position (see the colored stars in Figure 3.6). These configurations allow obtaining data associated with different beam shaping for the ongoing DL MU-MIMO transmissions. Overall, we collected 90 traces, i.e., 9 traces for each of the 10 Compex Wi-Fi modules. Each trace contains the beamforming feedback angles sent by the two beamformees during two minutes of data transmission. Such feedbacks can be promptly grouped based on the beamformee identifier by applying a filter on the packets source address.

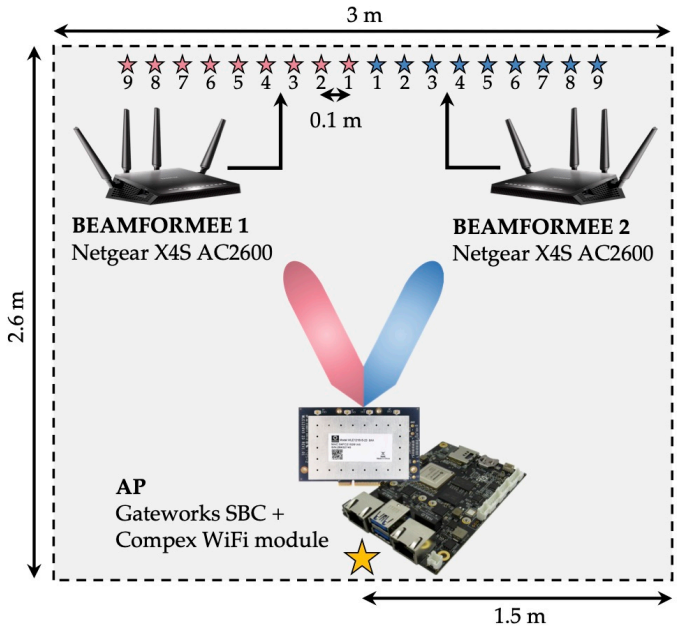


Figure 3.6: Indoor environment configuration. The position of the AP (beamformer) remains the same for all the acquisitions (yellow star). The beamformees are first placed in front of the AP and next, for each new experiment, beamformees 1 and 2 are respectively moved 10 cm to the left and 10 cm to the right. The subsequent positions of the beamformees are marked with red and blue stars respectively and labeled with a number $\in \{1, \ldots, 9\}$.

set	beamformees position																
	training									testing							
	1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
S1	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
S2	□	□	■	■	□	□	■	■	□	□	■	■	□	□	■	■	□	□
S3	■	■	□	□	■	■	□	□	■	■	□	□	■	■	□	□	■	■

Table 3.1: Training/testing sets used to assess the `DeepCSI` performance.

3.4.2 `DeepCSI` training and testing procedure

The `DeepCSI` classifier (see Figure 3.4) was trained using different PHY configurations, to evaluate its robustness in correctly identifying the beamformer device (the AP) as the position of the beamformees change. Table 3.1 summarizes the different training/testing sets that were considered, where the beamformees positions are depicted in Figure 3.6. When the same positions are considered in the training and testing phase, the first 80% of the collected data is used for training and validating the model, while the remaining 20% serves as test data. In all cases, the last 20% of training data is used for model validation. We assess the performance of `DeepCSI` on $\tilde{\mathbf{V}}$ sub-matrices. This makes it possible to evaluate the impact of using (i) different groups of transmitter antennas and spatial streams, and (ii) different portions of the radio spectrum. For (i), we vary N_{ch} and N_{row}. For (ii), we pick a subset of the K available sub-channels.

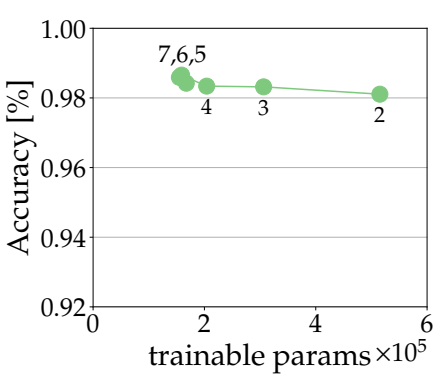
For each network configuration, DeepCSI is independently trained on the beamforming feedbacks from the two beamformees, obtaining one trained model for each of them. In this way, we evaluate a realistic usage scenario where each beamformee authenticates the beamformer based on locally acquired information, without relying on some other, possibly malicious, entities. The results considering both the beamformees are also reported for completeness.

3.5 Experimental results

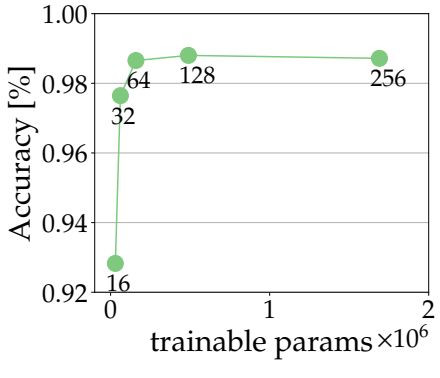
DeepCSI was experimentally evaluated on the Wi-Fi network setups of Table 3.1, evaluating the effectiveness of the extracted beamformer fingerprint for different beamformer and beamformees configurations. Next, we first briefly discuss the DNN hyper parameters selection process and then present the DeepCSI performance by varying the PHY parameters of the MU-MIMO transmission mode.

3.5.1 DeepCSI hyper parameters selection

Figure 3.7a and Figure 3.7b respectively evaluate the effect of tuning the number of convolutional layers and filters for the DNN presented in Section 3.3.3. Noticeably, the accuracy remains almost constant when varying the number of layers. Also, the accuracy increases with an increasing number of filters, at a cost of a higher network complexity (i.e., more trainable parameters). As a trade-off between accuracy and complexity, we selected $N_{\text{conv}} = 5$ convolutional layers with 64 filters each and kernel sizes of (1, 7) for the first three layers, (1,5) for the fourth and (1,3) for the last one by using the elbow method [106]. The max-pooling kernels are set to (1, 2) and the alpha-dropout between the three dense layers is applied with retain probability of 0.5 and 0.2, respectively.



(a) DeepCSI accuracy by varying the number of convolutional layers, with 64 filters each, from 2 to 7.

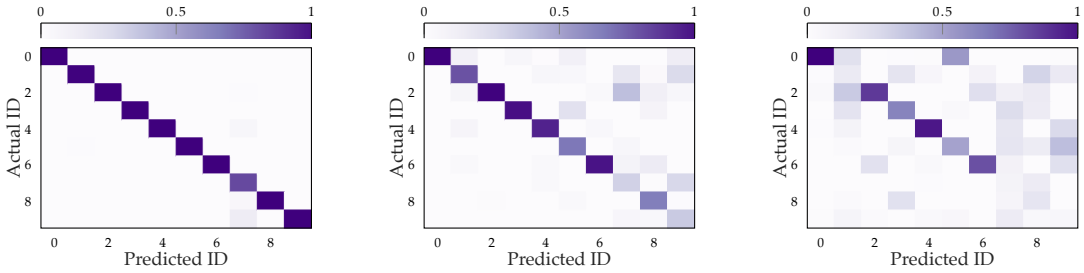


(b) DeepCSI accuracy by using 5 convolutional layers and varying the number of filters in each layer, from 16 to 256.

Figure 3.7: DeepCSI accuracy for beamformer 1, on S1 validation data, by varying the DNN parameters.

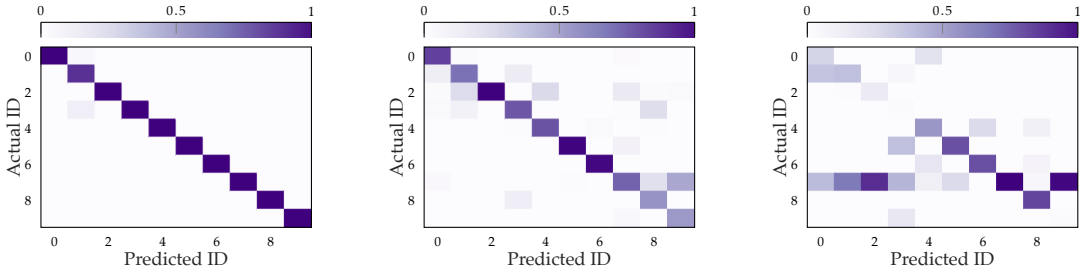
3.5.2 `DeepCSI` performance using different beamformees configurations

Figs. 3.8-3.9 show the accuracy of `DeepCSI` in correctly identifying the beamformer among the 10 Compex Wi-Fi modules in the dataset. The results were obtained using the beamforming feedback angles from a single beamformee at a time. The confusion matrices are reported for each of the three training / testing configurations in Table 3.1, where ID refers to the AP module identifier. We notice that the accuracy increases with more spatial diversity in the training data, reaching 97.68% and 98.37% for beamformee 1 and 2, respectively, when all the configurations are used at training time (see Figs. 3.8a-3.9a for set `S1`). With sets `S2` and `S3`, the beamformee positions at training and testing times differ. The lowest accuracy is obtained with `S3` (worst-case configuration). This is because `S3` is the set with the largest difference between training and testing positions. The performance improves when going from `S3` to `S2`, as the latter provides `DeepCSI` with a more balanced set of positions during training, allowing the classifier to fill the knowledge gaps by "interpolating" the patterns learned from adjacent positions. This increases the diversity among the beams to be shaped at the beamformer side and, in turn, across the collected $\tilde{\mathbf{V}}$ matrices. Hence, the network reuses information from similar beam patterns leading to an identification accuracy of 73%, even when the beamformee is at a position that was not contained in the training set (see Figs. 3.8b-3.9b).



(a) `S1`. Accuracy: 97.68% **(b)** `S2`. Accuracy: 73.19% **(c)** `S3`. Accuracy: 48.63%

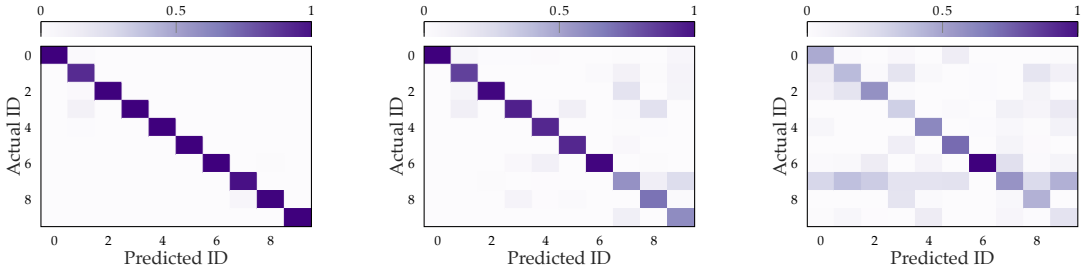
Figure 3.8: Confusion matrices for beamformee 1, 3 TX antennas, spatial stream 0. ID in this and in the following plots refers to the AP Wi-Fi module identifier.



(a) `S1`. Accuracy: 98.37% **(b)** `S2`. Accuracy: 76.55% **(c)** `S3`. Accuracy: 46.08%

Figure 3.9: Confusion matrices, beamformee 2, 3 TX antennas, spatial stream 0.

The same applies to Figure 3.10, where the beamforming feedback angles of both beamformees are used to build the training set. This allows to slightly increase the `DeepCSI` accuracy on sets `S2` and `S3`. However, using this technique in real-world scenarios poses security concerns associated with the reciprocal trustworthiness of the beamformees in a Wi-Fi network.



(a) `S1`. Accuracy: 97.96% **(b)** `S2`. Accuracy: 81.25% **(c)** `S3`. Accuracy: 49.84%

Figure 3.10: Confusion matrices, mixed beamformees, 3 TX antennas, spatial stream 0.

The impact of the number of beamformee training positions is evaluated in Figure 3.11. We report the accuracy obtained by increasing the number of positions used at training time from 1 to 9 for set `S1` and from 1 to 5 for sets `S2` and `S3`, according to Table 3.1. In all the cases, the accuracy increases with more beamformee positions in the training data, which confirms that the fingerprint is more effective when high spatial diversity is present in the training data.

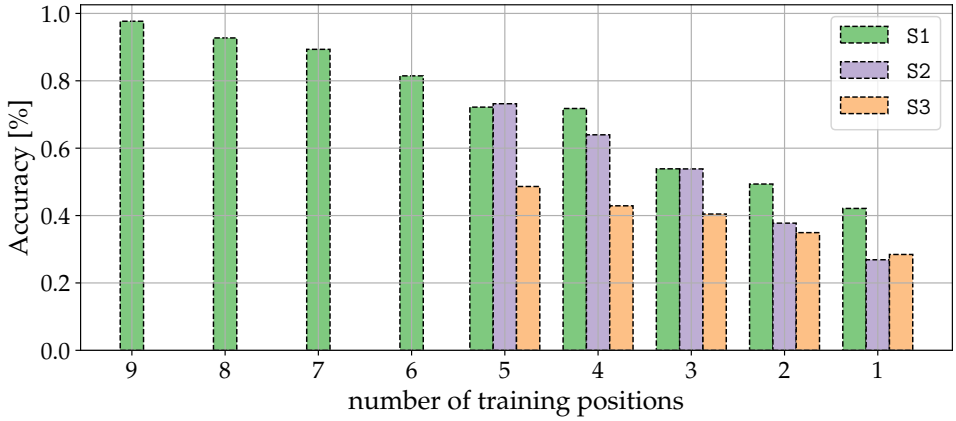
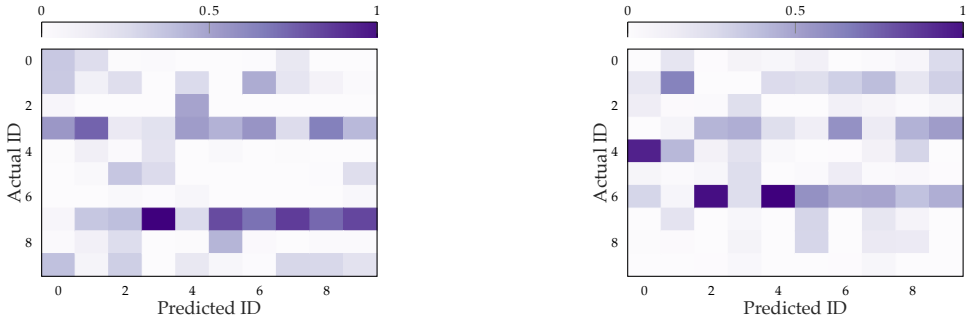


Figure 3.11: `DeepCSI` accuracy by varying the number of training positions from the considered set (see Table 3.1). Set `S1` is trained on a maximum of 9 beamformee positions while `S2` and `S3` on 5.

In Figure 3.12, we evaluate the effect of swapping the beamformees used at training and testing times for the same network configuration. We trained `DeepCSI` with data from a given beamformee and used the trained DNN model to identify the AP module from the $\tilde{\mathbf{V}}$ matrices computed by a different beamformee (for the same AP module). The learned fingerprint in this case performs poorly as matrix $\tilde{\mathbf{V}}$ captures hardware inaccuracies of both devices, i.e., the beamformer (the AP) *and* the beamformee. While a well designed learning architecture can identify with high

accuracy the beamformer when the beamformee remains the same at training and testing times, it hardly succeeds when these devices differ. We reasonably believe that in a real-world scenario the impact of this will be even stronger, as the beamformees can be from different vendors and have different hardware configurations.



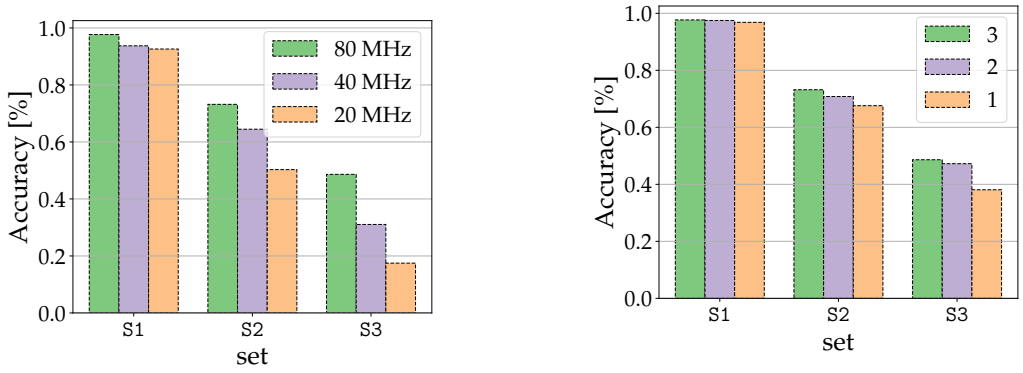
(a) Training on beamformee 1 and testing on beamformee 2. Accuracy: 10.35%

(b) Training on beamformee 2 and testing on beamformee 1. Accuracy: 12.07%

Figure 3.12: Confusion matrices for set `S1`, training on one beamformee and testing on the other, 3 TX antennas, spatial stream 0.

3.5.3 `DeepCSI` performance when varying the beamformer transmission parameters

In Figure 3.13a we compare the accuracy of `DeepCSI` when considering different portions of the radio spectrum. According to the IEEE 802.11ac OFDM channels specifications [92], from the 234 sub-channels on an 80 MHz channel, we can group sub-channels belonging to two 40 MHz and four 20 MHz channels. Therefore, from the data collected on the IEEE channel 42 at 80 MHz, we extracted 110 sub-channels for the 40 MHz channel 38 and 54 sub-channels for the 20 MHz channel 36, and assessed the performance of `DeepCSI` on these subsets. These results prove that the accuracy increases with a larger bandwidth, especially when considering the most challenging configurations `S2` and `S3`. Figure 3.13b evaluates the impact of increasing from 1 to 3 the number of transmitter antennas used to compute the fingerprint. Note that the accuracy mainly depends on the number of selected antennas and only weekly depends on their IDs. Thus, we only show results for a single selection pattern out of the possible ones for each number of antennas. The `DeepCSI` performance remains almost constant on set `S1`, while the accuracy increases on `S2` and `S3` going from 1 to 3 transmitter antennas. These results confirm that exploiting to the maximum extent the spatial diversity at the beamformer – by considering all the available OFDM sub-channels and transmitter antennas – is key to designing a robust fingerprinting algorithm.



(a) `DeepCSI` accuracy by varying the channel bandwidth, i.e., selecting respectively $N_{\text{col}} = 234$, 110, 54, out of the $K = 234$ OFDM sub-channels.

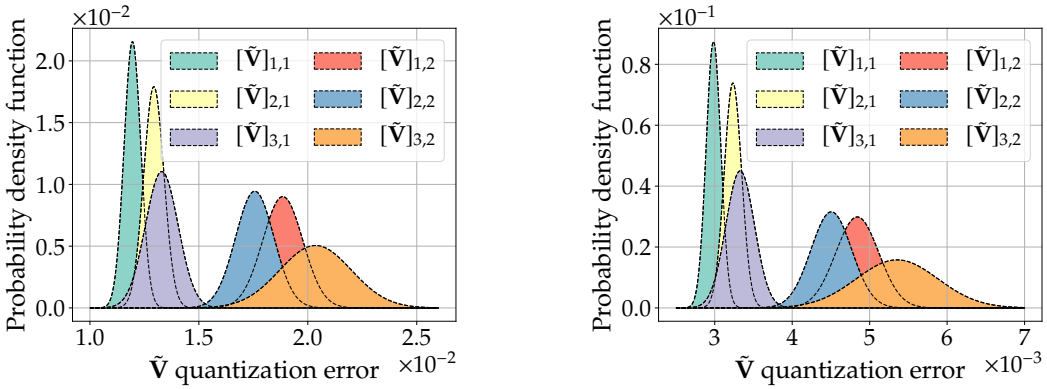
(b) `DeepCSI` accuracy by varying the number of transmitter antennas, i.e., selecting respectively $N_{\text{ch}} = 3$, 2, 1 rows of the beamforming feedback matrix $\tilde{\mathbf{V}}$.

Figure 3.13: `DeepCSI` accuracy by varying the channel bandwidth and the number of transmitter antennas, using spatial stream 0.

3.5.4 `DeepCSI` performance when changing the reference beamformee spatial stream

To evaluate the effect of changing the DNN input spatial stream on the beamformer fingerprinting accuracy, we consider the impact of the beamforming feedback angles quantization on the columns of $\tilde{\mathbf{V}}$, representing the spatial streams dimensions. From Algorithm 3.1, it follows that the impact of the quantization error increases going from the first to the last reconstructed stream. We verified this fact by simulating an OFDM MU-MIMO channel, considering the ray tracing model of [107]. We obtained the channel matrix $\mathbf{H}$ for 48,000 transmissions in MU-MIMO mode, and we derived $\tilde{\mathbf{V}}$ via SVD. Hence, we computed the q_ϕ and q_ψ quantized angles following Algorithm 3.1 and using the quantization parameters defined in the standards [13], [92]. These operations are the same performed by the beamformees to generate the feedback. Next, we reconstructed $\tilde{\mathbf{V}}$ from the quantized angles and evaluated the reconstruction error on each combination of transmitter antennas and spatial streams. We plot the probability density functions (PDFs) of the quantization error using ($b_\psi = 5$, $b_\phi = 7$) and ($b_\psi = 7$, $b_\psi = 9$) bits for quantization in Figs. 3.14a and 3.14b. We notice the reconstruction of the second column of $\tilde{\mathbf{V}}$, i.e., the second stream, is less accurate than the reconstruction of the first, for all the three transmitter antennas. This is intrinsically related to the construction of the $\mathbf{D}_{k,i}$ and $\mathbf{G}_{k,\ell,i}$ matrices from the quantized angles, and to their combination for the computation of $\tilde{\mathbf{V}}$ (see Eq. (3.6)). Indeed, the algorithm has a recursive structure by which the quantization error on the first stream propagates to the next ones, leading to worse approximations for the higher order columns of matrix $\tilde{\mathbf{V}}$. The quantization error can also be visualized from our empirical measurements. In Figure 3.15, we plot an excerpt of the $\tilde{\mathbf{V}}$ matrix reconstructed by `DeepCSI` from the quantized angles obtained at the beamformee side in static conditions. The quantization error is clearly visible for the second spatial stream (column 2 of matrix $\tilde{\mathbf{V}}$). Thus, the performance of `DeepCSI` decreases when considering as DNN

input the data associated with the second spatial stream (Figure 3.16) instead of the first one (Figure 3.8). While on set `S1` the beamformer can still be identified with high accuracy using data from the second spatial stream, when considering sets `S2` and `S3` – thus reducing the number of training positions – the beamformer fingerprint can no longer be effectively extracted, leading to a considerable drop in the classification accuracy.



(a) $b_\psi = 5$ and $b_\phi = 7$ bit.

(b) $b_\psi = 7$ and $b_\phi = 9$ bit.

Figure 3.14: PDF of the $\tilde{\mathbf{V}}$ quantization error using the two standard-compliant sets of values for the beamforming feedback angles quantization bits.

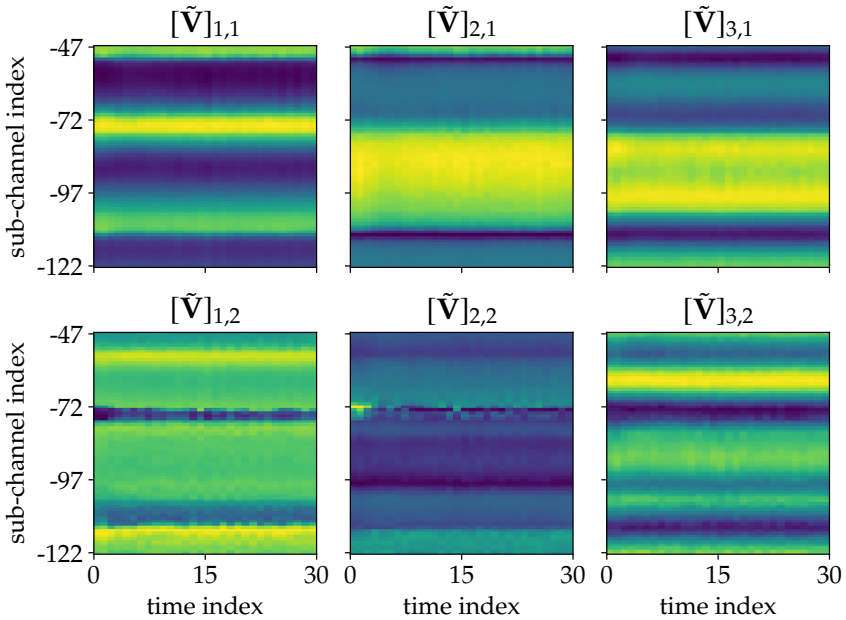
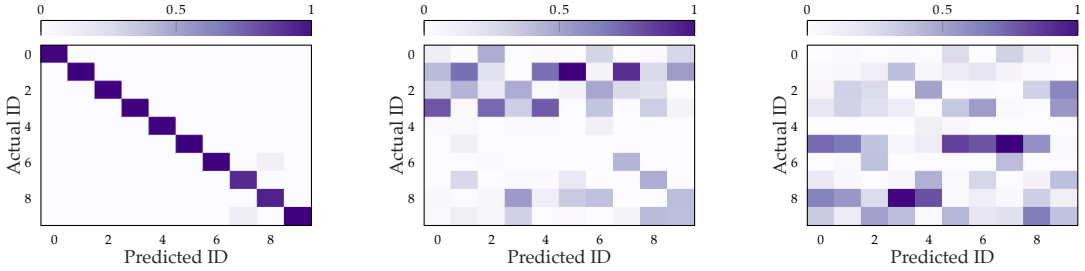


Figure 3.15: Time evolution of $\tilde{\mathbf{V}}$ for the first 75 OFDM sub-channels, in static conditions. The columns refer to the transmit antennas while the rows to the spatial streams.

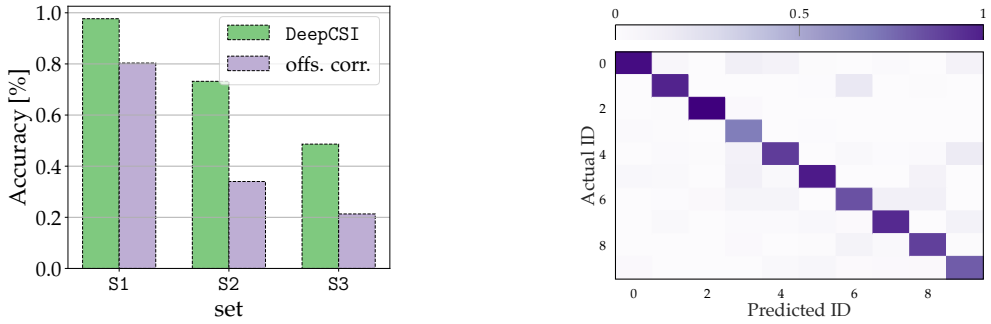


(a) `S1`. Accuracy: 97.46% **(b)** `S2`. Accuracy: 10.10% **(c)** `S3`. Accuracy: 9.60%

Figure 3.16: Confusion matrices, beamformee 1, 3 TX antennas, spatial stream 1.

3.5.5 `DeepCSI` performance compared with learning from a processed input

`DeepCSI` learns beamformer-specific features directly from the I/Q samples of matrix $\tilde{\mathbf{V}}$. As an alternative approach, we evaluated the effect of pre-processing such I/Q data before using it as input for the DNN. Specifically, we applied to the beamforming feedback matrices the data cleaning algorithm presented in Section 2.4.1. As detailed in Section 2.3.3, the CFR estimated at the beamformee on the NDP – and from which $\tilde{\mathbf{V}}$ is derived – slightly deviates from the theoretical model in Eq. (3.1) due to hardware imperfections causing undesired phase offsets. Besides the PDD, all the other contributions to Eq. (2.12) are associated with imperfections at the transmitter device, that in our case is the target of our fingerprinting technique, i.e., the AP. Our key intuition is that also the beamforming feedback matrix $\tilde{\mathbf{V}}$ – derived from $\mathbf{H}$ as discussed in Section 3.3.2 – would be affected by the phase offsets (i)-(v). Thus, we may use the offsets cleaning algorithm of 2.4.1 to improve its quality.



(a) `DeepCSI` accuracy compared with the one obtained using the processed input.

(b) Confusion matrix for `S1` after offset correction. Accuracy: 80.34%

Figure 3.17: Comparison with the accuracy obtained by learning the fingerprints from the processed version of $\tilde{\mathbf{V}}$, i.e., after applying the offsets correction (offs. corr.) in 2.4.1. Beamformee 1, 3 TX antennas, spatial stream 0.

Along this line of reasoning, we evaluate in Figure 3.17 the impact of a preliminary offset cleaning phase on matrix $\tilde{\mathbf{V}}$ on the fingerprinting accuracy. The proposed `DeepCSI` (with no offsets

cleaning) outperforms its version with the described offset correction capability across all the training/testing sets. In other words, the offsets introduced by the beamformer hardware imperfections are strategic to reliably recognize the device, and any offset cleaning may result in their partial removal, affecting the fingerprinting quality.

3.6 Concluding remarks

In this chapter, we have proposed `DeepCSI`, a novel approach to Wi-Fi radio fingerprinting which leverages standard-compliant steering matrices to authenticate MU-MIMO Wi-Fi devices. Conversely from prior work, `DeepCSI` does not require SDR technologies. We have extensively evaluated the performance of `DeepCSI` through a massive data collection campaign with off-the-shelf equipment, where 10 Wi-Fi radios emit MU-MIMO signals in different positions. Experimental results indicate that `DeepCSI` is able to correctly identify the transmitter with an accuracy above 97%. Besides, we have evaluated `DeepCSI` fingerprinting accuracy by differentiating the set of positions for the devices at training and testing times. Our technique achieves an accuracy of 46% when tested on positions that sharply differ from those in the training set. On the other hand, the accuracy increases to 73% when training is performed on a more balanced set of spatial points, which allow the neural network classifier to interpolate the training patterns for the missing points, using those from adjacent training positions. Overall, we believe that `DeepCSI` represents a significant step forward in the way radio fingerprinting is performed in Wi-Fi networks. Indeed, as `DeepCSI` runs on off-the-shelf devices, without relying on dedicated implementations on software defined radios, makes it ready to be used in commercial Wi-Fi systems.

Part II

Sensing-assisted communication for cellular networks

4

Users' mobility prediction for mobile networks

4.1 Introduction

We now move on to the second aspect of our discussion on the interplay between wireless communications and sensing, being *sensing-assisted communication* applications. As already introduced, we focus on the computation and exploitation of long-term mobility predictions, i.e., ranging from one to a few seconds into the future. We target an urban scenario where users are vehicles moving within the city while being constantly connected to the network providing both communication and computation services. Specifically, in this chapter we detail the design and implementation of a novel learning based algorithm that tracks the movements of a user (vehicle) and predict its next point of attachment after the handover from the current serving cell. Next, the devised algorithm is used within the resource allocation strategies detailed in Chapter 5 and Chapter 6.

As discussed in Section 1.2.1, fifth generation (5G) mobile networks are expected to provide broadband access in dense areas, high communication capacity and ultra-low latency, for moving users, thanks to the introduction of, among others, millimeter wave (mmWave) communications and massive multi-input, multi-output (MIMO) technologies. Besides other benefits, the high bandwidth available at mmWave frequencies and the antenna arrays employed for massive-MIMO, will allow 5G networks to provide *network-based positioning* that, with an expected accuracy of less than 1 meter, will outperform that provided by the global navigation satellite system (GNSS) (5 meters) [108]–[110]. Network based positioning was considered as an optional feature in previous generation cellular systems and was implemented mainly for emergency calls, through different techniques [111]. Among them, the ones that provide the highest accuracy are based on trilateration and triangulation. The former relies on time of arrival (ToA), time difference

of arrival (TDoA) or received signal strength (RSS) metrics, while the latter exploits the angle of arrival (AoA) of the received signals. Both of them entail the combination of the measurements from three different evolved nodes base (eNBs). In 5G networks, the use of two dimensional antenna arrays with the introduction of massive MIMO will enable two dimensional terminal positioning using a single AoA measurement [112], without performing triangulation.

In this context, our idea is that 5G eNBs could independently track the connected users to estimate their physical trajectories as they move through the radio cells. In a vehicular network, the mobility is constrained to the physical connectivity structure of road links, as well as their morphology. Therefore, our intuition is that vehicular mobility is a highly correlated process, which can be effectively captured by the eNBs by observing the radio signals from their connected mobile users. In the literature, this prediction task has been tackled exploiting Markov chains (MCs) and support vector machines (SVMs), using as input the sequence of *previously visited cells* [113]–[115]. Although this approach may be effective in simple scenarios, the historical information about the sequence of serving eNBs does not suffice to provide accurate position estimates as the number of physical trajectories increases, and this is especially true in urban environments served by densely deployed networks. In these cases, more accurate models would be desirable. As a solution, we propose to exploit the location information that will be provided by 5G networks. Our approach is *decentralized*, as each eNB tracks its own connected users by estimating the probability distribution of the next serving eNB (taken from the set of neighboring eNBs). Learning is carried out through recurrent and convolutional neural networks (NNs) combined with standard MCs. Once trained, the predictor returns mobility estimates, i.e., a probability vector for the next serving eNB, from the moment a user connects to the cell to the moment it leaves it, and by continuously improving such estimates as subsequent radio samples from this user are collected.

To the best of our knowledge, the only other work adopting a similar approach is [116], where the authors estimate the next eNB that the user will be connected to using RSS values acquired every 500 ms, and a recurrent neural network (RNN) as a mobility predictor. The work in [116] is preliminary as a simple RNN design is considered, training it in a *centralized* fashion on a small-scale scenario featuring 8 eNBs and 3 users moving in a grid topology over an area of 6×6 square kilometers. Moreover, the output of such predictor is a hard decision variable corresponding to the identifier of the next serving eNB. The approach presented in this chapter provides significant advancements over this prior research along several dimensions: *i)* a real-life mobility scenario is considered, using simulation of urban mobility (SUMO) [29] to emulate more than 50 thousands vehicles moving within the city of Cologne; *ii)* a realistic 5G network is emulated, with densely deployed eNBs within the monitored area (coverage radius of 200 m, for a total of 77 eNBs); *iii)* the approach is decentralized by design, i.e., a single NN-MC based predictor is trained for each eNB, solving issues such as the lack of data for less represented trajectories, and *iv)* our tool returns soft-predictions in the form of a probability vector over the next serving eNB. Numerical results indicate that the new serving eNB can be estimated with an accuracy higher than 94% from four seconds before the handoff, which seems to be appropriate for proactive resource management algorithms.

The rest of the chapter is organized as follows. In Section 4.2 we detail the proposed mobility

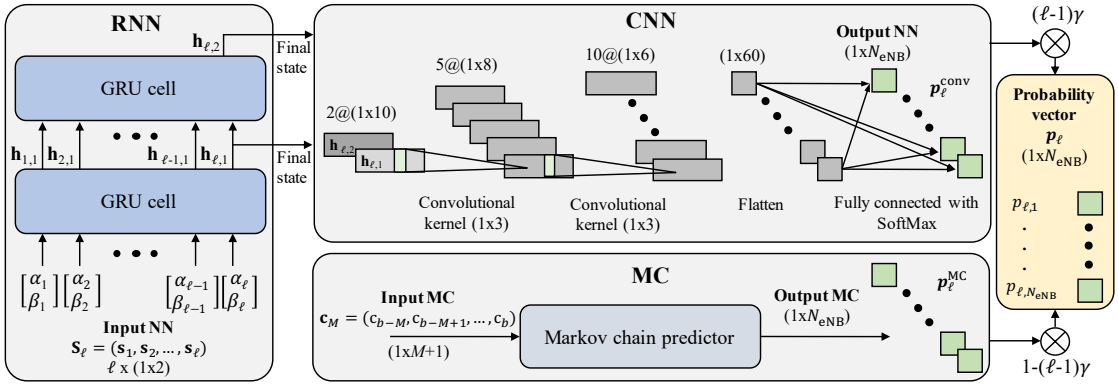


Figure 4.1: Processing pipeline. The classification is performed in parallel by the NN block (gated recurrent unit (GRU)-based RNN + CNN), using as input the (α, β) angles describing the trajectories, and the MC block, exploiting the sequence of previous visited cells. The output vector of probabilities for each adjacent eNB ($\boldsymbol{p}_\ell$) is obtained as a combination of the two classification outputs.

prediction framework. In Section 4.3 we describe the experimental scenario considered for the assessment of the prediction performance in Section 4.4. Concluding remarks are in Section 4.5.

4.2 Sequential learning architecture

Next, we detail the proposed mobility estimation framework. The location of a vehicle is collected through the *azimuth* (α) and *elevation* (β) angles that describe its relative position with respect to the serving eNB. We refer to a *trajectory* as the sequence of azimuth and elevation angles that describes the positions of the vehicle from the time it connects to a eNB ($t = 0$) to the instant in which it hands over to another eNB ($t = LT_s$), where L is the number of samples of the complete trajectory, collected with a sampling period of T_s seconds. Formally, an L-long trajectory is denoted by $\mathbf{S}_L \triangleq (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_L)$, with $\mathbf{s}_\ell = (\alpha_\ell, \beta_\ell)^T$. Our framework returns a new estimate every time a new trajectory sample is gathered, with an accuracy increasing with the number of samples.

Considering that the eNB to which the user will hand over must be one among the N_{eNB} adjacent eNBs, we formulate the problem as a classification task and we design a supervised learning framework to carry it out. The classification is carried out in parallel by two blocks: *i)* a NN block, consisting of a RNN (see Section 4.2.1) and a convolutional neural network (CNN) (see Section 4.2.2), that uses as input the user trajectory samples, and *ii)* a final MC block (as proposed by [114]) that exploits the sequence of previously visited cells (see Section 4.2.3). The classification results from *i)* and *ii)* are then combined to obtain the final estimates using a parameter that is learned as part of the training (see Section 4.2.4). The blocks are shown in Figure 4.1, while the training process is detailed in Section 4.2.6. To complete our framework, we propose a pre-processing algorithm to deal with possible imbalanced datasets (see Section 4.2.5), which is key to also learn those trajectories for which there are fewer training examples.

The benefits of adopting an AoA and learning-based mobility predictor are two-fold. First,

this approach allows preserving the user's privacy, as we estimate the positions of the vehicles on the eNBs without requiring them to send information such as position, speed, and direction. This also avoids the additional network traffic that would be caused by regular exchanges of such information between the vehicles and the eNBs. The second reason is that learning is key to automatically capture complex road topologies and mobility behaviors, which would hardly be represented via, e.g., first order models entailing straight motion and constant velocity.

4.2.1 GRU-based RNN for mobility feature extraction

RNN is the most used learning tool when dealing with sequential data, as RNNs are able to track temporal correlation among input samples and allow handling inputs with different numbers of samples [117]. An RNN is composed of one or more *memory cells* whose number is indicated by the number of layers, L_{rec}. The temporal correlation between subsequent input samples, from $\mathbf{s}_1$ to $\mathbf{s}_\ell$, is recorded in the values of the neurons constituting the cell *internal state*, $\mathbf{h}_\ell$, where ℓ is the sample index. To that end, the input sequence $\mathbf{S}_\ell \triangleq (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_\ell)$, is fed one sample at a time into the first layer of the RNN. In case of multiple layers, the internal state of the first layer cell becomes the input of the second layer cell, and so on. Every time a new sample arrives, the internal states of the RNN cells are updated based on the values of the previous states and on the new sample. The update procedure is described by the *cell update function*, that defines the memory cell, whose parameters are adjusted to minimize a *loss function*, i.e., a user-defined distance between the expected and the actual network output. Dropout layers can be inserted between recurrent layers, providing network regularization by randomly zeroing some of the elements of the input, following a Bernoulli distribution. In our implementation, we use $L_{\text{rec}} = 2$ GRUs [118] memory cells with 20 neurons per layer, and a dropout layer with a retain probability of 0.9. For each input sequence, we collect the internal state values of the two stacked RNN cells, each time a new trajectory sample is processed. Note that the RNN internal state is a *feature vector* capturing the most representative traits of the input time series. After applying layer normalization, this feature vector is used by the subsequent CNN network that acts as a classifier.

4.2.2 CNN for next serving eNB prediction

A CNN consists of the cascade of layers (i.e., functions), where all or some of them use the *convolution operator*, i.e., small-size kernels (matrices of weights), which are convolved with the entire input data volume [117]. The result of such convolution is passed through a non-linearity, called *activation function*, to produce an output *activation map*. The number of kernel filters employed at one layer defines the *depth* of the layer output, i.e., the number of activation maps produced for a single input, each of which captures different aspects of the input. One interesting feature of CNNs is their *parameter sharing*: the kernel filters are applied to each position of the input, allowing one to capture specific input peculiarities, no matter their specific location. As for RNNs, dropout layers can be interposed between convolutional layers. We use a one dimensional CNN, where the kernels are vectors, with $L_{\text{conv}} = 2$ convolutional layers with `ReLu` activation functions, and two dropout layers with a retain probability of 0.9. Batch normalization is used after each

convolutional layer to reduce the training time. The CNN input corresponds to the two (stacked) final internal states extracted from the GRU-based RNN. The output of the last convolutional layer is then passed through a fully connected layer with `SoftMax` activation function that produces an N_{eNB}-dimensional vector, $\boldsymbol{p}_\ell^{\text{conv}} \triangleq (p_{\ell,1}^{\text{conv}}, p_{\ell,2}^{\text{conv}}, \ldots, p_{\ell,N_{\text{eNB}}}^{\text{conv}})$, containing the probability that the user will hand over to each of the N_{eNB} neighboring cells after leaving the current one.

4.2.3 MC for next serving eNB prediction

MC is here used to capture the correlation in the sequence of cells that are visited by a user. Indicating with $c_{b+1,i}$ the i-th neighbouring eNB to the current (serving) eNB, $i \in \{1, \ldots, N_{\text{eNB}}\}$, and $\mathbf{c}_M = (c_{b-M}, c_{b-M+1}, \ldots, c_b)$ the sequence of $M+1$ cells visited up to time $t = \ell T_s$, each element of the N_{eNB}-dimensional vector $\boldsymbol{p}_\ell^{\text{MC}} \triangleq (p_{\ell,1}^{\text{MC}}, p_{\ell,2}^{\text{MC}}, \ldots, p_{\ell,N_{\text{eNB}}}^{\text{MC}})$ is computed as follows:

$$p_{\ell,i}^{\text{MC}} = \frac{\sum_{\text{dataset}} \texttt{Number}(c_{b+1,i}, \mathbf{c}_M)}{\sum_{\text{dataset}} \texttt{Number}(\mathbf{c}_M)}, \tag{4.1}$$

where `Number` indicates the times the sequence inside the parenthesis appears in the training dataset. The value of M, i.e., the MC order, used within the algorithm, results from a performance assessment reported in Sec. 4.4.

4.2.4 Combination of CNN and MC predictors

The final N_{eNB}-dimensional probability vector $\boldsymbol{p}_\ell \triangleq (p_{\ell,1}, p_{\ell,2}, \ldots, p_{\ell,N_{\text{eNB}}})$ is obtained by combining the NN and MC estimations using a parameter $\gamma \in \{0, 1\}$ as follows:

$$k = \min\{(\ell - 1)\gamma, 1\}, \tag{4.2}$$

$$\boldsymbol{p}_\ell = k\boldsymbol{p}_\ell^{\text{conv}} + (1 - k)\boldsymbol{p}_\ell^{\text{MC}}. \tag{4.3}$$

This combination strategy allows the network to weight differently the NN and MC estimations as the number of gathered trajectory samples ℓ changes. Note that γ is fixed for each NN-MC predictor and is automatically learned during training. Eq. (4.2) ensures that the combination parameter k remains inside the range $[0, 1]$.

4.2.5 Dealing with imbalanced datasets

In an urban scenario, the number of examples available for each of the possible trajectories inside a eNB's coverage area usually differs. Using such an imbalanced dataset for training would lead to a neural network with poor mobility prediction capabilities for those users that follow under represented trajectories. To cope with this, we propose a method to artificially generate new trajectories, so that they resemble the ones belonging to the under represented classes. This oversampling process is only performed on the training dataset, whose trajectories are labeled with

the indication of the next eNB towards which the user will be moving after the handover. Based on this label, the input data is split into different *groups*. The groups with cardinality less than 70% that of the most populated one undergo an oversampling process, while the others remain unchanged. The trajectories belonging to each of these groups are further split into a number of *classes* reflecting a quantization on the trajectory space, for the trajectories that lead to the same final eNB (same group). To do this, we first obtain fixed-length trajectory representations (*codes*), as standard clustering algorithms do not deal with variable length inputs. A simple and common coding method in machine learning is *padding*, i.e., to concatenate each input vector with as many zeros as needed to reach a common (fixed) length. However, in our case, this is not the best choice as the inputs have a specific physical meaning (city roads). Instead, we interpolate the trajectories with a degree 3 polynomial function, and use the interpolated versions, i.e., the codes, as inputs for the (unsupervised) hierarchical density-based spatial clustering of applications with noise (HDBSCAN) algorithm [119]. For each trajectory class, we then produce artificial trajectories picking at random trajectories from the class, and adding white Gaussian noise with standard deviation $\sigma = 5 \times 10^{-4}$ rad to each sample. The new trajectories are assigned to the proper label to be used in the training process. The oversampling procedure is summarized in Algorithm 4.1.

Algorithm 4.1 Oversampling procedure.

Require: trajectories inside a cell - training set
- group the trajectories based on their labels (next eNB);
- $h_c \leftarrow$ cardinality of most populated trajectory group;

for all groups with cardinality smaller than $0.7 \times h_c$
- encode the trajectories into fixed-length representations (20 samples) using degree 3 polynomial interpolation;
- apply HDBSCAN to the fixed length trajectories (codes), to further split them into classes;
- `new_samples_per_class` $\leftarrow 0.7 \times \frac{h_c}{\texttt{number\_of\_classes}}$;

for all classes
- generate new trajectories by adding $w \sim \mathcal{N}(0, \sigma^2)$ to `new_samples_per_class` trajectories selected at random from those inside the class;

end for

end for

4.2.6 Training process

Next, we detail how the building blocks composing our solution are combined and jointly trained, as summarized in Algorithm 4.2. First, the training trajectories are used to generate artificial examples to balance the training dataset. Hence, from each (L-long) trajectory we extract all the possible $(L-1)$ sub-trajectories of size greater than two, i.e., ($[\mathrm{s}_1, \mathrm{s}_2]$, $[\mathrm{s}_1, \mathrm{s}_2, \mathrm{s}_3]$, ...$[\mathrm{s}_1, \boldsymbol{s}_2, \ldots \mathrm{s}_L]$). Each of these sub-trajectories is forwarded through the NN block to obtain the probability vector $\boldsymbol{p}_\ell^{\mathrm{conv}}$ for the next eNB. The sequence of previous visited cells $\mathbf{c}_M$ is used to compute the probability vector $\boldsymbol{p}_\ell^{\mathrm{MC}}$ that is then combined with $\boldsymbol{p}_\ell^{\mathrm{conv}}$ to obtain $\boldsymbol{p}_\ell$. $\boldsymbol{p}_\ell$ is then compared with the ground truth to determine the prediction loss. (The ground truth is a N_{eNB}-dimensional "one hot vector"

with the element corresponding to the actual next eNB being equal to 1, and the other ones to 0). The loss is computed by multiplying the negative log-likelihood by an exponential penalty term that depends on the sub-trajectory length (ℓ) and on the total length (L) of the trajectory as follows:

$$\text{penalty}(\ell) = 2^{\ell/L} - 1. \tag{4.4}$$

The backpropagation of sub-trajectory prediction losses allows the network to adjust its parameters to correctly estimate the next eNB even when the complete L-long trajectory is not yet available. Through Eq. (4.4), higher penalties are assigned as the number of collected samples, ℓ, approaches L. This forces the network to become more accurate as the number of samples increases. Every 20 epochs, the performance of the framework is assessed using the validation data. The RNN+CNN weights and the combination parameter (γ) are extracted and saved if they outperform the previously saved ones in terms of validation accuracy. Otherwise, an ε-greedy approach is adopted, see, e.g., [120]: with probability $\varepsilon = 0.3$ the training continues from the weights computed in the last iteration, while with probability $1 - \varepsilon$ it restarts from the previously saved network parameters. Besides, every 20 epochs the artificial training examples used to balance the dataset are re-computed. These combined actions allow increasing the generalization capacity of the framework as during each training round (20 epochs) the dataset is re-populated maintaining all the real training data, while adding *different* artificial examples.

Algorithm 4.2 Deep learning framework: training process.

```
- num_epochs ← 100, ε ← 0.3, max_val_accuracy ← 0;
for j ← 1 to (num_epochs/20)
  - add training trajectory examples – Section 4.2.5;
  - extract training sub-trajectories;
  for jj ← 1 to 20
    - split the set of sub-trajectories in batches with batch_size = 64 examples each;
    for all batches
      - forward propagate the input – Section 4.2.6;
      - compute the error using penalty in Eq. (4.4);
      - backpropagate the error through the network;
    end for
  end for
  - assess prediction performance with validation set;
  - val_accuracy ← accuracy on the validation set;
  if val_accuracy > max_val_accuracy
    - save network parameters (weights, biases, γ);
    - max_val_accuracy ← val_accuracy;
  else if random(0, 1) > ε
    - restore previously saved network parameters;
  end if
end for
```

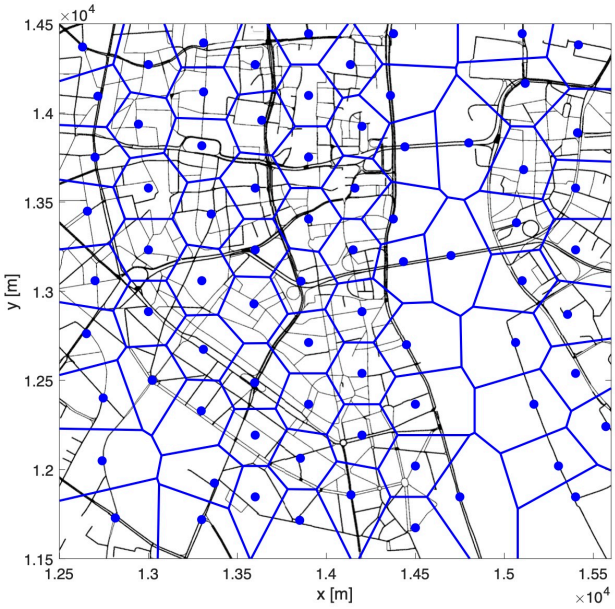


Figure 4.2: Deployment of the eNBs over the selected area of the city of Cologne. The eNBs are the centroids of the Voronoi cells. The x and y coordinates are the same used in the TAPAS Cologne simulation scenario.

4.3 Experimental mobility setup

We consider an urban 5G scenario in the center of the city of Cologne ($3,000 \times 3,000$ square meters), with 5G enabled vehicles that move around while being continuously connected to the 5G network. The mobility is emulated with SUMO, an open-source traffic simulation suite that allows generating the movement of emulated users around a predefined city road map, and extracting the needed metrics [29]. Specifically, we use the "TAPAS Cologne" simulation scenario, which mimics the vehicular traffic within the city for a whole day on the basis of travelling habits of the city dwellers [121]. The vehicle density changes during the day presenting two peaks, in the time intervals between 8-9 a.m. and 4-6 p.m., reflecting the mobility due to getting to and leaving work. The eNBs are deployed according to the road topology: eNBs are placed at the street crosses or along the roads, ensuring an inter-distance among nodes of around 400 m. The mobility area is covered with convex polygonal cells, each with a 5G eNB in its center, see Figure 4.2. In detail, the considered city region is partitioned into Voronoi cells, using the positions of the eNBs as generating points (the centroids), and considering the Euclidean distance to shape the cell edges (the faces of the Voronoi cells). The average number of vehicles within each eNB coverage ranges from 10 to 50 depending on the position of the eNB and on the hour of the day, with velocity ranging between 0 and 45 km/h. The proposed simulation scenario is in line with field trials, e.g., [122].

Twenty-four-hour long SUMO mobility traces with $T_s = 0.25$ s granularity were collected for each of the 77 eNBs in the deployment: 70% of such data was used to train and validate the mobility prediction algorithm, while the remaining 30% served to assess the performance of the

proposed VM replication strategy. For such assessment, we considered vehicles $v \in \mathcal{A}(t)$ approaching the edge of the serving eNB coverage area, i.e., that are about to hand over to a new eNB. For the following results, on average this occurs when a user is less than $\delta = 40$ meters apart from the radio cell's edge.

4.3.1 Channel model for user-eNB association

For the wireless link between the vehicles and the eNB, we use the mmWave channel model of [123], describing urban non line of sight (NLoS) scenarios. At every point in time, each vehicle connects with the eNB providing the best communication conditions, measured in terms of path loss (PL),

$$\mathrm{PL}(d) = \rho_1 + \rho_2 10 \log_{10}(d) + \eta \ \ [\mathrm{dB}], \tag{4.5}$$

$$\eta \sim \mathcal{N}(0, \sigma^2) \ \ [\mathrm{dB}], \tag{4.6}$$

where d is the distance between the transmitter and the receiver and $\mathcal{N}(0, \sigma^2)$ represents a Gaussian r.v. with zero mean and variance σ^2. Considering $d = 200$ m, the channel parameters are set to $\rho_1 = 46.61$, $\rho_2 = 3.63$ and $\sigma = 9.83$ dB, see [123]. For the handover user association, a hysteresis mechanism, with parameter ε_P, is considered (see, e.g., [124]). Specifically, a handover is performed to a new eNB only if the power received from this new eNB exceeds that received from the serving one by an amount greater than $\varepsilon_P = 2$ dB.

4.3.2 Vehicle location information acquisition

We emulate the single eNB AoA positioning approach – that will be integrated in 5G systems – by processing the x-y traces extracted from the SUMO simulator to obtain the angular information. White Gaussian noise with zero mean and σ_w^2 variance was added to the extracted data to account for estimation errors. Specifically, indicating with h_{eNB} the height of the eNB, the azimuth (α) and elevation (β) components of the AoA are obtained as follows, see Figure 4.3,

$$\Delta x = x_{\mathrm{user}} - x_{\mathrm{eNB}} + w, \ \ w \sim \mathcal{N}(0, \sigma_w^2), \tag{4.7}$$

$$\Delta y = y_{\mathrm{user}} - y_{\mathrm{eNB}} + w, \ \ w \sim \mathcal{N}(0, \sigma_w^2), \tag{4.8}$$

$$\alpha = \begin{cases} \arctan\left(\frac{\Delta y}{\Delta x}\right), & \text{if } \Delta x > 0 \\ \arctan\left(\frac{\Delta y}{\Delta x}\right) + \pi, & \text{otherwise} \end{cases}, \tag{4.9}$$

$$\beta = \arctan\left(\frac{\sqrt{\Delta x^2 + \Delta y^2}}{h_{\mathrm{eNB}}}\right). \tag{4.10}$$

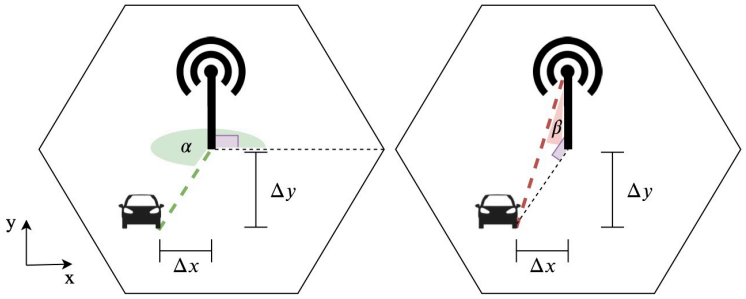


Figure 4.3: Azimuth (α) and elevation (β) angles that identify position of each vehicle inside the (serving) cell. The relative coordinates of the vehicle with respect to the eNB are indicated with Δx and Δy. 90-degree angles are drawn in violet.

We set $h_{\text{eNB}} = 8$ m and $\sigma_w = 5/3$ m in our simulations, accounting for possible positioning errors as large as five meters ($\pm 3\sigma_w$).

4.4 Prediction performance

We independently trained the proposed mobility prediction approach for each of the 77 eNBs in the deployment. The twenty-four hour long mobility traces were labeled according to the next serving eNB identifier and split as follows: 60% for training, 10% for validation, and 30% for the final test. Note that, although the architecture is the same for all the eNBs, the NN weights and the combination parameter γ are independently adjusted during the training processes, and are in turn eNB specific.

The best order for the MC predictor is assessed by evaluating the next cell's prediction accuracy obtained using MC only, for different values of M. As can be seen from Figure 4.4 the best choice is $M = 2$, as using higher MC orders does not lead to any noticeable increase in the accuracy. Note that if the number of previously visited eNBs, m, at time t is smaller than the MC order M, the prediction is performed by additionally training an MC with order $m < M$.

In Figure 4.5, we show the hard-decision accuracy obtained by averaging the results achieved by the 77 NN+MC classifiers, grouped based on the number of eNBs in their handover set. For comparison, we also report the accuracy achieved by the MC classifier. We do not report the cases when the handover set contains fewer than four eNBs because they rarely occurred in our simulations, and hence we could not collect enough data for a reliable training and assessment in such cases. As expected, the NN+MC estimation technique outperforms the state of the art MC classifiers, proving the effectiveness of collecting users' positions inside the serving eNB to estimate the next serving eNB, rather than only relying on the sequence of previously visited eNBs. In all the cases, the average accuracy was found to be greater than 94%, starting from 4 s before the handover and remaining above 91% until 10 s before it.

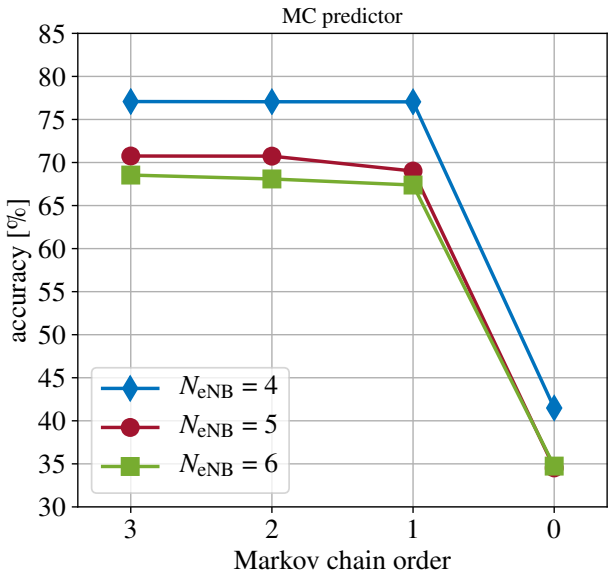


Figure 4.4: Next cell estimation accuracy using different order MCs.

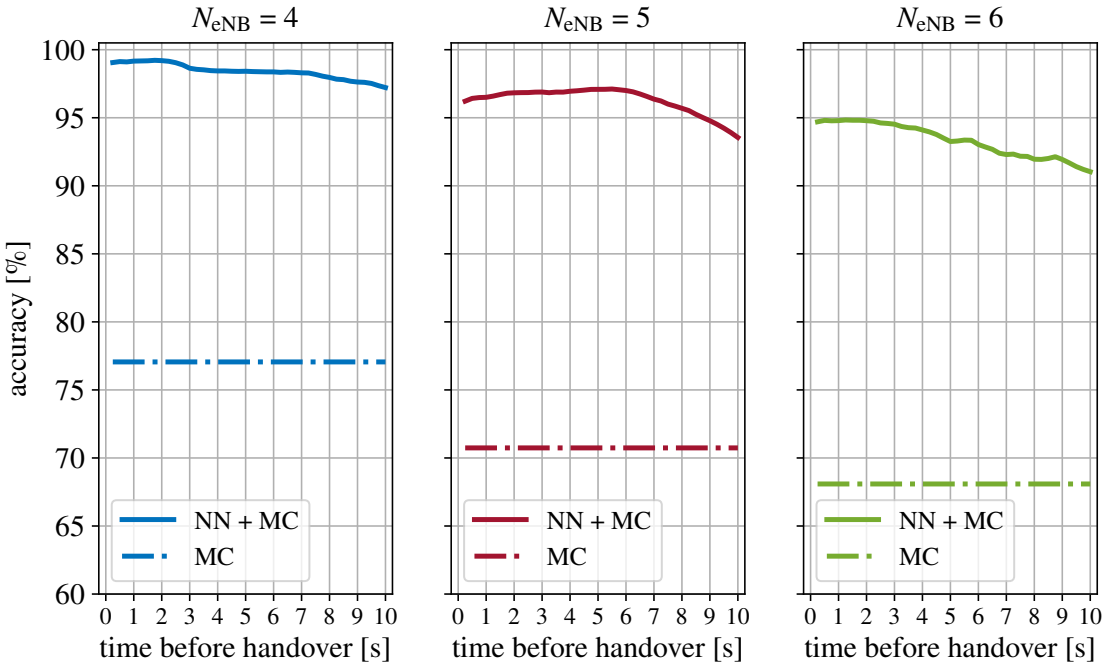


Figure 4.5: Prediction accuracy: the x-axis shows the time before the handover occurs, normalized with respect to the average permanence time inside the serving cell.

4.5 Concluding remarks

In this chapter, we have proposed a decentralized technique for the estimation of mobility trajectories in 5G vehicular networks. Our objective was to predict, with high accuracy, the next serving eNB a few seconds before the handover actually happens. Such estimates are computed by each eNB by processing the radio signals from its connected users. A prediction framework combining neural networks and Markov chains is put forward, detailing its design and proposing a novel training approach that effectively copes with under represented trajectories. Mobility trajectories are collected through the SUMO mobility simulator for the city of Cologne, considering a deployment with 77 eNBs. The final results confirm the effectiveness of the proposed approach, showing accuracies greater than 91% for predictions performed 10 seconds before the handover. Our technique can be utilized in conjunction with resource allocation algorithms, e.g., to proactively allocate communication and computing tasks prior to the actual handover events. Proactive strategies are expected to enhance the quality of service experienced by the users, while improving the way in which network resources are assigned. In the next two chapters of the thesis we provide two practical application examples in the context of optimal resource allocation for multi-access edge computing (MEC) enabled networks.

5

Mobility-aware migration of computing services in MEC enabled networks

5.1 Introduction

In this chapter we leverage the long-term vehicles' mobility prediction algorithm presented in Chapter 4 to devise a novel strategy for network computing resources management in an Internet of vehicles (IoV) scenario, providing an example of the potential of integrating environmental information into the network management processes. Specifically, we are concerned with location and mobility aware computing services which will play a key role in the IoV [22], [125]. Vehicles are becoming connected entities, and with the advent of online gaming, on demand streaming and assisted driving services, are expected to turn into data hubs with abundant computing needs. Pioneered by the Google car, the IoV is becoming the new fabric where vehicles will interact with one another, while communicating with the network infrastructure and with the edge/cloud, where services are provided. With 5G, connectivity, computing and caching will converge into the so called multi-user MEC, which will offer computing power right at the network edge. In this chapter, we tackle the problem of ensuring *continuity of computing tasks* as mobile users (vehicles) move within a city-wide 5G network, and we do so by migrating their services as the users change their point of attachment to the network. Past work dealt with this by exploiting *reactive* approaches [51], [125]–[128], where the virtual machines (VMs) hosting the services are migrated *after* the users have moved to the new radio cell. Here, we advocate the use of mobility predictions to *proactively* allocate computing resources, by replicating the VM on an optimal number of neighboring locations prior to the execution of the handover. If carefully orchestrated and implemented, the proposed proactive allocation strategies lead to relevant energy savings, by keeping (computation) service discontinuity to a minimum. The mobility estimates are ob-

tained through the algorithm presented in Chapter 4 of this thesis. Next, we devise an algorithm that select the best VM migration strategy for each vehicle at each handover event, proactively reallocating its MEC resources to ensure MEC *service continuity*, while concurrently *minimizing the energy consumption* at network level. The proposed VM migration strategy operates independently at each eNB/MEC host (MEH). The mobility predictions are used within a decentralized and online Lyapunov-based optimization technique to decide *where* and *how many* computation resources (VMs) are to be (proactively) migrated for each vehicle. The optimal Lyapunov-based VM migration strategy is obtained in closed form, leading to a low complexity and online technique. The algorithm outputs the number of VM replicas (*how many*) and the eNBs (MEHs) *where* these VMs are to be migrated prior to the handover execution.

The proposed solution is numerically assessed in the real-life vehicular mobility scenario introduced in Chapter 4, using SUMO [29] to emulate thousands of vehicles moving within the city of Cologne, and simulating a realistic 5G network with densely deployed eNBs (and co-located MEHs). Our numerical results reveal that the proposed mobility-aware VM migration strategy achieves a low energy expenditure, i.e., close to the lower bound where each VM is replicated to a single next MEH, by granting a service discontinuity risk close to that of the *full replication* strategy, i.e., where VMs are replicated to all the available next MEHs. We also assess the impact of the predictor's accuracy on the VM migration decisions, finding a large performance improvement of NN+MC with respect to prior techniques.

The rest of this chapter is organized as follows. The related work is reviewed in Section 5.2. The system model is detailed in Section 5.3. In Section 5.4, the optimization problem's formulation is presented along with the online closed form solution, whereas the numerical results are presented in Section 5.5. Concluding remarks are drawn in Section 5.6.

5.2 Related work

In wireless mobile networks, mobile users that change their serving eNB experience service connectivity continuity thanks to handover procedures. The same principles apply to MEC services, where not only the connectivity, but also the MEC offloading service process is migrated from the current serving MEH to the MEH in the next connectivity domain. There are two main techniques to handle MEC offloading service continuity in mobile networks during handovers: (i) The VM that hosts the ongoing computation process is kept on the MEH where the MEC service was initiated, and where resources were allocated in the first place [129]. In this case, standard communication handover procedures ensure that the user equipment (UE) remains connected to the serving MEH while handing off from one eNB/MEH to a new one [130]. Such approach was proven to be inefficient in the case of frequent handovers due to the latency of and the energy drained at the backhaul links. (ii) A second option consists of *migrating* the VM to a new MEH in proximity of the UE, leading to a reduction in the communication delay and in the UE uplink transmission power. In this chapter, we focus on this second option, although the combination of the two approaches is also possible as we discuss in Chapter 6.

In the literature, several migration mechanisms have been proposed to determine *when* and

where to migrate VMs. The majority of the approaches rely on the *reactive* migration of VMs, i.e., applied as UEs exit the coverage area of the eNB/MEH where the VM is currently hosted [131]–[133]. Although this can be effective in handling delay tolerant services, it is inapt to manage delay-sensitive tasks, for which *proactive* approaches represent a better solution [126]. Proactive VM migration decisions can be made based on the UE-MEH distance as the sole metric [51], on mobility predictions [127] and on joint mobility prediction and MEH availability estimates [125]. The currently proposed proactive solutions for VM migration rely on probabilistic mobility models and, in the case of poor mobility estimates, the VM migration process can experience a major performance degradation. To take this into account, the authors of [128] propose to proactively replicate the VM into different neighboring MEHs, restoring service in a new MEH as the user moves from the current eNB to the next one. However, the problem formulation in [128] entails a computationally expensive integer linear programming optimization task that is difficult to deploy in practice. Moreover, the mobility prediction model relies on historical data about user's movements between the radio cells, disregarding the online evolution of the user's position *inside* the serving cell. In this chapter, we propose to jointly (i) improve the reliability of mobility predictions, by also including in the estimates the user's movement inside the radio cells (see Chapter 4), and, (ii) dynamically adapt the VM replication strategy (its overhead, intended as the number of VM copies) based on mobility prediction estimates. This makes it possible to dynamically evaluate the risk of service discontinuity given an identified set of target MEHs, by concurrently minimizing the cost incurred in the VM migration. In addition, we tackle the complexity issue via a lightweight optimization framework that is solved in closed form, entailing low complexity calculations. We advocate that accurate and online mobility models are key to the design of effective MEC migration strategies during handovers. In fact, as we quantify in the numerical results of Section 5.5, better mobility estimates lead to the replication of fewer VM instances for the same reliability target, with a subsequent reduction in the amount of resources (memory, computation and energy) that are employed to support the MEC services during handovers.

5.3 System model

We consider a vehicular network consisting of a set $\mathcal{N}$ of eNBs, each co-located with a MEH, and a set $\mathcal{V}$ of vehicles which move following the morphological road constraints. Vehicles send offload requests for intensive computation services to the MEHs. The computation services can experiences a large number of handovers due to vehicle mobility and to the radio coverage range of eNBs.

To handle the computation offloading requests and guarantee service continuity, each MEH includes: (i) several VMs to execute the users' services, (ii) a MEC platform, (iii) a mobility prediction unit, (iv) a *VM migration control unit* (including admission control), and (v) a *virtualization infrastructure*. Each vehicle is associated with a specific VM in the serving MEH, which handles the required services and that can be migrated to new MEHs tracking the vehicle's movements. For that, as a vehicle approaches the edges of a cell, the radio network interface service (RNIS) running on the MEC platform informs entities (iii) and (iv) that the vehicle is about to perform the

handover to a new radio cell. At this point, the unit running the mobility prediction algorithm starts tracking the vehicle's position to estimate the next point of attachment. The information is then used by the VM migration control unit to decide whether to migrate the user's VM to a new location, and in that case, in which MEHs the replication is to be performed. The VM migration control unit has an overview of the complete mobile edge system and serves as the orchestrator. Finally, the virtualization infrastructure provides computation, storage, and network resources to instantiate VMs, execute offloaded services, and manage their migration process by booking memory and computing capacities at MEHs identified by the VM migration control unit. The MEH architecture detailed above is compliant with [41], [134].

While our system model can be considered for both live and static VM migration, in our investigation we focus on live VM migration and *stateful* application processes. VM migration is ensured between MEHs through backhaul links whose data rate is assumed to be sufficiently high so that the time required for the VM migration can be neglected. This assumption can be relaxed, but in the following of this chapter, we restrict the analysis to this case.

Our system model is time-dependent and operates in an online fashion. Time is slotted, with time slots of fixed duration τ. At each time slot, a cost function representing the *energy expenditure* is minimized subject to a risk constraint, seeking a good balance between energy minimization and service continuity.

5.4 Problem formulation

Hereafter, we present the proposed online optimization algorithm to dynamically control the number of VM replication instances that are migrated prior to a handover event and also the MEHs destinations of such migrations. Specifically, we address the issue of VM migration due to *off-board* service computation requested by vehicles that are exiting the serving eNBs (radio) coverage area and that are at a distance smaller than δ meters from the edge of the cells, i.e., vehicles that are about to perform a handover to a new eNB. The investigated problem is formulated for the system model described in Section 5.3 under the assumption that the considered computing processes cannot be processed on-board on the vehicle, i.e., they must be offloaded to a MEH in the network for their execution. This assumption is especially realistic for those applications that require data that is not stored in the vehicle or software not installed on it. The extension of the present analysis to the local computation of tasks is left for future work.

The set of vehicles that are about to leave the cell, i.e., that are a distance smaller than δ meters from each edge, is here referred to as $\mathcal{A}(t)$. We identify as $v \in \mathcal{A}(t)$ (any) one of such vehicles. When the VM migration control unit decides to migrate the VM handling services for vehicle v, $M_v(t)$ parallel migrations are performed, replicating the VM that is serving vehicle v towards the $M_v(t)$ distinct MEHs selected among the available next eNBs. This will allow restoring the service computation in a new MEH in case the handover is performed. Hence, $M_v(t)$ is an integer indicating the number of selected MEH sites where the VM serving vehicle v will be migrated to in time slot t. Note that, for any vehicle at time t, there are N_{MEH} available eNBs towards which it can potentially hand over to and, each of them has an associated MEH. N_{MEH} does not depend

on the position of the user inside the cell: it is fixed and specific for each eNB, as it reflects the underlying mobility process, i.e., the road topology and the way eNBs are deployed. So, at time t there are N_{MEH} MEH candidates for VM migration and $M_v(t) \leq N_{\text{MEH}}$ of these are selected. Specifically, $M_v(t)$ is chosen from the set $\mathcal{N}_{\text{MEH}} = \{1, \ldots, N_{\text{MEH}}\}$ that contains the possible numbers of VM replicas to be performed, $M_v(t) \in \mathcal{N}_{\text{MEH}}$. Most importantly, using the mobility prediction framework of Chapter 4, each vehicle in the considered area (i.e., less than δ meters apart from the cell's edge), obtains a mobility-prediction vector $\boldsymbol{p}_v = (p_{v,1}, \ldots, p_{v,N_{\text{MEH}}})$ of size N_{MEH}, with $\sum_i p_{v,i} = 1$. The entries in the probability vector $\boldsymbol{p}_v$ are arranged in decreasing order, namely,

$$p_{v,1} > p_{v,2} > \cdots > p_{v,N_{\text{MEH}}}. \tag{5.1}$$

Vector $\boldsymbol{p}_v$ provides a probabilistic estimate of the next visited eNB site when leaving the current serving one, by ranking the N_{MEH} candidate eNBs according to their probability of being visited. $p_{v,i}$ represents the probability that the vehicle will hand over to the eNB ranked as the i-th most probable among the neighboring ones. This information is obtained by learning (at runtime) the underlying vehicular mobility dynamics via the combined NN+MC mobility prediction model of Section 4.2 (note that in this chapter the trajectory length index ℓ is omitted in the interest of readability). When we select the $M_v(t)$ MEHs for VM replication, we pick eNB indices according to the order of the corresponding probabilities $p_{v,i}$, i.e., we select the most probable ($p_{v,1}$) first and add further eNBs until $M_v(t)$ sites are picked, in the order expressed by Eq. (5.1). This means that once $\boldsymbol{p}_v$ is obtained, the number $M_v(t)$ of eNBs that we select is all we need to provide a complete description of the VM migration process. In fact, we may alternatively use a state vector $(b_{v,1}, \ldots, b_{v,N_{\text{MEH}}})$, where $b_{v,i} \in \{0, 1\}$ is a binary number indicating whether eNB i is selected or not. However, this vector can be fully replaced (and obtained) by the information contained in vector $\boldsymbol{p}_v$ (providing the ranking among eNBs) and $M_v(t)$ (the number of eNBs that are picked).

We define the binary function $o_v(t)$, indicating whether the VM migration control unit decides to migrate or not the VM serving vehicle v that is leaving the cell

$$o_v(t) = \begin{cases} 1, \text{ VM will be migrated} \\ 0, \text{ VM will not be migrated} \end{cases}. \tag{5.2}$$

If $o_v(t) = 1$, the VM serving vehicle v will undergo the replication process to guarantee service continuity as the vehicle moves out of the eNB coverage area. If instead $o_v(t) = 0$, the VM will not be migrated in any MEH and the handled service will be interrupted when the handover is performed. Being $\psi_v(t)$ the energy required to perform the VM migration towards a single eNB site, the energy consumed in the current MEH to replicate the VM to the $M_v(t)$ selected sites is

$$E_v(t) = M_v(t)\psi_v(t), \tag{5.3}$$

and it follows that the total energy consumed in time slot t for the VMs replications is

$$E(t) = \sum_{v \in \mathcal{A}(t)} o_v(t) E_v(t). \tag{5.4}$$

Let $N(t) = |\mathcal{A}(t)|$ be the number of vehicles that are about to leave the cell, i.e., the serving MEH, in time slot t. We define a control action vector for the VM replication process as follows,

$$\Omega(t) = \left(\Omega_1(t), \ldots, \Omega_{N(t)}(t)\right), \tag{5.5}$$

where $\Omega_v(t)$ is the control action taken in time slot t for vehicle $v \in \mathcal{A}(t)$,

$$\Omega_v(t) = (o_v(t),\, M_v(t)). \tag{5.6}$$

The objective function that we want to minimize is the long-term average energy expenditure associated with the migration of VMs,

$$\lim_{T \to +\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[E(t)\right]. \tag{5.7}$$

Note that only accounting for the energy cost, through Eq. (5.7), would lead to a solution where no VM replication is performed, as the energy is trivially minimized by setting $o_v(t) = 0$, for all v and t. Therefore, for a proper formulation, we need to account for an additional cost encoding the fact that computing tasks may be discontinued during handovers. To this end, we define a *risk metric* $\zeta_v(t)$ for each vehicle v, which encodes the total probability that the associated VM is not correctly migrated, i.e., that is migrated to MEHs which will not be visited after the handover. It follows that $0 \leq \zeta_v(t) \leq 1$ and for any vehicle v at time t, we define

$$\zeta_v(t) = 1 - o_v(t) \sum_{i=1}^{M_v(t)} p_{v,i}. \tag{5.8}$$

The rationale behind Eq. (5.8) is as follows. If the VM will not undergo the migration process, i.e., $o_v(t) = 0$, the risk is $\zeta_v(t) = 1$ as the vehicle v will suffer from service discontinuity. On the other hand, when the VM migration control unit decides to migrate the VM to all the available MEHs, i.e., to N_{MEH} eNBs, the risk is zero, as the vehicle will hand over with probability one to one of the available MEHs and the computing process will be resumed there. However, it may happen that the VM migration control unit establishes to replicate the VM in the first $M_v(t) < N_{\text{MEH}}$ MEHs (first $M_v(t)$ probabilities from vector $\boldsymbol{p}_v$), but the vehicle actually hands over to any one of the $N_{\text{MEH}} - M_v(t)$ remaining MEHs, where the VM was not replicated. This occurs with probability $\zeta_v(t)$ and corresponds to the case in which migration is performed to the wrong servers, i.e., after the handover, the computing task will not be resumed by the new serving MEH, as it was not migrated there. This risk shall be small and is here used as a constraint to counterbalance the minimization of the energy consumption in Eq. (5.7).

We define the average risk across all tasks in the current time slot as

$$\overline{\zeta}(t) = \frac{1}{N(t)} \sum_{v \in \mathcal{A}(t)} \zeta_v(t). \tag{5.9}$$

Informally, the proposed optimization framework is constrained on the fact that the average (long-term) risk, obtained by averaging Eq. (5.9) across multiple time slots, must be smaller than a pre-defined threshold $0 < \xi \leq 1$. ξ is a constant that we use to control the so-called *violation probability* for a processing task, i.e., the probability that the task computation will not be resumed after handing over to a new MEH.

To enforce this constraint, we define a virtual queue $Z(t)$ with the following update equation

$$Z(t+1) = \max\left\{Z(t) + \overline{\zeta}(t) - \xi, 0\right\}, \tag{5.10}$$

which is utilized to transform the long-term inequality constraint into a corresponding queue mean rate stability problem. Hence, we must ensure that the virtual queue $Z(t)$ is mean rate stable, which is mathematically expressed as [135, Chapter 2]

$$\lim_{t \to +\infty} \frac{\mathbb{E}[Z(t)]}{t} = 0. \tag{5.11}$$

The optimization problem in Eq. (5.7) is now changed to account for service priorities through an additional parameter $\chi_v(t)$ for each service class. For example, a service that is computationally (and energy) expensive will be discarded more often by the optimizer due to its high energy consumption. The priority parameter compensates for this: setting $\chi_v(t) = 0$ means not to assign any priority to the associated VM, while setting $0 < \chi_v(t) \leq 1$ allows controlling the migration process in case of VMs with different migration costs. It follows that Eq. (5.7) should be modified by considering the weighted energy expenditure

$$F(t) = \sum_{v \in \mathcal{A}(t)} o_v(t) E_v(t) (1 - \chi_v(t)), \tag{5.12}$$

instead of the pure energy expenditure $E(t)$, see Eq. (5.4).

The optimization problem for VM migration is then formulated as follows

$$\begin{aligned}
&\lim_{T \to +\infty} \min_{\substack{\Omega(t), \\ t \in \{0,\ldots,T-1\}}} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[F(t)] \\
&\quad \textbf{Subject to:} \\
&\quad (a)\ \lim_{T \to \infty} \frac{\mathbb{E}\left[Z(T)\right]}{T} = 0, \\
&\quad (b)\ o_v(t) \in \{0,1\}, \forall\, v, t, \\
&\quad (c)\ M_v(t) \in \mathcal{N}_{\text{MEH}}, \forall\, v, t,
\end{aligned} \tag{5.13}$$

which amounts to minimizing the overall weighted energy drained by the VM migration process (the objective function in the first line), while maintaining the average risk below constant ξ (queue stability constraint (a)).

5.4.1 Online optimization algorithm

To design a stable and online control algorithm, we use the *Lyapunov drift-plus-penalty* framework, see, e.g., [135],[136], [137]. This allows obtaining a dynamic decision making system, that controls the VM replication process in each time slot.

Given that $Z(t)$ represents the virtual queue backlog process, the *Lyapunov function* is defined as

$$L(Z(t)) = \frac{1}{2}Z(t)^2 \tag{5.14}$$

and the *one-step Lyapunov drift* is computed as

$$\Delta L(Z(t)) \stackrel{\text{def}}{=} L(Z(t+1)) - L(Z(t)). \tag{5.15}$$

The expression for the Lyapunov drift-plus-penalty is then obtained by combining the average weighted energy consumption in Eq. (5.12) and the average queue backlog in Eq. (5.15) through a parameter $V > 0$ that weighs the importance of the two quantities in the minimization problem, i.e.,

$$\Delta(t) = \Delta L(Z(t)) + VF(t). \tag{5.16}$$

Theorem 1 (Upper bound on the drift-plus-penalty)**.** *Eq.* (5.16) *can be upper bounded as*

$$\Delta(t) \leq \frac{1}{2} + Z(t)\left(\overline{\zeta}(t) - \xi\right) + VF(t). \tag{5.17}$$

Proof. Applying inequality $\max(x, 0)^2 \leq x^2, \forall x \in \mathbb{R}$, to the expression of the queue evolution in Eq. (5.10), we obtain

$$Z(t+1)^2 \leq Z(t)^2 + \left(\overline{\zeta}(t) - \xi\right)^2 + 2Z(t)\left(\overline{\zeta}(t) - \xi\right), \tag{5.18}$$

from which it follows that

$$Z(t+1)^2 - Z(t)^2 \leq \left(\overline{\zeta}(t) - \xi\right)^2 + 2Z(t)\left(\overline{\zeta}(t) - \xi\right). \tag{5.19}$$

Since, by construction, $0 \leq \overline{\zeta}(t) \leq 1$ (see Eq. (5.8) and Eq. (5.9)) and $0 < \xi \leq 1$ (by definition), we have $(\overline{\zeta}(t) - \xi)^2 \leq 1$ and

$$Z(t+1)^2 - Z(t)^2 \leq 1 + 2Z(t)\left(\overline{\zeta}(t) - \xi\right). \tag{5.20}$$

It follows that

$$\begin{aligned}\Delta(t) &= \Delta L(Z(t)) + VF(t) \\ &= \frac{1}{2}\left[Z(t+1)^2 - Z(t)^2\right] + VF(t) \\ &\leq \frac{1}{2} + Z(t)\left(\overline{\zeta}(t) - \xi\right) + VF(t).\end{aligned} \tag{5.21}$$

□

The *drift-plus-penalty* optimization algorithm observes $Z(t)$ in every time slot t and chooses an action $\Omega(t)$ that minimizes the right-hand-side of the inequality in Eq. (5.17). To do so, we use the concept of *opportunistically minimizing an expectation* [135, §1.8]. This is accomplished by greedily minimizing Eq. (5.16), neglecting the constant term 1/2. Hence, the online optimization problem is written as,

$$\begin{aligned}&\min_{\Omega(t)} \left[Z(t)\left(\overline{\zeta}(t) - \xi\right) + V \sum_{v \in \mathcal{A}(t)} o_v(t) M_v(t) \psi_v(t)(1 - \chi_v(t))\right] \\ &\textbf{Subject to:} \\ &(b)\ o_v(t) \in \{0,1\}, \forall\, v, \\ &(c)\ M_v(t) \in \mathcal{N}_{\text{MEH}}, \forall\, v.\end{aligned} \tag{5.22}$$

In the following proposition, we show how the optimization problem in Eq. (5.22) can be solved in closed form, leading to an efficient implementation of the solver.

Proposition 1 (Closed form solution of Eq. (5.22))**.** *Defining function* $g_v(M_v(t))$ *as*

$$g_v(M_v(t)) \stackrel{\text{def}}{=} VM_v(t)\psi_v(t)(1 - \chi_v(t)) - \frac{Z(t)}{N(t)} \sum_{i=1}^{M_v(t)} p_{v,i}, \tag{5.23}$$

the optimum control action at time t for vehicle $v \in \mathcal{A}(t)$ *is* $\Omega_v^*(t) = (o_v^*(t), M_v^*(t))$*, where*

$$M_v^*(t) = \underset{M_v(t) \in \mathcal{N}_{\text{MEH}}}{\operatorname{argmin}}\ g_v(M_v(t)), \tag{5.24}$$

and

$$o_v^*(t) = \begin{cases} 1, & \text{if } g_v(M_v^*(t)) < 0 \\ 0, & \text{otherwise} \end{cases}. \tag{5.25}$$

Proof. The minimization problem in Eq. (5.22) can be simplified by observing that term $Z(t)\xi$ at time t does not depend on the decision variables $\Omega(t)$. Henceforth, Eq. (5.22) can be equivalently rewritten as

$$\min_{\Omega(t)} \left[V \sum_{v \in \mathcal{A}(t)} o_v(t) M_v(t) \psi_v(t)(1 - \chi_v(t)) + Z(t)\overline{\zeta}(t)\right]. \tag{5.26}$$

Using the equations for the risk Eq. (5.8) and Eq. (5.9), we obtain

$$\min_{\Omega(t)} \left[V \sum_{v \in \mathcal{A}(t)} o_v(t) M_v(t) \psi_v(t)(1 - \chi_v(t)) - \frac{Z(t)}{N(t)} \sum_{v \in \mathcal{A}(t)} \left(o_v(t) \sum_{i=1}^{M_v(t)} p_{v,i} \right) \right], \tag{5.27}$$

where the terms not depending on the optimization variables $\Omega(t)$ are omitted. Rearranging the previous equation, we get

$$\min_{\Omega(t)} \sum_{v \in \mathcal{A}(t)} o_v(t) \left[V M_v(t) \psi_v(t)(1 - \chi_v(t)) - \frac{Z(t)}{N(t)} \sum_{i=1}^{M_v(t)} p_{v,i} \right]. \tag{5.28}$$

It follows that Eq. (5.28) naturally decomposes across variable v, so minimizing the sum corresponds to minimizing each term in isolation.
Due to this, the optimal control action $\Omega_v^*(t) = (o_v^*(t), M_v^*(t))$ is obtained from Eq. (5.28) for each $v \in \mathcal{A}(t)$ as follows.

First, we find the optimal number of VM replications, $M_v^*(t) \in \mathcal{N}_{\text{MEH}}$, to be performed if the VM migration control unit decides to migrate the VM serving vehicle v. Let $g_v(M_v(t))$ be

$$g_v(M_v(t)) \stackrel{\text{def}}{=} V M_v(t) \psi_v(t)(1 - \chi_v(t)) - \frac{Z(t)}{N(t)} \sum_{i=1}^{M_v(t)} p_{v,i}, \tag{5.29}$$

the optimum choice for $M_v(t)$ is

$$M_v^*(t) = \underset{M_v(t) \in \mathcal{N}_{\text{MEH}}}{\operatorname{argmin}} \; g_v(M_v(t)). \tag{5.30}$$

Now, Eq. (5.28) can be re-expressed as,

$$\sum_{v \in \mathcal{A}(t)} \min_{o_v(t)} o_v(t) g_v(M_v^*(t)), \tag{5.31}$$

given that $g_v(M_v^*(t)) \in \mathbb{R}$ and $o_v(t) \in \{0, 1\}$ the minimum in Eq. (5.31) is achieved setting

$$o_v^*(t) = \begin{cases} 1, & \text{if } g_v(M_v^*(t)) < 0 \\ 0, & \text{otherwise} \end{cases}. \tag{5.32}$$

□

5.4.2 Dealing with computation resource limitation in the neighboring MEHs

Exploiting the side information gathered from the network, the VM migration control unit at one eNB can avoid migrating the VM associated with vehicle v when no computation resources are left on its neighboring MEHs. This requires MEHs to periodically send their load status to

their neighbors. With this information, the probability vectors $\boldsymbol{p}_v$ of all vehicles at time t can be modified to consider possible resource limitations. Let C_i be the number of additional VMs that can be instantiated on the neighboring MEH i. This MEH ranks the vehicles in the current cell according to the probability of migrating towards it, from highest to lowest. The first C_i migration requests (vehicles) are accepted, whereas the remaining ones are not admitted by masking their probability vector, i.e., by assigning a zero to their probability of migrating towards MEH i, i.e., $p_{v,i} \leftarrow 0$ if vehicle v is not admitted. This is a form of *admission control* which is implemented prior to computing Eq. (5.8) and running the optimization, to avoid a later rejection due to insufficient computation resources at the MEH.

5.5 Numerical results

We consider the urban 5G scenario emulated with SUMO [29] detailed in Section 4.3. Additionally, each eNBs in the 5G network is endowed with MEH functionalities. In turn, the 5G enabled vehicles moving around the city use the 5G network to offload intensive computing tasks.

5.5.1 Energy expenditure model for VMs migration

We use the model in [138], where the authors exploit CloudSim, a widely used simulator for cloud computing infrastructures and services, to estimate the energy drained in a MEH to perform VM migration. Following their model, $\psi_v(t)$ is computed as a function of the VM memory usage $W_{[\text{MB}]}$ (in megabytes) as

$$\psi_v(t) = 3600 \times (0.512 W_{[\text{MB}]} + 20.165) \ \ [\text{J}], \tag{5.33}$$

where $W_{[\text{MB}]}$ depends on service type. In our numerical evaluation, we focus on three different service types (see [139] for their memory use): optimized calculus (OC), general purpose (GP) and hardware accelerated services requiring a GPU (GPU). The energy drained at the MEH to perform the VM migration is computed using Eq. (5.33) and is reported in Table 5.1.

service type	$\psi_v(t)$
OC	15.91 MJ
GP	79.89 MJ
GPU	221.73 MJ

parameter	value
τ	3 s
ξ	10^{-2}
V	8×10^{-12}

Table 5.1: Service types and simulation parameters considered in the numerical evaluation of VM migration strategies.

5.5.2 System parameters selection

The right part of Table 5.1 summarizes the numerical parameters. The duration of a time slot, τ, indicates the time between two consecutive executions of the optimization algorithm for the selection of the proper MEHs, towards which the VM is to be replicated. Note that, the user

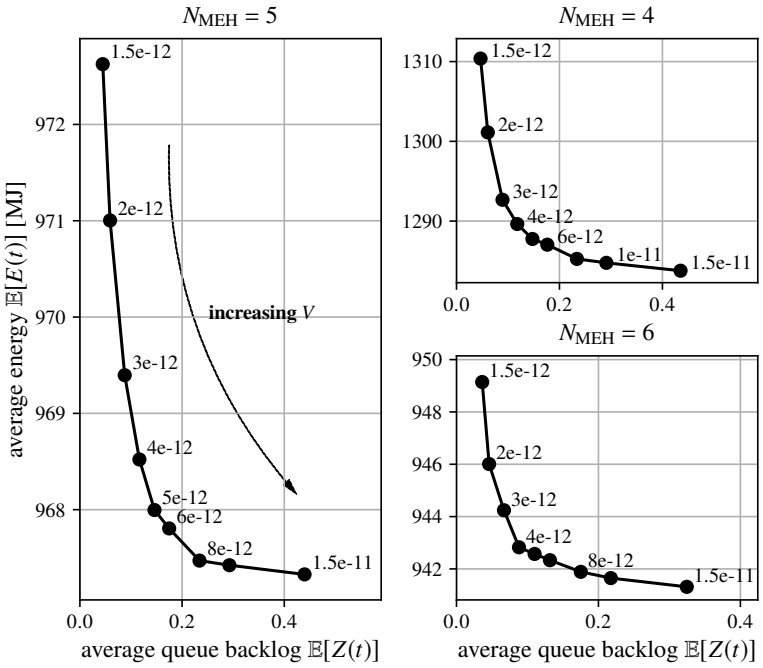


Figure 5.1: Energy consumption as a function of the queue backlog by varying V as a free parameter. Three plots are shown, by varying the number of candidate eNBs (MEHs) for the handover, $N_{\text{MEH}} \in \{4, 5, 6\}$.

before moving to a new cell is served by a local VM which runs inside the current (serving) cell, our optimization process only handles when and where the current VM is migrated. Hence, the value of τ does not affect the end-to-end latency of a URLLC bearer, but rather determines how many migration requests are collected before triggering the migration mechanism. The rationale is to proactively trigger the migration before a user moves to a new cell, so that the VM will be already instantiated as the user gets there. The parameter τ is selected considering the (typical) maximum speed of the vehicles $v_{\max}$ for the users approaching the cell's edge, and a distance parameter $\delta = 40$ m. Specifically, the inequality $\tau \leq \delta / v_{\max}$ is to be satisfied. This allows a vehicle requesting a service and entering the border region of a cell (distance smaller than δ with respect to its edge) at the beginning of a time slot τ not to suffer from service discontinuity during the handover. In our setup, being $v_{\max} = 45$ km/h, we fix $\tau = 3$ s.

The value of the risk parameter ξ is connected with the quality of service that we would like to guarantee to the users requiring computation offloading. We set $\xi = 10^{-2}$, i.e., we require the long-term average risk of our VM replication strategy not to exceed 10^{-2} (the VM should be migrated to the wrong set of eNBs once every 100 requests, on average).

In this first part of performance evaluation, we focus on a single service type, considering VMs for optimized calculus that require $\psi = 15.91$ MJ to be migrated, according to [138], [139]. The priority parameter $\chi_v(t)$ is set to zero for every vehicle v and time slot t (see Section 5.5.4 for the evaluation in case of multiple service types). We also consider that MEHs have enough

computing capacity to accept all the VM migrations from neighboring MEHs, i.e., no blockage can occur (resources limitation is considered in Section 5.5.5).

In Figure 5.1, we show the long-term average energy consumption over $1,000$ time slots and the average size of the virtual queue $Z(t)$ as a function of the parameter V. Each point in the graph corresponds to a different value of V, which is varied as a free parameter in the interval $V \in [1.5 \times 10^{-12}, 1.5 \times 10^{-11}]$. This evaluation is carried out across several eNBs, for different values of N_{MEH} (three examples, for $N_{\text{MEH}} = 4, 5, 6$, are shown in Figure 5.1). In general, $V = 8 \times 10^{-12}$ consistently represents a suitable choice across all the cases, as it provides a good trade-off between energy minimization and queue size. $V = 8 \times 10^{-12}$ may be referred to as the "optimal" point according to the elbow method [106].

The behavior of the queue backlog for $V = 8 \times 10^{-12}$ is plotted in Figure 5.2 for $N_{\text{MEH}} = 5$. As shown in this plot, after a transient phase, the queue $Z(t)$ stabilizes, as imposed by the mean rate stability constraint of Eq. (5.11).

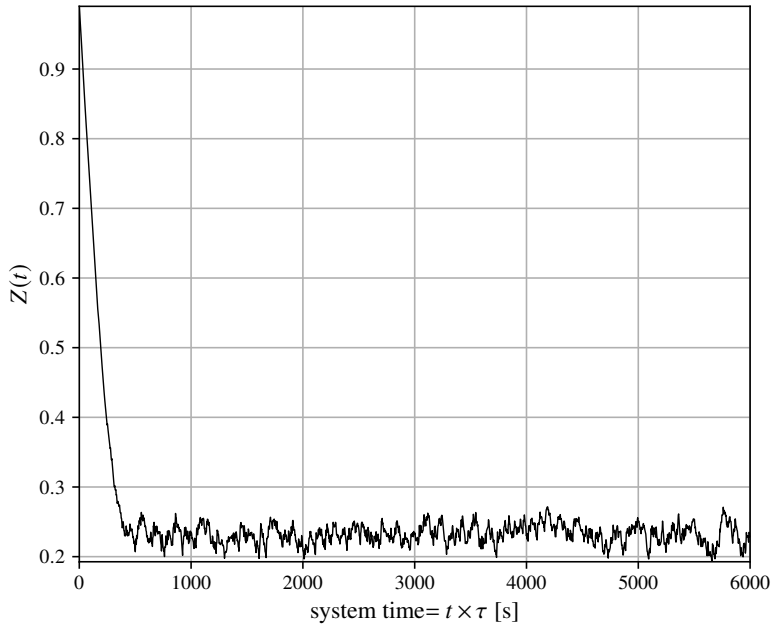


Figure 5.2: The evolution of the virtual queue $Z(t)$ depends on the number of times $\overline{\zeta}(t)$ exceeds $\xi = 10^{-2}$. Using our approach, Eq. (5.11) holds and the queue stabilizes for increasing t.

5.5.3 Performance analysis and comparison against prior art

To assess the performance of the proposed algorithm, referred to here as *optimal replication*, we compare it with the following commonly adopted strategies,

- *full replication*: VM serving vehicle v is migrated to all the N_{MEH} neighboring eNBs. This approach has been adopted in, e.g., [126].

- *single replication*: VM serving vehicle v is migrated only to the MEH that is the most likely to be visited after the handover, i.e., the MEH associated with $p_{v,1}$ in Eq. (5.1). This approach was considered in, e.g., [127].

Moreover, to assess the impact of the mobility prediction accuracy (i.e., the quality of probability estimates $\boldsymbol{p}_v$) on the energy and risk performance, we have run simulations using (i) our proposed NN+MC predictor and (ii) a predictor solely based on MCs (see Section 4.2.3), as presented in, e.g., [114].

In Figure 5.3, we show the time evolution of the total energy drained by the considered migration techniques for $N_{\text{MEH}} = 5$. For a clearer visualization of the differences among them, the cumulative energy is plotted rather than the instantaneous one.

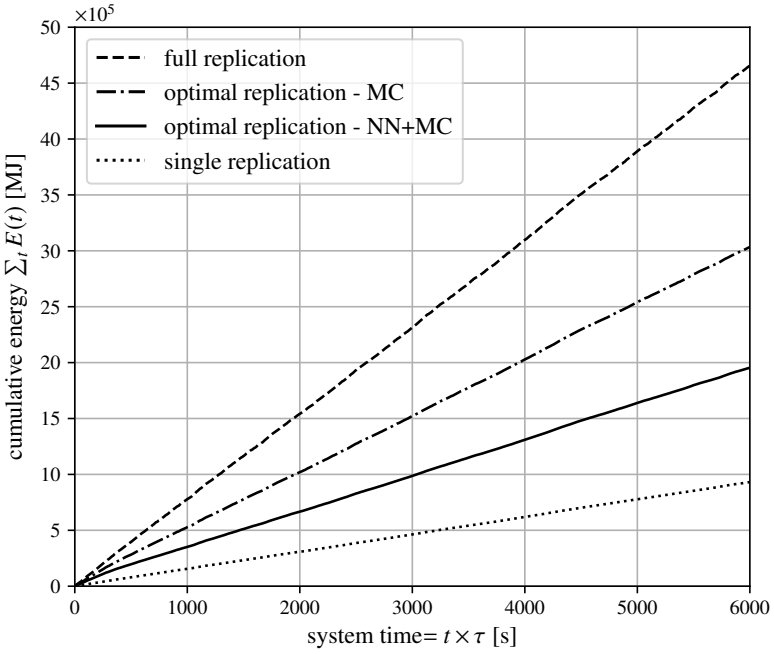


Figure 5.3: Total energy consumption *vs* time. The proposed approach considerably reduces the energy consumption as compared to the full replication scheme, by optimizing the number of VM replications for each submitted task.

As expected, the single replication strategy leads to the lowest energy consumption, as it replicates the VM to a single MEH at all times (for every vehicle). On the contrary, the full replication strategy leads to the highest, as it replicates the VM to *all* the available neighboring N_{MEH} sites. From this graph, we see that our approach (optimal replication) effectively reduces (by more than 50%) the total energy drained as compared to the full replication strategy. This is possible thanks to the careful selection of the number and destination of the replicated VM instances for each computation task. Moreover, we underline the impact of using the combined NN+MC mobility prediction algorithm against the simplest one based on MCs. Note that NN+MC provides more accurate probability estimates $\boldsymbol{p}_v$ which, in turn, entail the selection of fewer next eNBs for the

same risk target. This translates into lower energy consumption. As expected, for both full and single replication strategies, the energy expenditure is independent of the mobility estimation technique. In fact, with the former, mobility estimates are not used, whereas with the latter the energy expenditure amounts to copying the VM to a single next MEH, and the energy cost of this is the same across all the candidate sites.

The risk $\overline{\zeta}(t)$ is shown in Figure 5.4 for the NN+MC and MC mobility predictors, with $N_{\text{MEH}} = 5$. For a better visualization, the plot interval is restricted to the last 300 slots towards the end of the simulation. As expected, the risk is always zero in the full replication case. For the single replication strategy, it instead highly depends on the performance of the mobility predictor. For this scheme, a perfect prediction, e.g., carried out by a genie, would lead to a zero risk, and in general the better the prediction is, the lower the risk. From the numerical results, single replication achieves an average long-term risk of 0.463 when using the MC predictor, which is decreased to 0.057 using NN+MC, due to the highest accuracy of the latter. Nevertheless, the average risk of the single replication strategy is still at least five times higher than that of the proposed approach, for which the average risk stabilizes to $\xi = 10^{-2}$.

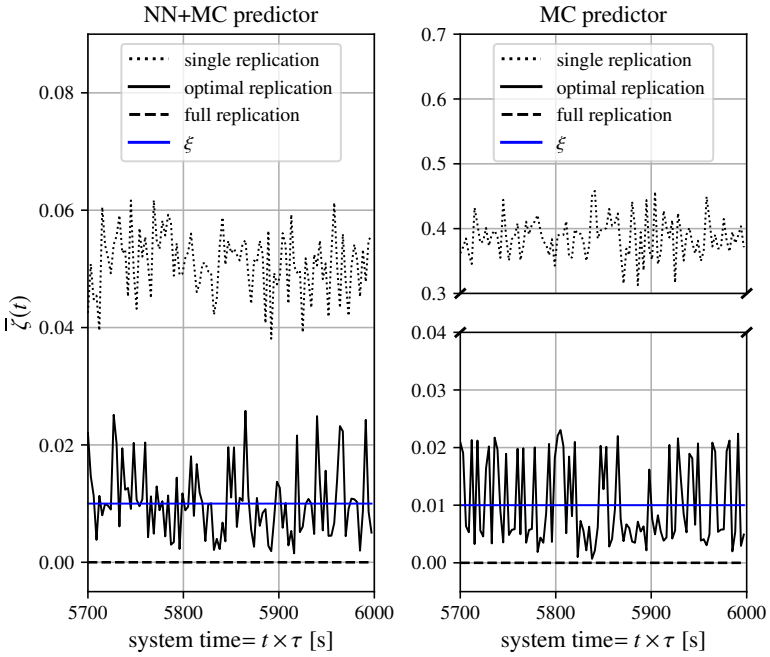


Figure 5.4: Temporal evolution of the average risk $\overline{\zeta}$. Our approach is constrained to meet the risk constraint ξ in an average sense (long-term).

In more detail, for the proposed technique, the risk is subjected to a constraint that must be met regardless of the accuracy of the predictor. For this reason, the average risk of the proposed optimal technique in the left and right plots of Figure 5.4 behave similarly, remaining both bounded around the same target risk $\xi = 10^{-2}$. This is a good property of the proposed algorithm, whose behavior is adjustable as a function of the user-defined average risk constraint ξ, which will be

always met by design. Also, as seen from Figure 5.3, although the risk performance is unaffected, a better mobility estimator translates into lower energy consumption, as thanks to it fewer replications are performed for the same risk target ξ.

As a last performance metric, we compute the successful migration probability: we say that the VM migration succeeds when the user hands over to an eNB (MEH) where its VM was proactively migrated into. In this case, the user resumes the computation on the new MEH without loss of continuity. In Figure 5.5, we compare the successful migration rate for the single and the proposed migration strategies. The success rate depends on the accuracy of the mobility predictor used to estimate the probability vector over the next serving eNBs. The performance gap between MC and NN+MC is evident for the single replication case. Instead, the proposed technique is less sensitive to the adopted mobility estimator, as it compensates for the less accurate prediction accuracy of MC by replicating the VM to more MEHs. As expected, full replication (not shown in the figure) always achieves a success rate of 100%, as it sends the VM to all the neighboring MEH sites.

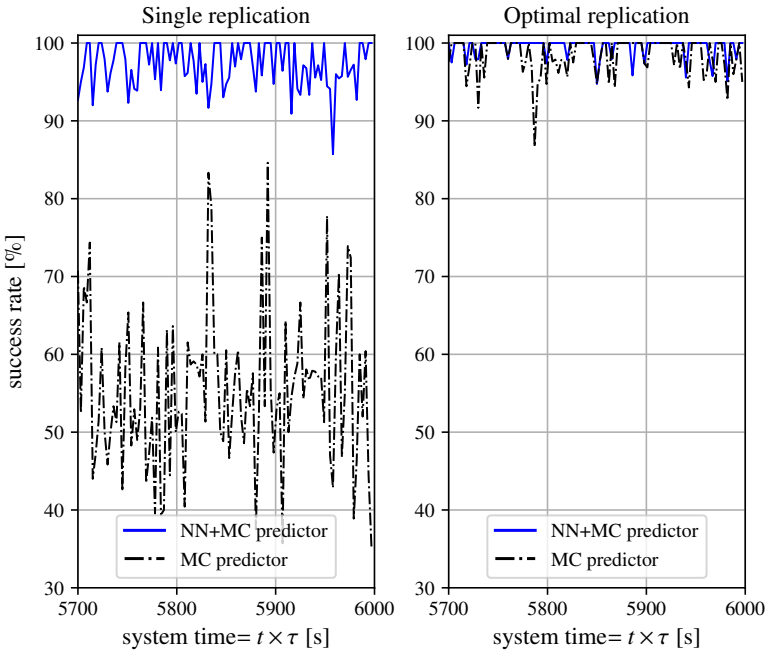


Figure 5.5: Success rate in VM migrations. Although the success rate also depends on the mobility predictor's accuracy, the proposed replication approach overtakes the problem by increasing the number of VM replicas, by always meeting the average risk constraint ξ. The metric is obtained only considering the offloaded tasks, i.e., the ones for which $o_v(t) = 1$.

5.5.4 Performance analysis for different service types

Here, we evaluate the performance of the proposed algorithm in case vehicles request different service types and, in turn, their VMs drain a different amount of energy for their migration. We

consider the three services introduced before, i.e., OC, GP and GPU, comparing two different scenarios. In the first one, we set $\chi_v(t) = 0,\ \forall v, t$. Table 5.2 ("same priority") details the average number of optimal (see Proposition 1) VMs migrations for this case, by grouping the vehicles requesting the same service. Figure 5.6, left plot, shows the evolution of the average risk for each service type. In this first scenario, the VMs with the lowest migration costs are served more often, whereas higher cost VMs are penalized. In the second scenario, we set $\chi_v(t) = 0.8$ and $\chi_v(t) = 0.4$ for the GPU and GP services respectively, while set the priority parameter for OC to zero. This allows counterbalancing the higher energy cost of GPU services, re-equalizing the risk, see the right plot in Figure 5.6 and Table 5.2 ("different priorities").

service type	same priority		different priorities	
	$\chi_v(t)$	$\overline{M_v}(t)$	$\chi_v(t)$	$\overline{M_v}(t)$
OC	0	3.2	0	2.7
GP	0	2.4	0.4	2.0
GPU	0	1.7	0.8	2.1

Table 5.2: Average number of replicas for VMs handling different services.

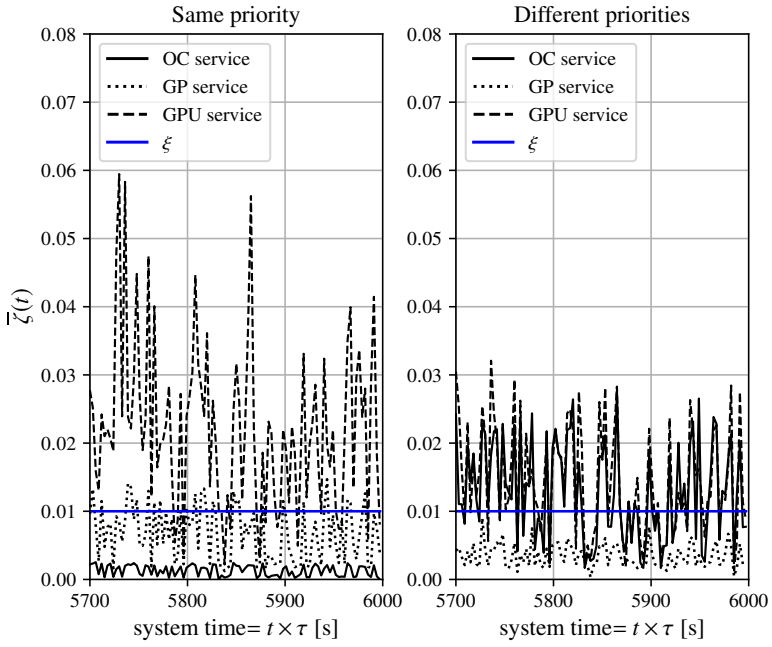


Figure 5.6: Temporal evolution of the risk ζ averaged for the different computing services.

5.5.5 Impact of limited computation resources at the MEHs

In the previous sections, we have considered that the computation capacity at the MEHs is always sufficient to accept all the incoming computation requests. Instead, in case the MEHs have limited computation capacity, e.g., due to a system overload or to their poor dimensioning, some VM migrations may be blocked. In Figure 5.7, we show the average risk as a function of the MEH capacity, i.e., the number of VMs that can be simultaneously hosted at a MEH. In this evaluation, the energy cost and the priority are respectively set to 15.91 MJ (OC) and zero for all the vehicles. Each curve in the plot refers to a different eNB in the neighboring set. For this plot, the MEH capacity was varied from 0 to 50 in steps of 5, the risk was averaged over $2,000$ time slots by grouping vehicles according to their next eNB. As the capacity approaches zero, the blockage probability and, in turn, the risk, both approach one. Instead, an increasing capacity leads to a decreasing risk, which eventually stabilizes around the target, which is $\xi = 0.01$. A proper dimensioning of the MEH capacity depends on the local amount of requests per unit time, which is in turn linked with the road topology and with the amount of vehicular traffic in the considered area.

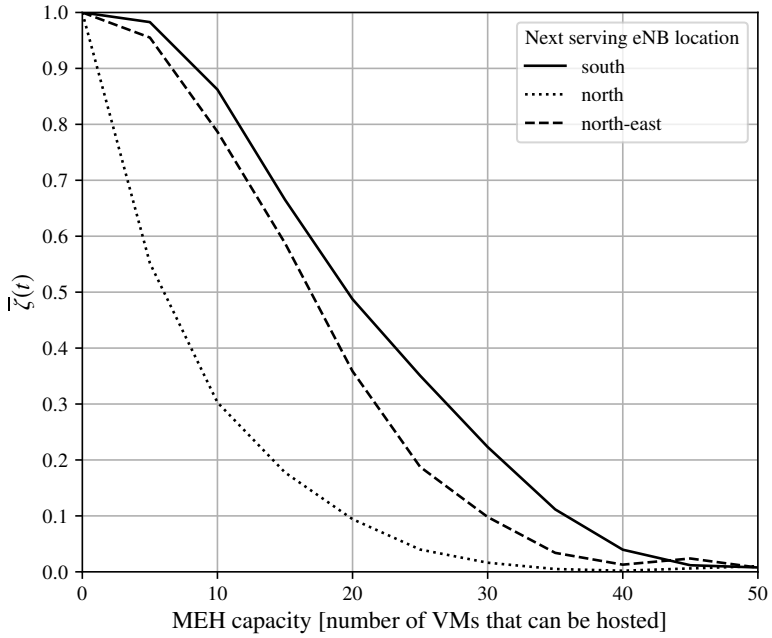


Figure 5.7: Evolution of the average risk as a function of the MEHs capacity. The vehicles are grouped based on the eNBs where they are going.

5.6 Concluding remarks

In this chapter, we have presented a new proactive technique for online migration of computation services (VMs) for vehicular 5G networks. Our approach combines a mobility prediction algorithm with an online optimization solver. The mobility estimates are expressed as a probability vector over the neighboring radio cells, i.e., the next vehicle's point of attachment. The mobility prediction algorithm, which combines neural networks and Markov chains, is independently trained and exploited at each radio cell and the output probability vectors are utilized as the input of a Lyapunov based online optimizer. The resulting approach, exploiting online decision making and mobility estimation, allows striking a good balance between the number of VMs that are replicated (representing an energy cost for the system) and the risk of losing service continuity. Numerical results demonstrate the effectiveness of the proposed solution in comparison with two benchmark (proactive) migration strategies from the literature. Our technique effectively reduces, by more than 50%, the total energy drained as compared to replicating the VM to all the neighboring radio cells, by keeping the risk of losing continuity of computation services as vehicles hand over across radio cells at a (user-defined) low value.

6

Mobility- and energy-aware computing job scheduling in MEC enabled networks

6.1 Introduction

In this chapter we design EASE (energy-aware job scheduling at the edge), a novel policy to allocate computing jobs within the network facilities, considering the same IoV scenario of the previous chapters. In Chapter 5 we proposed to proactively *replicate* the virtual entity handling the computing service on neighboring MEHs thus ensuring service continuity as the user (vehicle) moves within the network area. Here, we move a step forward by devising an online strategy for job allocation that includes the possibility for the service to keep executing on the MEH where it started, considering backhaul links to route the job results toward the new user's location. The decision on whether to move the entity executing the service on a MEH closer to the user or to complete the computation where it started is based on energy, memory and computing power considerations. As done in Chapter 5, we leverage the mobility predictor of Chapter 4 to estimate the best sites where the users' jobs can be allocated, meeting both the network's and the users' requirements. The proposed strategy is based on *single migrations* of the virtual entities – instead of their replication – as we consider eNBs equipped with energy harvesting devices and, in turn, we are concerned with the efficient utilization of the available green energy while tolerating a small risk on the service interruption. As quantified in Figure 5.5, by leveraging the NN+MC mobility prediction algorithm in Chapter 4 we obtain very accurate estimates that make the interruption of the services unlikely to happen.

To the best of our knowledge, this is the first attempt to designing a complete framework for the *energy efficient* scheduling of computing jobs over MEHs networks, by exploiting mobility aware procedures. The devised system provides job schedules for time slots of fixed duration,

minimizing the carbon footprint at the network side – for the computation and communication services – subject to job latency and mobility constraints. The proposed job scheduling policy consists of two phases, the former is independently and locally executed at the eNBs (MEHs), while the latter is implemented as a decentralized consensus process. In the first phase, each MEH leverages estimates of the renewable (cost-free) energy, the computational power and the memory available within a prediction window to decide upon the optimal local amount of workload to be executed, subject to users' mobility and delay constraints. Each MEH also identifies the jobs that should be migrated to neighboring MEHs as associated with vehicles that are approaching the border of their current serving cell. The mobility predictor developed in [140] is used to determine the desired workload to transfer to each neighboring MEH. Then, in the second phase, the MEHs reach an agreement on the amount of workload to exchange to reduce the overall energy expenditure, while guaranteeing adequate quality of service (QoS) to the end-users. Here, a close-to-optimal integer solution for jobs migration is derived through a consensus algorithm followed by a rounding step, which use mobility predictions to make job migration decisions.

The proposed solution is evaluated in the real-world scenario emulated through the SUMO software and presented in Chapter 4. Numerical results reveal that the developed allocation strategy significantly reduces the carbon footprint of the edge network, with an increasing gain over heuristic strategies when the available green energy is scarce. At the same time, it properly allocates workload to the processing units according to their specific computing power, by delivering better QoS to the users with respect to heuristic solutions and meeting delay constraints. When possible, service migrations also follow the UE during handovers.

The related work is analyzed in the next Section 6.2, whereas the solution workflow is presented in Section 6.3, where we also detail the remaining sections of the chapter.

6.2 Related work

The resource allocation problem in a MEC scenario is extensively addressed in the literature. Among the most recent works, in [141] the authors present a job scheduler for containers management at the MEHs, aiming at reducing the network carbon footprint and its energy consumption. In [142], [143], the task offloading is optimized from a user perspective, minimizing the task completion time and the related energy expenditure. However, these approaches consider *static* users and, in turn, are not suitable for IoV scenarios, where the users – vehicles – move within the network area. In this chapter, we consider that jobs are generated by vehicles in an urban area, devising `EASE`, a scheduling algorithm to guarantee service continuity. The mobility management is a key aspect towards the effective implementation of MEC assisted IoV networks [144]. Computation service handovers entail not only the exchange of control messages, but also the migration of the data associated with the specific job under execution. The users' requests are served at a so called serving MEH through the instantiation of a virtual entity – either a VM or a container – empowered with adequate memory and computing resources to satisfy the service requirements [145]. Therefore, when a computing service handover is triggered, the virtual entity must be transferred to the target MEH to allow restoring the computation from the point where

the current serving MEH stopped. This poses several issues associated with the job latency constraints and the network energy migration costs. A paper addressing the latency challenge, and proposing strategies to reduce the migration time is [146]. The main focus is on *how* to migrate the virtual entity, by defining protocols to transfer the container/VM from the current location to the target one. Machen et al. [147] propose a layered framework to migrate applications encapsulated either in VMs or containers, showing a reduction in the service downtime. The authors of [148] leverage the layered nature of the storage system to reduce the overhead in the container file system synchronization between the serving and the target MEHs. We stress that, the just mentioned approaches are *reactive*, i.e., the service migration is performed *after* the user has moved to the new MEH site. This results in an unavoidable processing delay due to the time required for the virtual entity re-instantiation at the new MEH. EASE is instead a *proactive* approach, as the virtual entity is migrated *before* the handover event occurs, thus reducing the service interruption time.

Proactive methods require the MEC orchestrator to predict the user's next point of attachment to trigger the migration process in advance. Some recent works in the literature show the effectiveness of such a strategy but (i) they fail to provide a complete framework to optimally allocate the computing jobs within the network entities while considering energy, memory and computing power constraints, and (ii) they rely on a centralized orchestrator that computes the best policy to adopt knowing the state of all the network entities. Among them, in [132], the authors exploit the vehicle velocity and its direction to decide if and where, i.e., in which target MEH, the virtual entity should be migrated to ensure service continuity. Campolo et al. [149] exploit the pre-planned vehicle's routes to proactively migrate the MEH container following the user's movements. In [131], the authors design an optimal decision policy to decide whether to migrate the virtual entity to the target MEH, or to keep the job execution on the first serving MEH. However, they consider a simple mobility model, using Markov chain based predictions, which is deemed inapt to prove the effectiveness of their solution in a real-world setup. A different approach is presented in [128], where the user's virtual entity is replicated to multiple neighboring MEHs before the handover event occurs. The authors suggest using mobility estimates to place the replicas, but leave this for future study. The last point is addressed in [140], where the authors integrate accurate mobility predictions into a VM replication strategy, towards reducing the network energy consumption. Rago et al. [150] use predictions on user locations and task requests to optimally and proactively allocate jobs on the available MEHs. In [151], the authors use mobility estimates to allocate users' jobs through a recursive procedure based on genetic algorithms with the sole objective of minimizing the latency (energy consumption is not addressed). In [152], user mobility predictions are instead used to allocate sufficient resources to each users' job, to ensure the completion of the process within the eNB coverage region. In [153], the MEC service migration process and the physical route for the user to get to the destination are jointly optimized. The problem is solved through a multi-agent deep reinforcement learning approach to meet the job delay requirements with minimum migration cost and travel time. In **perin2021towards**, the authors study the problem of managing the energy coming from renewable sources to minimize the electricity drained from the power grid. However, unlike what we do with EASE in the present

work, they disregard the mobility aspect. Moreover, they distribute the whole model predictive control (MPC) solution, resulting in a high computational burden, while EASE performs a preliminary local phase, to reduce the complexity.

The work in this chapter sharply depart from the existing literature by proposing EASE, an energy- and mobility-aware, distributed and proactive scheduling framework for computing jobs allocation and virtual entity migration, with the objective of minimizing the carbon footprint of the MEH network. This is achieved combining local policies with a decentralized consensus algorithm, thus obviating the need for an orchestrator.

6.3 Solution workflow

The optimization objective of EASE amounts to minimizing the energy drained from the power grid – considering as cost free the energy that each MEH collects from renewable energy resources – while meeting memory, processing constraints and accounting for the user mobility process. A high level diagram of EASE is presented in Figure 6.1. The scheduler operates according to two optimization phases: 1) local phase (left of the diagram): a predictive control phase, performed locally at each MEH node, and 2) distributed phase (right): a collaborative optimization based on distributed consensus (solved via message passing). In phase 1, the MEHs locally control the ongoing computations, estimating the local processing capacity and energy availability within a given prediction horizon. At the same time, the local algorithm assesses the amount of workload that should be migrated ("desired workload migration estimation") to the neighboring MEH nodes, predicts insufficient local resources ("MEH resource estimation"), and accounts for mobility estimates ("next MEH prediction"), i.e., the vehicle that generated the job request is about to hand over to a neighboring radio cell.

With phase 2, taking the desired workload to be migrated from phase 1 as input ("neighbors' resource information"), the MEHs collectively reach an agreement ("workload migration agreement") about *how many* and *which* jobs are to be actually migrated, as well as about the target MEH for their migration ("job selection and migration").

After step 2, each node updates its local state equations with the new jobs generated by the vehicles under coverage and those received from the neighbors, and goes back to step 1.

The remainder of this chapter is organized as follows. The system model is presented in the next Section 6.4. A first problem formulation for the optimal scheduling is detailed in Section 6.5. The final scheduling solution, composed of the two phases (local and distributed) is presented in Section 6.6. The performance assessment is reported in Section 6.7 and final remarks are provided in Section 6.8.

6.4 System model

Time t is discrete and evolves according to slots of fixed duration τ, i.e., $t = 0, \tau, 2\tau, \ldots$. The considered setup consists of an urban environment covered by a set $\mathcal{N}$ of eNBs, each co-located with

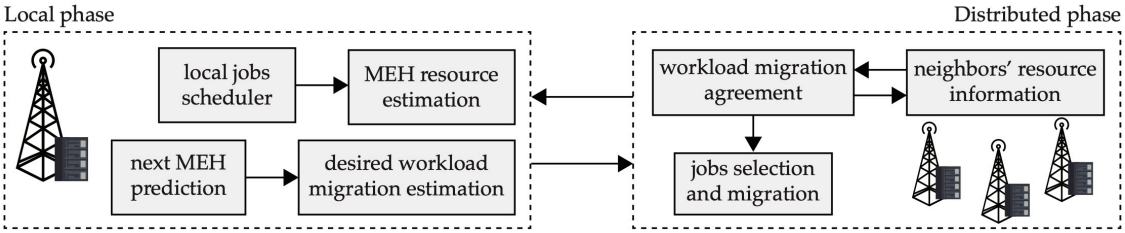


Figure 6.1: High level diagram of EASE. The local steps are detailed on the left and provide the resource and the desired workload migration estimates for each MEH in isolation. The distributed algorithm on the right allows the MEH to reach a consensus on the jobs allocation and trigger their migration.

a MEH. $\mathcal{V}$ represents the set of vehicles moving within the city, which are constantly connected to the nearest eNB node (providing communication support). Vehicle $v \in \mathcal{V}$ sends computing job requests to the closest MEH, which can locally execute the required workload or offload it, either partially or in full, to neighboring MEHs. Also, each vehicle can have a single outstanding job instance (being processed) and can generate a single job request at any time slot only if the previous request has been either fully processed or dropped by the serving MEH. For this reason, in the following analysis we will interchangeably identify a vehicle with the associated outstanding job to be computed. The set of neighboring eNBs to eNB i is denoted by $\mathcal{N}_i$. Jobs are executed through the instantiation of containers, which reserve the needed computing and memory resources. In this work, containers are favoured over VMs due to their lower memory footprint, which permits a faster migration process – a desirable feature in the considered mobile scenario [146]. Jobs that are being executed on one MEH but associated with vehicles that are about to leave the eNB/MEH coverage area are assessed by the migration controller. The latter decides whether to migrate their execution to another (target) MEH, or to finish it locally and send the processing result to the vehicle in a multi-hop fashion (from the old to the new serving eNB). eNBs are equipped with energy harvesting devices, e.g., photovoltaic panels (PVs), whose collected energy is managed by the system. Besides, eNBs are also connected to the power grid as relying only upon harvested energy would be risky due to its intermittent nature. The diagram of an eNB/MEH node is shown in Fig. 6.2.

Next, we detail the mathematical models for computing and communication services, along with the statistical processes involved in the envisioned scenario and the system constraints. The mathematical notation is summarized in Table 6.1.

6.4.1 Computation and communication models

Computing job parameters. At time t, each job k served by MEH i is characterized by the triplet $(I_{i,k}(t), D_{i,k}(t), S_{i,k}(t))$, where i) $I_{i,k}(t)$ is the residual job intensity, expressed in CPU cycles, ii) $D_{i,k}(t)$ is the residual (hard) execution deadline, in seconds, i.e., the time still available to execute the job, and iii) $S_{i,k}(t)$ is the remaining data to be processed, in bits. As the job is processed by the server,

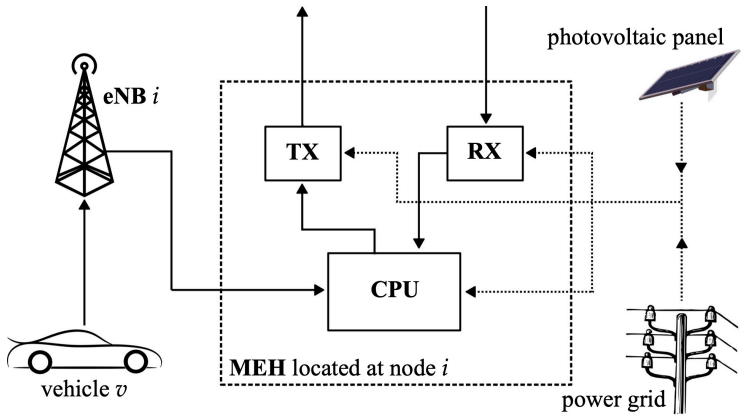


Figure 6.2: eNB/MEH node. Job requests arrive from connected vehicles v moving within the eNB coverage area. Containers handling the execution of the jobs are created at the serving MEH, and possibly migrated to other MEHs in case the associated vehicles exit the eNB coverage area.

the intensity, deadline and data size decrease in accordance with the following state equations

$$I_{i,k}(t+\tau) = I_{i,k}(t) - w_{i,k}(t)\,, \tag{6.1}$$

$$D_{i,k}(t+\tau) = D_{i,k}(t) - \tau\,, \tag{6.2}$$

$$S_{i,k}(t+\tau) = S_{i,k}(t) - \frac{S_{i,k}(0)}{I_{i,k}(0)} w_{i,k}(t)\,, \tag{6.3}$$

where $w_{i,k}(t)$ is the amount of workload belonging to job k and processed by MEH i during slot t. Note that (6.1) makes it possible to rewrite (6.3) as

$$S_{i,k}(t) = \frac{S_{i,k}(0)}{I_{i,k}(0)} I_{i,k}(t), \tag{6.4}$$

where the factor $S_{i,k}(0)/I_{i,k}(0)$ regulates the linear correlation between the remaining job intensity $I_{i,k}(t)$ and the data that still has to be processed for job k, $S_{i,k}(t)$.

Communication models. For the 5G wireless links between the eNBs and the vehicles we adopt i) the massive-MIMO energy consumption model of [154], and ii) the mm-wave – 28 GHz – urban NLoS channel model of [123]. Specifically, from [154] the following system parameters are obtained: i) the power needed to keep the wireless unit switched on (fixed circuit power consumption), P_{RAN}, ii) the energy required per transmitted bit via wireless links, E_b^{RAN}, iii) the fixed wired circuit power consumption, P_{wired}, iv) the energy expenditure for the wired backhaul links connecting the eNBs, E_b^{wired}. Note that the vehicles' energy utilization is not involved in the scheduling and, in turn, only the energy consumption at the eNB side is considered. The model in [123] is used for the vehicle-eNB association.

Container migration model. The migration of a container requires the hosting MEH to spend energy for freezing the status of the virtual entity and prepare the data to be sent to the target

Symbol	Meaning	Unit
$v \in \mathcal{V}$	vehicle identifier (ID) and set of vehicles	-
$i \in \mathcal{N}$	eNB/MEH ID and set of eNBs/MEHs	-
$\mathcal{N}_i$ and N_i	set of neighboring nodes of node i and its cardinality $\lvert\mathcal{N}_i\rvert$	-
$k \in \mathcal{K}_i(t)$ and $K_i(t)$	job ID, set of jobs in execution at MEH i at slot t, and its cardinality $\lvert\mathcal{K}_i(t)\rvert$	-
$\hat{\mathcal{K}}_{ij}$ and $\hat{K}_{ij}$	set of jobs running on MEH i with probable next MEH j and its cardinality	-
T	no. of slots in the prediction horizon	-
$t = [0, \ldots, T]$	scheduling time slot index	-
τ	length of a scheduling slot	s
$V_i(t)$ and $C_i(t)$	no. of results to be sent in the coverage area of eNB i at slot t and to be routed through the backhaul network	-
I_k	intensity of job k	cyc.
D_k	deadline of job k	s
S_k	size of job k	bit
p_v and p_ℓ	job generation probability and probability that it is of type ℓ	-
$\mathbf{p}_{i,k}(t)$	handover prob vector for vehicle v (job k) at slot t	
$w_{i,k}(t)$	workload of job k processed by MEH i in slot t	cyc.
L	(fixed) size of a container instantiated on a MEH	bit
E_b^{RAN}	energy per bit for eNB-vehicle wireless transmissions	J/bit
E_b^{wired}	energy per bit for eNB-eNB wired transmission	J/bit
σ_s and σ_d	energy per bit for migration at the source (destination) MEH	J/bit
E_s and E_d	(fixed) energy for migration at the source (destination) MEH	J
$E_i^{\text{H}}(t)$	harvested energy available at slot t	J
$P_i^{\text{PV}}(t)$	power supplied by the PV at node i, instant t	W
P_{RAN} and P_{wired}	(fixed) power to keep the wireless (wired) unit switched on	W
P_i^{idle}	(fixed) power to keep the server switched on	W
$N_i^{\text{inc}}(t)$ and $N_i^{\text{out}}(t)$	no. of MEH incoming (outgoing) jobs at slot t	-
F_i	maximum computational power of server i	W
M_i	maximum amount of RAM available at server i	bit
$\bar{w}_{ij}$	desired intensity requested by MEH i to neighbor j	cyc./s
$\bar{m}_{ji}$	memory space requested by MEH i to neighbor j	bit
$\hat{P}_i^{\text{H}}$	residual green power at node i after the local scheduling	W
$\hat{F}_i$	residual computing power at node i after local scheduling	W
$\hat{M}_i$	residual RAM memory at node i after the local scheduling	bit
o_{ij}	optimal amount of MEH i processing load to offload to j	cyc./s
$\tilde{o}_{ji}$	optimal processing load to be received at MEH i from j	cyc./s

Table 6.1: Summary of the symbols used within the chapter. "cyc." stands for "CPU cycles".

MEH for the correct re-instantiation. Hence, the target MEH has to spend energy to create the new virtual entity using the received information. The energy expenditure at the two sides consists of [155]: i) a contribution proportional to the size of the migration data, through the parameters σ_s and σ_d respectively, plus ii) a fixed energy contribution, equal to E_s for the source MEH and E_d for the target one, respectively. Additionally, the source spends some energy for transmitting the

data over the wired channel E_b^{wired}. Overall, it holds

$$E_{\text{source}}^{\text{migr}}(t) = \sigma_s L + E_b^{\text{wired}} S_k(t) + E_s, \quad \text{and} \tag{6.5}$$

$$E_{\text{dest}}^{\text{migr}}(t) = \sigma_d L + E_d, \tag{6.6}$$

where $S_k(t)$ is the (variable) data size associated with job k, and L is the (fixed) container size. According to [149], we account for a service downtime of T_k^{migr} when migrating the entities. In turn, T_k^{migr} seconds are additionally removed from the job's deadline $D_k(t)$ at every migration occurrence. Note that the delay associated with wired transmissions is negligible as compared to the service downtime.

6.4.2 Statistical processes

Energy harvesting model. We refer to $P_i^{\text{PV}}(t)$ as the power supplied by the PV co-located with eNB/MEH i at instant t and that varies from a minimum of $P_{\text{min}}^{\text{PV}}$ to a maximum of $P_{\text{max}}^{\text{PV}}$. Accounting for the power required to keep the server (P_i^{idle}) and the communication channels (P_{RAN} and P_{wired}) switched on, and the fixed amount of energy required for the container migration, the harvested energy available at eNB/MEH i for computations and data transmissions at time slot t is

$$\begin{aligned} E_i^{\text{H}}(t) = &\left(P_i^{\text{PV}}(t) - P_{\text{RAN}} - P_{\text{wired}} - P_i^{\text{idle}}\right)\tau - N_i^{\text{inc}}(t)\left(\sigma_d L + E_d\right) + \\ &- N_i^{\text{out}}(t)\left[\left(\sigma_s + E_b^{\text{wired}}\right)L + E_s\right] \end{aligned} \tag{6.7}$$

where $N_i^{\text{inc}}(t)$ and $N_i^{\text{out}}(t)$ are the known number of MEH incoming and outgoing jobs at MEH i and time t, which are scheduled at the previous step $t-\tau$. The terms in Eqs. (6.5)-(6.6) that depend on the data size $S_{i,k}(t)$ are not considered in $E_i^{\text{H}}(t)$ as they will be integrated in the optimization function (see Eq. (6.18)). Note that being $E_i^{\text{H}}(t)$ a difference between the harvested energy and that required to deliver the services, its value can be negative. $E_i^{\text{H}}(t)$ is known for the current slot t only. However, the developed MPC framework also needs estimates for $[E_i^H(t+\tau), \ldots, E_i^H(t+\tau(T-1))]$, within the time window $t+\tau, \ldots, t+\tau(T-1)$, where T is the prediction horizon. These estimates are computed by forecasting the time dependent quantities in (6.7): future values of $P_i^{\text{PV}}(t+\cdot)$ are estimated using a Gaussian r.v. with average P_{PV} and standard deviation σ_{PV}, estimates for the number of incoming $N_i^{\text{inc}}(t+\cdot)$ and outgoing $N_i^{\text{out}}(t+\cdot)$ jobs at eNB i in slot t are obtained considering t[h!]he vehicles in the external annulus of the eNB's coverage area. Finally, P_i^{idle} depends on the specific MEH characteristics at eNB i, as specified in Section 6.7.

Jobs types and arrival model. Three types of jobs are considered in this work, having different intensities, deadlines, and data sizes and identified through the index $\ell = \{1, 2, 3\}$. Every job type is associated with a generation triplet (I^ℓ, D^ℓ, S^ℓ), and a generation probability p_ℓ. Each vehicle $v \in \mathcal{V}$ can submit at most one computing job at a time to the network facilities, so that a bijective mapping vehicle-job ID can be derived. Once a job is finished or expired, the vehicle submits a new job to the MEH with probability p_v at each slot. This parameter is tuned in the simulations. Also in this case, for predictive optimization, an estimate for the future incoming jobs is needed.

To this purpose, a circular buffer containing the values of $I_{i,k}/D_{i,k}$ of the newly generated jobs is kept. A fixed estimate of the average of the last W seconds is used to predict the incoming traffic. In **perin2021towards**, the authors verified that even simple predictors are still effective with MPC if T is large enough.

Handover probabilities. Each job k is associated with a probability vector that depends on the position of the vehicle v requesting the service. Being i the serving eNB for vehicle v, we define $\mathbf{p}_{i,k}(t)$ as the N_i-dimensional vector containing the probabilities that vehicle v will hand over to any of the $|\mathcal{N}_i| = N_i$ neighboring radio cells, i.e., $\mathbf{p}_{i,k}(t) = [p_{i1,k}(t), p_{i1,k}(t), \ldots, p_{iN_i,k}(t)]$, with $\sum_j p_{ij,k} = 1$. Vector $\mathbf{p}_{i,k}(t)$ is updated every time a new trajectory sample is available for the associated vehicle v, either inside the same cell or in a new cell after performing the handover.

6.4.3 System constraints

The set $\mathcal{K}_i(t)$, with cardinality$K_i(t) = |\mathcal{K}_i(t)|$, collects the jobs being executed at time slot t at MEH i. The following systems constraints apply

Processing capacity. Indicating with F_i the maximum computing power of server i – expressed in CPU cycles per second – the following inequality on the sum of the workloads holds

$$\frac{1}{\tau} \sum_{k=1}^{K_i(t)} w_{i,k}(t) \leq F_i \,. \tag{6.8}$$

Storage capacity. Being M_i [bits] the maximum amount of RAM available at server i, the sum of the data sizes $S_{i,k}(t)$ of all the active jobs at MEH i must obey

$$\sum_{k=1}^{K_i(t)} S_{i,k}(t) \leq M_i \,. \tag{6.9}$$

Job execution time. In case the deadline of job k, $D_{i,k}(t)$, expires in the current time slot t, the job must be processed entirely and immediately at server i and cannot be further migrated, i.e.,

$$w_{i,k}(t) = I_{i,k}(t) \qquad \text{if } D_{i,k}(t) \leq \tau \,. \tag{6.10}$$

This guarantees the timely delivery of the computation result to the requesting vehicle, avoiding that the outcome becomes useless. As Eqs. (6.8)-(6.10) may not be jointly satisfied, in the following we will relax Eq. (6.8).

Workload conservation. Finally, note that, in general, the inequalities

$$0 \leq w_{i,k}(t) \leq I_{i,k}(t), \ \forall\, i \in \mathcal{N}, \ \forall\, k \in \mathcal{K}_i(t), \ \forall\, t \tag{6.11}$$

must always hold, because of the workload conservation principle.

6.5 Problem formulation

6.5.1 Step 1: Local controller and resources estimation

Each MEH $i \in \mathcal{N}$ estimates $w_{i,k}(t)$ for every job $k \in \mathcal{K}_i(t)$ to be executed at time t: in the analysis, $w_{i,k}(t)$ stands for the optimal fraction of computing intensity $I_{i,k}(t)$ to be locally executed at time slot t for the hosted job k. We define vectors $\mathbf{w}_i(t)$, $\mathbf{I}_i(t)$ and $\mathbf{D}_i(t)$ respectively collecting $w_{i,k}(t)$, $I_{i,k}(t)$ and $D_{i,k}(t)$ for all $k \in \mathcal{K}_i(t)$. As for the energy spent to transmit the processing results back to the vehicles, $V_i(t)\, E_b^{\mathrm{RAN}}$ is the (per bit) energy cost of sending the results to the $V_i(t)$ vehicles in the wireless coverage area, while $C_i(t)\, E_b^{\mathrm{wired}}$ is the energy cost entailed in routing the $C_i(t)$ jobs that are completed at node i and that have to be routed via the backhaul links to reach the corresponding user (vehicle). R_k is the size of the processing result of job k, and q_i^{proc} is the energy cost of processing a unit of workload.

Given these quantities, we define two local (at node i) functions $f_i(\cdot)$ and $g_i(\cdot)$, as follows.

$$f_i(\mathbf{w}_i; V_i, C_i, E_{\mathrm{H},i}) = q_i^{\mathrm{proc}} \mathbf{1}^T \mathbf{w}_i(t) + V_i(t)\, E_b^{\mathrm{RAN}} R_k + C_i(t)\, E_b^{\mathrm{wired}} R_k - E_i^{\mathrm{H}}(t)\,, \tag{6.12}$$

$$g_i(\mathbf{I}_i(t); \mathbf{D}_i(t)) = \sum_{k=1}^{K_i(t)} \left(\frac{I_{i,k}(t)}{D_{i,k}(t)} \right)^2 . \tag{6.13}$$

$f_i(\cdot)$ quantifies the difference between the total energy expenditure at node i in slot t (due to processing and communications processes) and the energy that is locally harvested at this node. Hence, $-f(\mathbf{w}_i; \cdot)$ represents the residual cost-free energy available for the migration process in the distributed phase. Minimizing $f_i(\cdot)$ corresponds to maximizing the local energy available at the node. $g_i(\cdot)$ represents the residual processing cost, which is proportional to $(I_{i,k}/D_{i,k})^2$. Minimizing $g_i(\cdot)$ forces the node to execute the jobs, especially prioritizing those with high intensity and whose deadline is about to expire. Note also that, due to Eq. (6.1), $I_{i,k}(t)$ depends on the optimization variable $w_{i,k}$ at previous time slots.

Considering a forecast optimization window of T slots into the future, and letting $t = 0$ be the current time slot, the local cost function at node i over the whole time horizon is formulated by combining $f_i(\cdot)$ and $g_i(\cdot)$, as

$$J_i\left(\mathcal{W}_i, \mathcal{I}_i; \mathcal{D}_i, \mathbf{V}_i, \mathbf{C}_i, \mathbf{E}_i^{\mathrm{H}}\right) = \gamma \sum_{t=0}^{T-1} g_i(\mathbf{I}_i(t); \mathbf{D}_i(t)) + \sum_{t=0}^{T-1} \max\{f_i(\mathbf{w}_i; \cdot), 0\}^2, \tag{6.14}$$

where $\mathcal{W}_i$, $\mathcal{I}_i$ and $\mathcal{D}_i$ represent the stacks of vectors $\mathbf{w}_i(t)$, $\mathbf{I}_i(t)$ and $\mathbf{D}_i(t)$ over the considered horizon T, respectively, while $\mathbf{V}_i$, $\mathbf{C}_i$ and $\mathbf{E}_i^{\mathrm{H}}$ are the vectors collecting $V_i(t)$, $C_i(t)$ and $E_i^{\mathrm{H}}(t)$ for $t \in \{0, \tau, \ldots, \tau(T-1)\}$. The coefficient $\gamma > 0$ is used to balance the processing state cost term ($g_i(\cdot)$) with respect to the energy cost ($f_i(\cdot)$).

Remark 1. *From a physical perspective, the processing energy consumption is not necessarily a quadratic function, but it varies based on the specific computing architecture [156]. A quadratic function for $f_i(\cdot)$*

was chosen, as it promotes smoothness of the controller in the transitions from a slot to the next one, and has the same curvature order of the processing state cost $g_i(\cdot)$. Also, the $\max\{\cdot\}$ *function is used to make the cost positive only when $f_i(\cdot) > 0$, i.e., the renewable energy is fully used and the node has to resort to the power grid.*

Next, the cost function in Eq. (6.14) is modified through the addition of a penalty term proportional to two new auxiliary variables ${}_i(t) = [\delta_{F_i}(t), \delta_{M_i}(t)]$, to ensure that the problem does not become infeasible when resources are scarce. Therefore, rewriting the constraints (6.8) and (6.9), we define for each MEH the following local problem at node i,

$$
\begin{aligned}
P_i^{\text{loc}}: \quad \min_{\mathcal{W}_i, {}_i} \quad & J_i\left(\mathcal{W}_i, {}_i; \cdot\right) + c_i \sum_{t=0}^{T-1} {}_i(t) \\
\text{s.t.} \quad & (6.1)\text{-}(6.3),\ (6.10),\ (6.11), \\
& \frac{1}{\tau} \sum_{k \in \mathcal{K}_i} w_{i,k}(t) - \delta_{F_i}(t) \leq F_i, \\
& \sum_{k \in \mathcal{K}_i} S_{i,k}(t) - \delta_{M_i}(t) \leq M_i,
\end{aligned}
\tag{6.15}
$$

where $c_i > 0$ is the coefficient weighting the penalty variables. By solving (6.15), each MEH obtains the optimal control $\mathbf{w}_i(0)$ which is implemented in the current time step.

6.5.2 Step 2: workload migration

From (6.15), each server estimates its own future energy and processing resources. Specifically, let $\hat{P}_i^{\text{H}}$ be the residual available green power, possibly negative if the grid support is sought, $\hat{F}_i$, and $\hat{M}_i$ be the residual computational power, and RAM memory at node i, respectively. Note that, since constraints (6.8) and (6.9) are relaxed in (6.15), $\hat{F}_i$ and $\hat{M}_i$ can be negative. These estimates are obtained averaging the values over the prediction horizon, excluding the current instant $t = 0$. Due to this averaging operation, while in (6.15) we deal with energy expenditures, in the following we refer to power quantities.

The migration task presents itself as a combinatorial mixed integer programming (MIP) problem, which is non-convex and is generally difficult to solve in a distributed fashion. Thus, we use heuristics to derive close-to-optimal solutions. In this work, the popular *relax and round* method is used, which consists in solving the convex counterpart of the original problem, and rounding the result to a feasible solution afterwards.

Based on the handover probability vector $\mathbf{p}_{i,k}$ presented in Section 6.4.2, each MEH determines the average resource demand requested from its neighbors in the migration process. Specifically,

the CPU cycles per second and memory space that are requested from neighbor j are

$$\bar{w}_{ij} = \sum_{k \in \hat{\mathcal{K}}_{ij}} \frac{I_{i,k}}{D_{i,k}}, \qquad \text{and} \tag{6.16}$$

$$\bar{m}_{ij} = \sum_{k \in \hat{\mathcal{K}}_{ij}} S_{i,k}, \tag{6.17}$$

respectively, where $\hat{\mathcal{K}}_{ij}$ represents the set of jobs that are currently running at server i, associated with vehicles that are about to exit the coverage area of the co-located eNB i and whose most probable next eNB is co-located with MEH j. With $\bar{\mathbf{w}}_i = [\bar{w}_{i1}, \ldots, \bar{w}_{iN_i}]$ we denote the vector collecting the *desired* processing intensity per second to be sent to each of the N_i neighbors of MEH i, computed via (6.16). We also introduce the new optimization variables $\mathbf{o}_i = [o_{i1}, \ldots, o_{iN_i}]$ and $\tilde{\mathbf{o}}_i = [\tilde{o}_{1i}, \ldots, \tilde{o}_{N_i i}]$ representing the optimal total amount of processing load to be sent to, and to be received from each neighbor, respectively. The deviation from the desired $\bar{\mathbf{w}}_i$ to be migrated is penalized with the l_2-norm $\|\bar{\mathbf{w}}_i - \mathbf{o}_i\|^2$, and the migration cost is defined as

$$\Gamma_i\left(\mathbf{o}_i, \tilde{\mathbf{o}}_i; \bar{\mathbf{w}}_i, \hat{P}_i^{\mathrm{H}}\right) = \max\left\{\left(q_i^{\mathrm{tx}} - q_i^{\mathrm{proc}}\right) 1^T \mathbf{o}_i \; + \left(q_i^{\mathrm{rx}} + q_i^{\mathrm{proc}}\right) 1^T \tilde{\mathbf{o}}_i - \hat{P}_i^{\mathrm{H}}, 0\right\} + \rho \|\mathbf{o}_i - \bar{\mathbf{w}}_i\|^2, \tag{6.18}$$

where q_i^{proc}, q_i^{tx} and q_i^{rx} are the processing, transmission and reception costs of server i (expressed as powers), respectively. The $\max\{\cdot\}$ term represents the power needed from the power grid to migrate the jobs, whereas the quadratic term encodes the fact that the optimal $\mathbf{o}_i$ should be as close as possible to the desired $\bar{\mathbf{w}}_i$ – this corresponds to moving the jobs to the next serving eNB. Finally, $\rho > 0$ is a weight that balances the importance of the two cost terms. Note that minimizing Eq. (6.18) returns a solution $\mathbf{o}_i$ that matches vector $\bar{\mathbf{w}}_i$ if the residual harvested power is sufficient and the constraints are satisfied. Specifically, as system constraint we consider the following variation of (6.8) and (6.9),

$$\sum_{j \in \mathcal{N}_i} \left(\tilde{o}_{ji} - o_{ij}\right) \leq \min\{\hat{F}_i, \xi_{M_i} \hat{M}_i\} + \hat{\delta}_i, \qquad \forall\, i \in \mathcal{N}. \tag{6.19}$$

Remark 2. *The meaning of this constraint is that the workload surplus that server i has during the following time steps, i.e., the* incoming *workload minus the* outgoing *one, should satisfy the average (long-term) power ($\hat{F}_i$) and memory ($\hat{M}_i$) availability at node i. The coefficient ξ_{M_i} relates the memory availability to the residual computational power. This follows from the assumption of direct proportionality between the data size S_k and the processed workload w_k.*

Since the general goal is to minimize the energy drained network-wide from the power grid, a cost function that represents the global welfare and at that at the same time is amenable to a distributed solution is the sum

$$\Gamma\left(\mathbf{o}, \tilde{\mathbf{o}}, \hat{;} \bar{\mathbf{w}}, \hat{\mathbf{P}}^{\mathrm{H}}\right) = \sum_{i \in \mathcal{N}} \left[\Gamma_i\left(\mathbf{o}_i, \tilde{\mathbf{o}}_i; \bar{\mathbf{w}}_i, \hat{P}_i^{\mathrm{H}}\right) + \hat{c}_i \hat{\delta}_i^2\right], \tag{6.20}$$

where $\hat{c}_i > 0$ is the cost coefficient associated with the penalty term $\hat{\delta}_i^2$. This leads to the constrained optimization problem

$$\begin{aligned} P^{\text{glob}}: \quad \min_{\mathbf{o},\tilde{\mathbf{o}},\hat{}} \quad & \Gamma\left(\mathbf{o},\tilde{\mathbf{o}},\hat{};\bar{\mathbf{w}},\hat{\mathbf{P}}^{\text{H}}\right) \\ \text{s.t.} \quad & \mathbf{o},\tilde{\mathbf{o}},\hat{} \geq 0, \ (6.19), \\ & o_{ij} = \tilde{o}_{ij} \ \forall i,j, \end{aligned} \tag{6.21}$$

with $\mathbf{o}$, $\tilde{\mathbf{o}}$,$\hat{}$, $\bar{\mathbf{w}}$ and $\hat{\mathbf{P}}^{\text{H}}$ being the vectors collecting $\mathbf{o}_i$, $\tilde{\mathbf{o}}_i$, $\hat{\delta}_i$, $\bar{\mathbf{w}}_i$ and $\hat{P}_i^{\text{H}}$ respectively, for all the MEHs $i \in \mathcal{N}$. The equality $o_{ij} = \tilde{o}_{ij}$ is called *consensus constraint* and is added to ensure that the amount of workload exiting node i and directed to j is equal to the one that j expects to receive from i.

6.6 Final scheduling solution via local and distributed processes

6.6.1 Step 1: local MPC solution

At each MEH, the local MPC problem of (6.15) is solved over the whole horizon T [157]. MPC uses the *receding horizon* technique, which consists of solving the given problem within a prediction window of size T, applying the optimal computed control only for the current time step $t = 0$, moving forward the optimization window by one time slot (τ seconds) and repeating the procedure. In this way, the controller progressively adapts to new observations and estimates of the exogenous processes. In the considered setup, at any given instant, MEH i computes the optimal policy throughout the whole horizon of T slots, but only $\mathbf{w}_i(0)$ is applied as the control action. The exogenous processes are the future jobs and the harvested energy availability, see Section 6.4.2.

6.6.2 Step 2a: distributed workload migration

In the following, the scheduling slot index t is omitted in the interest of readability. Eq. (6.21) is a *consensus* problem, i.e., it entails the reaching of an agreement on the value of some variables among multiple agents in a distributed system. In our context, the MEHs must agree on the amount of processing load to exchange among each other. A way to solve this problem – written as the sum of separable convex cost functions – is via the *dual ascent* algorithm **bertsekas1999nonlinear**. Given a generic cost function $\psi(\mathbf{x})$, its *Lagrangian* is defined as

$$\mathcal{L}(\mathbf{x},\mathbf{z}) = \psi(\mathbf{x}) + \mathbf{z}^T(A\mathbf{x} - \mathbf{d}), \tag{6.22}$$

where $\mathbf{z}$ are the Lagrange multipliers associated with the constraints $A\mathbf{x} = \mathbf{d}$. The dual ascent solves the problem by iteratively i) minimizing $\mathcal{L}(\mathbf{x},\mathbf{z})$ with respect to $\mathbf{x}$ (primal step), and ii) updating the value of $\mathbf{z}$ (dual step). To formalize the solution of problem (6.21) via dual ascent,

we split the local cost functions (6.18) as

$$\begin{aligned}\tilde{\Gamma}_i\left(\mathbf{o}_i,\tilde{\mathbf{o}}_i,\hat{\delta}\right) = &\max\left\{\left(q_i^{\text{tx}}-q_i^{\text{proc}}\right)1^T\mathbf{o}_i+\left(q_i^{\text{rx}}+q_i^{\text{proc}}\right)1^T\tilde{\mathbf{o}}_i-\hat{P}_i^{\text{H}},0\right\}+\\ &+\frac{\rho}{2}\|\mathbf{o}_i-\bar{\mathbf{w}}_i\|^2+\frac{\rho}{2}\|\tilde{\mathbf{o}}_i-\tilde{\mathbf{w}}_i\|^2+\hat{c}_i\hat{\delta}_i^2,\end{aligned} \tag{6.23}$$

exploiting the fact that $o_{ij}=\tilde{o}_{ij}$, and defining $\tilde{\mathbf{w}}_i=\{\bar{w}_{ji}\,|\,j\in\mathcal{N}_i\}$. Intuitively, node i is responsible for half of the quadratic cost from its neighbors and for half of its own local cost. For compactness, let $\mathbf{x}=\{\mathbf{x}_i=[\mathbf{o}_i,\tilde{\mathbf{o}}_i,\hat{\delta}_i],\,\forall\, i\in\mathcal{N}\}$ be the global optimization variable, $\mathbf{b}_i=[\bar{\mathbf{w}}_i,\tilde{\mathbf{w}}_i,0]$ the tracking target vector, and $\mathbf{q}_i=[\mathbf{q}_i^{\text{tx}}-\mathbf{q}_i^{\text{proc}},\mathbf{q}_i^{\text{rx}}+\mathbf{q}_i^{\text{proc}},0]$ the linear costs vector. Moreover, we define matrix $Q_i=I_{2N_i+1}\mathbf{m}_i$, with $\mathbf{m}_i=[\frac{\rho}{2},\dots\frac{\rho}{2},\hat{c}_i]$, and the global block diagonal matrix Q, collecting each Q_i on the diagonal. With these definitions, problem (6.21) can be expressed in the following form

$$\min_{\mathbf{x}}\quad\sum_{i\in\mathcal{N}}\left(\|\mathbf{x}_i-\mathbf{b}_i\|_{Q_i}^2+\max\left\{\mathbf{q}_i^T\mathbf{x}_i-\hat{P}_i^{\text{H}},0\right\}\right) \tag{6.24}$$

$$\text{s.t.}\quad A_1\,\mathbf{x}\le\mathbf{d}, \tag{6.25}$$

$$A_2\,\mathbf{x}=0, \tag{6.26}$$

where $\|\mathbf{x}\|_Q^2=\mathbf{x}^TQ\mathbf{x}$. The inequalities (6.25) collect (6.19) and the non-negativity constraints $\mathbf{o},\tilde{\mathbf{o}},\hat{}\ge 0$, while the equalities (6.26) correspond to the consensus constraints $o_{ij}=\tilde{o}_{ij}$, $\forall i\in\mathcal{N}$, $j\in\mathcal{N}_i$. Here, the matrices A_1 and A_2 are used to select the concerned variables, whereas $\mathbf{d}=\{\mathbf{d}_i=[\min\{\hat{F}_i,\xi_{M_i}\hat{M}_i\},0]\,|\,i\in\mathcal{N}\}$. We can now write the Lagrangian as

$$\mathcal{L}\left(\mathbf{x},\mathbf{y},\mathbf{z}\right)=\sum_{i\in\mathcal{N}}\tilde{\Gamma}_i\left(\mathbf{x}_i;\mathbf{b}_i,\hat{P}_i^{\text{H}}\right)+\mathbf{y}^T\left(A_1\mathbf{x}-\mathbf{d}\right)+\mathbf{z}^TA_2\mathbf{x}\,, \tag{6.27}$$

where $\mathbf{y}=\{\mathbf{y}_i=[\lambda_{i,\,i},\tilde{}_i,\hat{\varphi}_i]\,|\,i\in\mathcal{N}\}$ are the Lagrange multipliers associated with the inequality constraints (6.25), and $\mathbf{z}=\{\mathbf{z}_i=\,_i\,|\,i\in\mathcal{N}\}$ are the multipliers associated with equalities (6.26). Specifically, the Lagrange multipliers λ_i refer to constraints (6.19), $_i=\{\gamma_{ij}\}$, $\tilde{}_i=\{\tilde{\gamma}_{ji}\}$ and $\hat{\varphi}_i$ to $\mathbf{o}_i\ge 0$, $\tilde{\mathbf{o}}_i\ge 0$, and $\hat{\delta}_i\ge 0$, respectively, and $_i=\{\mu_{ij}\}$ to $o_{ij}=\tilde{o}_{ij}$, for every server $i\in\mathcal{N}$, and $j\in\mathcal{N}_i$. Using the $^+$ sign to denote the update at the following iteration, we detail in Algorithm 6.1 the dual ascent procedure that solves the problem

$$\inf_{\mathbf{x}}\sup_{\mathbf{y}\ge 0,\mathbf{z}}\ \mathcal{L}\left(\mathbf{x},\mathbf{y},\mathbf{z}\right). \tag{6.28}$$

Algorithm 6.1 Dual ascent algorithm solving problem (6.21)

1: $\mathbf{x}^+=\text{argmin}_{\mathbf{x}}\ \mathcal{L}\left(\mathbf{x},\mathbf{y},\mathbf{z}\right)$ {primal step}
2: $\mathbf{y}^+=\max\left\{\mathbf{y}+_{\mathbf{y}}\left(A_1\,\mathbf{x}^+-\mathbf{d}\right),0\right\}$ {dual update}
3: $\mathbf{z}^+=\mathbf{z}+_{\mathbf{z}}A_2\,\mathbf{x}^+$ {dual update}

The dual update requires in this case two different forms, depending on whether the constraint

is an equality or an inequality one. Inequality constraints may actually be inactive, and the associated Lagrange multipliers would be null in this case. The parameters $_{\mathbf{y}}$ and $_{\mathbf{z}}$ in the algorithm tune the stability and the convergence speed. The presented compact version of the dual ascent translates into the following local procedure, from a server perspective. Defining the vectors $\tilde{}_i = \{\mu_{ji}\}$ and $\bar{\mathbf{o}}_i = \{\tilde{o}_{ij}\}$ collecting variables that are kept in memory by the neighborhoods of i, the local Lagrangian at node i is

$$\begin{aligned}\mathcal{L}_i\left(\mathbf{o}_i, \tilde{\mathbf{o}}_i; \bar{\mathbf{w}}_i, \tilde{\mathbf{w}}_i, \hat{P}_i^{\mathrm{H}}\right) = \tilde{\Gamma}_i\left(\mathbf{o}_i, \tilde{\mathbf{o}}_i; \bar{\mathbf{w}}_i, \tilde{\mathbf{w}}_i, \hat{P}_i^{\mathrm{H}}\right) + \lambda_i\left[1^T(\tilde{\mathbf{o}}_i - \mathbf{o}_i) - \hat{\delta}_i\right] + \\ - {}_i^T\mathbf{o}_i - \tilde{}_i^T\tilde{\mathbf{o}}_i + {}_i^T\mathbf{o}_i - \tilde{}_i^T\tilde{\mathbf{o}}_i - \hat{\varphi}_i\hat{\delta}_i,\end{aligned} \tag{6.29}$$

and the procedure is presented in Algorithm 6.2, where a fixed step size α is assumed.

Algorithm 6.2 Dual ascent from a server perspective

1: receive $\tilde{}_i = \{\mu_{ji}\}$ from the neighbors
2: $\mathbf{o}_i^+, \tilde{\mathbf{o}}_i^+, \hat{\delta}_i = \operatorname{argmin}_{\mathbf{x}_i}\ \mathcal{L}_i(\mathbf{x}_i, {}_i)$
3: send $\tilde{o}_{ji}^+$ to the correspondent neighbor j
4: $\lambda_i^+ = \max\left\{\lambda_i + \alpha\left(\sum_{j\in\mathcal{N}_i}\left(\tilde{o}_{ji}^+ - o_{ij}^+\right) - \hat{F}_i\right), 0\right\}$
5: $\hat{\varphi}_i^+ = \max\left\{\hat{\varphi}_i - \alpha\,\hat{\delta}_i^+, 0\right\}$
6: ${}_i^+ = \max\left\{{}_i - \alpha\,\mathbf{o}_i^+, 0\right\}$
7: $\tilde{}_i^+ = \max\left\{\tilde{}_i - \alpha\,\tilde{\mathbf{o}}_i^+, 0\right\}$
8: receive $\bar{\mathbf{o}}_i^+ = \{\tilde{o}_{ij}^+\}$ from the neighbors
9: ${}_i^+ = {}_i + \alpha\left(\mathbf{o}_i^+ - \bar{\mathbf{o}}_i^+\right)$
10: send μ_{ij}^+ to the correspondent neighbor j

Note that, to minimize the Lagrangian in the primal step at line 2, server i not only needs its own Lagrange multipliers, but also the introduced $\tilde{}_i$, which collects the μ_{ji} of neighbors $j \in \mathcal{N}_i$. Therefore, node i must first receive these multipliers from the neighborhood. Also, while updating ${}_i$ in the dual step at line 9, $\bar{\mathbf{o}}_i^+$ is needed, which collects the $\tilde{o}_{ij}^+$ variables kept by the neighborhood of i, and which are to be received after the computation of j's primal step ($\forall j \in \mathcal{N}_i$). Hence, this amounts in two communication rounds among neighbors per dual ascent iteration. The dual updates are computationally inexpensive, whereas the primal step requires to solve a local convex subproblem, which is complicated by the $\max\{\cdot\}$ operator in the cost function (6.23). Eventually, note that an additional communication is required at the beginning of the procedure, to inform the neighborhood about the values of $\tilde{\mathbf{w}}_i$.

Solution to the primal step (line 2). The solution of the local primal subproblems is computed in closed form, distinguishing three cases. We consider the local primal subproblems in compact form with variables $\mathbf{x}_i$, and collect the Lagrange multipliers of (6.29) in ${}_i = [\lambda_i 1, \varphi_i, {}_i, \tilde{}_i, \tilde{}_i]$, with

associated variables selection matrix A_i. We split $\mathcal{L}_i(\mathbf{x}_i;\cdot) = u_i(\mathbf{x}_i) + h_i(\mathbf{x}_i)$, so that

$$u_i(\mathbf{x}_i) = \|\mathbf{x}_i - \mathbf{b}_i\|^2_{Q_i} + {}^{T}_{i}A_i\mathbf{x}_i, \tag{6.30}$$

$$h_i(\mathbf{x}_i) = \mathbf{q}_i^T\mathbf{x}_i - \hat{P}_i^{\mathrm{H}}. \tag{6.31}$$

Proposition 2. *The solution of the primal step of problem* (6.21) *is computed as one of the mutually exclusive cases*

i) $\mathbf{x}_i^+ = \operatorname{argmin}_{\mathbf{x}_i} u_i(\mathbf{x}_i)$, *if* $h_i\left(\mathbf{x}_i^+\right) \leq 0$, *or*

ii) $\mathbf{x}_i^+ = \operatorname{argmin}_{\mathbf{x}_i} u_i(\mathbf{x}_i) + h_i(\mathbf{x}_i)$, *if* $h_i\left(\mathbf{x}_i^+\right) > 0$, *or*

iii) $\mathbf{x}_i^+ = \operatorname{argmin}_{\mathbf{x}_i} u_i(\mathbf{x}_i)$, *s.t.* $h_i(\mathbf{x}_i) = 0$.

Proof. i) and ii) correspond to the cases where the $\max\{\cdot\}$ operator in (6.23) is replaced by 0 or $h_i(\mathbf{x}_i)$, respectively. Once the optimum is computed, the feasibility check must be done: if the minimum lies in the feasible region, the solution is accepted. However, it can also be that these two optima are both infeasible: in this case, the optimal solution must lie on the plane $h_i(\mathbf{x}_i) = 0$, and a constrained problem has to be solved (case iii)). □

Remark 3. *It is impossible that both solutions i) and ii) are feasible, otherwise the convex function* (6.23) *would have two minima, which is an absurd due to its convexity.*

The solutions for each of the cases of Proposition 2 are now given in the following result.

Proposition 3. *Consider the three cases of Proposition 2. Their closed form optimal solutions are expressed as*

i) $\mathbf{x}_i^+ = \mathbf{b}_i - \frac{1}{2}Q_i^{-1}A_i^T{}_i$

ii) $\mathbf{x}_i^+ = \mathbf{b}_i - \frac{1}{2}Q_i^{-1}\left(A_i^T{}_i + \mathbf{q}_i\right)$

iii) $\mathbf{x}_i^+ = \mathbf{b}_i - \frac{1}{2}Q_i^{-1}\left(A_i^T{}_i + \mathbf{q}_i^T\frac{2Q_i\left(\mathbf{b}_i - \frac{\hat{P}_i^{\mathrm{H}}}{\|\mathbf{q}_i\|^2}\mathbf{q}_i\right) - A_i^T{}_i}{\|\mathbf{q}_i\|^2}\mathbf{q}_i\right)$

Proof. The proof is straightforward for cases i) and ii): it is sufficient to set the gradient of the function to zero. In the third case, it is necessary to solve the constrained minimization of $u(\mathbf{x}_i)$ subject to $h(\mathbf{x}_i) = 0$. The Lagrange multipliers method can be used, where the Lagrangian of case iii) is $\mathcal{L}'(\mathbf{x}_i, \eta_i) = u(\mathbf{x}_i) + \eta_i h(\mathbf{x}_i)$, and its primal solution is

$$\inf_{\mathbf{x}_i}\sup_{\eta_i}\ \|\mathbf{x}_i - \mathbf{b}_i\|^2_{Q_i} + {}^{T}_{i}A_i\mathbf{x}_i + \eta_i\left(\mathbf{q}_i^T\mathbf{x}_i - \hat{P}_i^{\mathrm{H}}\right). \tag{6.32}$$

The partial derivatives with respect to $\mathbf{x}_i$, and η_i are

$$\frac{\partial \mathcal{L}'(\mathbf{x}_i, \eta_i)}{\partial \mathbf{x}_i} = 2\, Q_i\, (\mathbf{x}_i - \mathbf{b}_i) + A_{ji}^T + \eta_i\, \mathbf{q}_i, \tag{6.33}$$

$$\frac{\partial \mathcal{L}'(\mathbf{x}_i, \eta_i)}{\partial \eta_i} = \mathbf{q}_i^T \mathbf{x}_i - \hat{P}_i^{\mathrm{H}}. \tag{6.34}$$

Setting them to zero, we obtain

$$\mathbf{x}_i = \mathbf{b}_i - \frac{1}{2}\left[Q_i^{-1}\left(A_{ji}^T + \eta_i\, \mathbf{q}_i\right)\right] = \frac{\hat{P}_i^{\mathrm{H}}}{\|\mathbf{q}_i\|^2}\mathbf{q}_i, \tag{6.35}$$

from which it is possible to derive the optimal value for the Lagrange multiplier

$$\eta_i^* = \frac{\mathbf{q}_i^T\left[2\, Q_i\left(\mathbf{b}_i - \frac{\hat{P}_i^{\mathrm{H}}}{\|\mathbf{q}_i\|^2}\mathbf{q}_i\right) - A_{ji}^T\right]}{\|\mathbf{q}_i\|^2}. \tag{6.36}$$

Now, plugging (6.36) into (6.35) returns the optimal value $\mathbf{x}_i^+$ of case iii). □

Remark 4. *For quadratic programs it is also possible to find a condition on the step size α for which the algorithm is ensured to converge. This only depends on the constraint matrices A_1 and A_2, and on the quadratic cost matrix Q defining the curvature. Since these values do not change among the three different primal optimization cases, a common condition can be obtained, i.e.,*

$$\alpha \leq \frac{2}{\left\| \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} Q^{-1} \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}^T \right\|}. \tag{6.37}$$

Proof. This result can be derived using proposition 2.3.2 of **bertsekas1999nonlinear**. □

6.6.3 Step 2b: rounding to a feasible discrete solution

In this section, we show how to compute the actual discrete allocation of jobs by obtaining new variables $\mathbf{o}_i^r$, which are the *rounded* versions of the $\mathbf{o}_i$ that were previously computed through consensus (see Section 6.6.2). In particular, $\mathbf{o}_i$ contains the optimal continuous amount of workload that each MEH would like to send to its neighbors. Instead, its rounded version $\mathbf{o}_i^r$ contains a feasible allocation accounting for the facts that the number of jobs and the possible ways of allocating them are discrete.

To compute the new $\mathbf{o}_i^r$, as an initial solution, we select the jobs from set $\hat{\mathcal{K}}_{ij}$, whose associated vehicle is about to migrate from eNB site i to j. The rounded $\mathbf{o}_i^r$ is thus initially set to $\bar{\mathbf{w}}_i$, assuming that the minimizer of the objective function (6.18) is the vector that minimizes the quadratic term. Then, the difference between this guess and the actual optimum obtained from the proposed dual ascent algorithm is computed, $\mathbf{o}_i^{\mathrm{diff}}$. For every neighbor j it is now clear whether more workload

is to be added to (in case $o_{ij}^{\text{diff}} < 0$) or removed from ($o_{ij}^{\text{diff}} > 0$) the initial guess o_{ij}^{r}. The jobs that were initially scheduled for migration to node j but that are eventually retained for computation at node i are those minimizing $\|\mathbf{o}_i^{\text{diff}}\|_1$. Instead, new jobs are added to the migration list using the prediction vectors $\mathbf{p}_{ij}$. In detail, the added jobs are those for which the handover probabilities towards j are maximized. A threshold ε_P is used to approximate the rounded solution, as the continuous optimum $\mathbf{o}_i$ will likely not coincide with any possible discrete approximation. The procedure is detailed in Algorithm 6.3.

Algorithm 6.3 Job-neighbor association

Input: mobility pattern predictions matrix P_i; optimal outgoing workload amount $\mathbf{o}_i$; set of the jobs $\mathcal{J}_i$ in execution at MEH i; tolerance threshold ε_P.
Output: job-neighbor association sets $\mathcal{Z}_{ij}\, \forall\, j \in \mathcal{N}_i$; rounded $\mathbf{o}_i^r$.
remove jobs $\{k \mid I_{i,k} < \varepsilon \vee D_{i,k} < 2\}$ from $\mathcal{J}_i$
$\mathbf{o}_i^r \leftarrow \bar{\mathbf{w}}_i$
$\mathcal{Z}_{ij} \leftarrow \hat{\mathcal{K}}_{ij}$
$\mathcal{J}_i \leftarrow \mathcal{J}_i \setminus \bigcup_{j \in \mathcal{N}_i} \hat{\mathcal{K}}_{ij}$
$\mathbf{o}_i^{\text{diff}} \leftarrow \mathbf{o}_i^r - \mathbf{o}_i$ {workload to be adjusted}
for all neighbors j in $\mathcal{N}_i$
  while $o_{ij}^{\text{diff}} > \varepsilon_P$
    $k \leftarrow$ job of $\mathcal{Z}_{ij}$ minimizing $\left| o_{ij}^{\text{diff}} \right|$
    remove job k from $\mathcal{Z}_{ij}$
    $o_{ij}^r \leftarrow o_{ij}^r - I_{i,k}/D_{i,k}$
    $o_{ij}^{\text{diff}} \leftarrow o_{ij}^{\text{diff}} - I_{i,k}/D_{i,k}$
    add job k to $\mathcal{J}_i$ {make it available for other neighbors}
  end while
  while $o_{ij}^{\text{diff}} < -\varepsilon_P$
    take job $k \in \mathcal{J}_i \mid k \in \text{argmax}\, \mathbf{p}_{ij}$ {most probable job from i to j}
    add job k to $\mathcal{Z}_{ij}$
    $o_{ij}^r \leftarrow o_{ij}^r + I_{i,k}/D_{i,k}$
    $o_{ij}^{\text{diff}} \leftarrow o_{ij}^{\text{diff}} + I_{i,k}/D_{i,k}$
    mask entry $p_{ij,k}$ {s.t. k is not selected again}
  end while
end for

6.6.4 Handling pathological cases

Since system constraints are made soft to avoid primal infeasibility, three pathological cases may occur, namely, 1) the optimal processed workload at the current instant exceeds the computational capacity; or 2) the data size for the currently running jobs do not fit the RAM memory; or 3) the deadline expires during the current slot, but the residual intensity is greater than zero. A greedy algorithm is developed to handle all of them. For the first two cases, the MEH ranks the active jobs through a double ordering criterion, considering as the first ranking criterion the time slot when they expire, and as the second their intensity (or data size). Next, it momentar-

ily pauses the execution of the services starting from the last one in the ordered list, until the resources suffice to proceed. In case 1, when pausing a job m, the amount of processed workload becomes $\sum_{k \in \mathcal{K}_i} w_{i,k} - w_{i,m}$, while in case 2, the data relative to suspended jobs is deleted from the RAM. The number of suspended jobs is the minimum such that the requirements are satisfied. Moreover, in case 1, it is likely that, when a job is suspended, additional computational power becomes available. In such a case, the new computational resources are assigned to the jobs that are closest to their deadline. Case 3 is managed considering the amount of residual intensity $I_{i,k}$. If $I_{i,k}$ is smaller than a threshold ε, then the deadline is extended by a small amount, so that the controller will privilege the execution of the corresponding job in the next slot. In this way, jobs are allowed to finish with a little additional delay (within one slot). If, however, the amount of residual intensity is larger than ε, the job is dropped, i.e., in this case the scheduling algorithm failed to provide an acceptable solution.

6.7 Numerical results

The proposed job scheduling algorithm is assessed in an emulated environment featuring 5G enabled vehicles moving within an urban scenario. Mobility traces are obtained with SUMO [29], an open-source traffic simulator that allows generating the movement of mobile users around a predefined city road map. Specifically, we use the "TAPAS Cologne" scenario, which mimics the vehicular traffic within the city of Cologne for a whole day on the basis of travelling habits of the city dwellers [121]. The mobile network is composed of 8 eNBs endowed with MEH functionalities, wired connected through optical links. The mobility area is covered with hexagonal cells with an eNB in the center, and with an inter-distance among nodes of 400 m. We generated and collected 24h long SUMO mobility traces with 25 ms granularity, for each of the 8 eNBs in the deployment. The first 15 hours were used to train and validate the mobility prediction algorithm, which is taken from [140], whereas the remaining ones to assess the performance of `EASE`. For the evaluation, we considered vehicles approaching the edge of the serving eNB coverage area, i.e., that are about to hand over to a new eNB/MEH. With the considered setup, this occurs, on average, when a user is less than 40 meters apart from the radio cell's border. The energy consumption of the MEHs is computed based on the SPECpower benchmark [156]. We selected two different edge computing platforms, namely, an HP ProLiant DL 110 Gen 10 Plus and a Nettrix R620 G40, obtaining two clusters of edge servers with different energy consumption, processing speed and memory, see Table 6.2. In Table 6.3, we report the jobs intensities, deadlines, data sizes, and generation probabilities, according to the system model of Section 6.4.2. The other system parameters are listed in Table 6.4.

In the following analysis, we compare our solution against two heuristic strategies: the first, termed "keep" never migrates jobs, which are always and entirely computed by the MEH that received the job processing request in the first place, i.e., the one co-located with the serving eNB at the time when the request was issued. The second heuristic, dubbed "migrate", whenever a vehicle gets close to the border of its serving eNB, always migrates the associated job to the most probable next eNB/MEH among the eNB neighbors.

	HP ProLiant DL 110	Nettrix R620 G40
idle power P_i^{idle}	94 W	110 W
max load power $P_i^{\max}$	299 W	468 W
computational power F_i	3.3 Gflops	7.6 Gflops
RAM memory M_i	64 GB	256 GB

Table 6.2: Servers specifications [156].

	I^ℓ [Gflop]	D^ℓ [s]	S^ℓ [GB]	p_ℓ
type 1	10	20	2	0.4
type 2	16	30	10	0.2
type 3	12	40	0.1	0.4

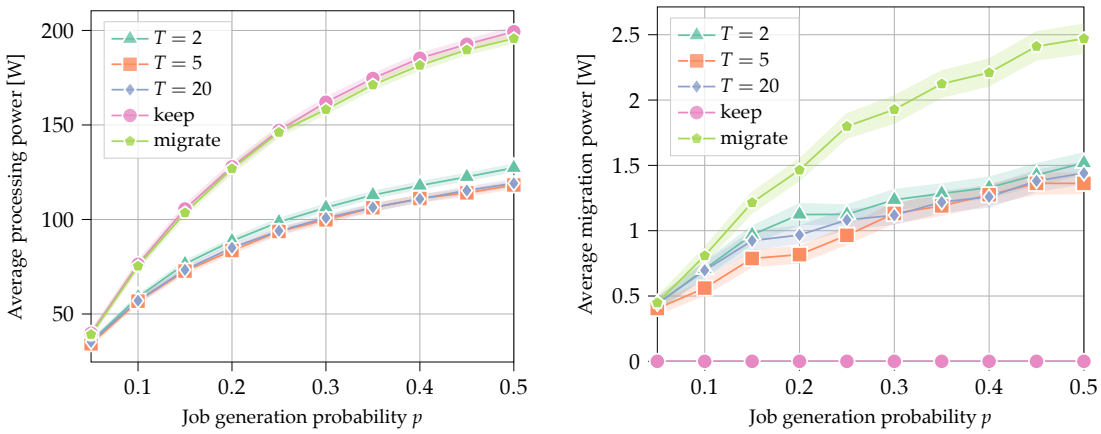
Table 6.3: Jobs parameters for the simulations.

Parameter	Value
number of nodes $\lvert\mathcal{N}\rvert$	8
fixed wireless circuit power consumption P_{RAN}	50.2 W
fixed wired circuit power consumption P_{wired}	20 W
energy per transmitted bit via wireless link E_b^{RAN}	1 nJ/bit
energy per transmitted bit via wired link E_b^{wired}	250 pJ/bit
PV panel minimum power $P_{\min}^{\text{PV}}$	250 W
PV panel maximum power $P_{\max}^{\text{PV}}$	400 W
PV panel average power P_{PV}	370 W
PV panel power std σ_{PV}	10 W
containers' size L	50 MB
weight parameters for L in (6.6) σ_s, σ_d	500 nJ/bit
fixed container migration energy expenditure E_s, E_d	250 mJ
delay associated with wired transmissions T_k^{migr}	2 s
window size to predict incoming traffic W	5 minutes
scheduler time slot τ	3 s
MPC horizon T	$\{2, 5, 20\}$
job generation probability p	0.25
weight of the soft constraint penalty c_i of (6.15)	500
weight of the soft constraint penalty $\hat{c}_i$ of (6.21)	10
state cost γ for Eq. (6.14)	100
weight of the quadratic term of (6.21) ρ	2.5

Table 6.4: Summary of simulation parameters.

6.7.1 Edge energy consumption

The edge energy consumption is evaluated through the processing and migration power, averaged across the MEHs. In Fig. 6.3a, the processing power is shown as a function of the job generation probability p. For all the strategies, a concave increase of processing power is observed for an increasing p. The two heuristic policies perform very similarly, draining between 40 to 200 W of power. On the other hand, `EASE` allows substantial savings, e.g., as much as 80 W at $p = 0.5$ (a gain of 40%). Although two types of servers are used in the simulation, no significant difference is observed regarding the energy consumption induced by the different strategies: the myopic heuristics consume more for both server types. The average power used to migrate the jobs is



(a) $P_{\text{PV}} = 370$ W

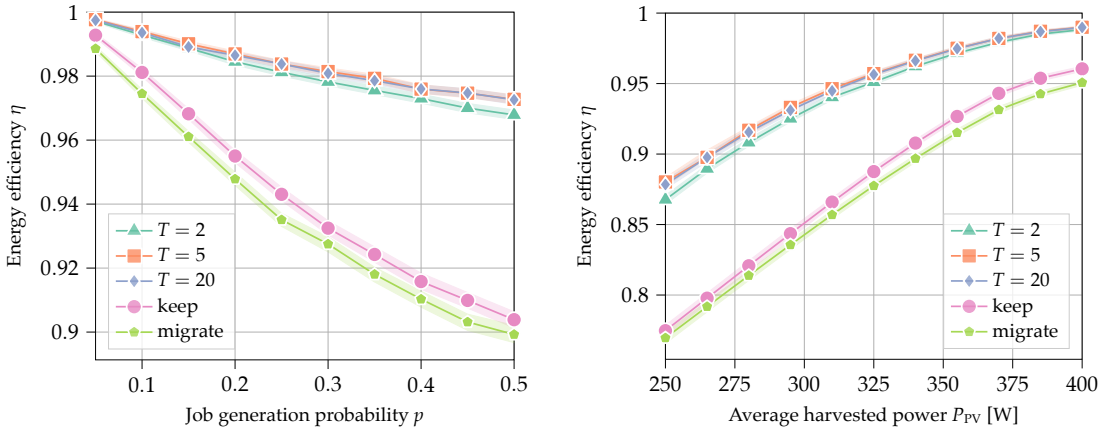
(b) $P_{\text{PV}} = 370$ W

Figure 6.3: Average processing (left) and migration (right) power dissipation of the edge servers.

shown in Fig. 6.3b. Since the "keep" strategy never migrates tasks, its job migration power is always zero. On the other hand, the strategy with the highest migration power is "migrate", while the optimized strategies perform in between. Concerning EASE, both processing and migration power depend slightly on the length T of the prediction horizon. A short prediction window, i.e., $T = 2$, has a higher energy consumption due to an imprecise estimation of future resources. The settings $T = 5$ and $T = 20$ lead to the same processing power performance, whereas $T = 20$ uses more migration power (see for, e.g., $p = 0.2$). This reflects the fact that with $T = 20$ the algorithm better captures the future system evolution, thus migrating the jobs to the next serving eNB at a slightly higher rate.

6.7.2 Energy and job efficiency

To assess the performance of the scheduling approaches, three metrics are used, namely, i) the energy efficiency, defined as $\eta = E_h/E_{\text{tot}}$, i.e., the fraction of harvested (green) energy used over the total energy drained (green plus grid energy), ii) the fraction of executed and finished jobs, and iii) the fraction of jobs finishing in the MEH that is co-located with the eNB serving the vehicle. In Fig. 6.4a, the energy efficiency η is shown as a function of the job generation probability. All the strategies show an almost linear decrease for increasing p. However, the absolute slope of such decrease is larger for "keep" and "migrate" with respect to EASE. At $p = 0.5$, EASE allows gaining about 7% in efficiency: the harvested energy can fully support the edge network for at least 97% of the total energy requirement. At the same time, EASE almost never discards jobs, while for, e.g., $p \geq 0.3$, the "migrate" and "keep" strategies respectively drop 0.5% and 1.5% of the tasks. The fraction of jobs finishing in the MEH co-located with the eNB providing the connectivity to the vehicle remains nearly constant with respect to p. The "migrate" strategy serves around 75% of the jobs in the same eNB/MEH where the vehicle is, while with EASE this only occurs in about 30% of the cases. This depends on the primary goal of the proposed



(a) $P_{PV} = 370$ W

(b) $p = 0.25$

Figure 6.4: Energy efficiency with respect to the job generation probability (left) and the power generated by the PV cells (right).

scheduling pipeline, which corresponds to reducing the carbon footprint of the edge network. As a second-order optimization criterion, and only if feasible, EASE migrates jobs to the next user location (eNB). EASE with $T = 20$ achieves a gain over $T = 5$ of about 2% on this last aspect, while $T = 2$ performs the best, being, however, the least energy efficient. The energy efficiency is also evaluated varying the amount of harvested energy (Fig. 6.4b), with the PV panel generating power in $[P^{PV}_{min}, P^{PV}_{max}]$ W. EASE can entirely sustain the edge at least at 87.5% of the time when the harvested energy is at its minimum, i.e., $P_{PV} = 250$ W, leading to a gain of 10% with respect to the other strategies, resulting in a significantly reduced carbon footprint. At $P_{PV} = 400$ W the gain is lower, but EASE performs very close to complete carbon neutrality ($\approx 99\%$ of efficiency). Note that 400 W are just sufficient to self-sustain (on average) the less powerful HP ProLiant server, but not the Nettrix computing unit at full load.

As a final consideration, from Fig. 6.4 the largest gain is achieved when either the computing demand is high (large p) or the harvested energy is scarce. These are the cases where it is important to use the available resources wisely, and EASE succeeds to do so.

6.7.3 Convergence of the dual ascent

In Fig. 6.5, the convergence speed of the proposed decentralized solution is evaluated. Specifically, the cost value reached at the current iteration is compared with the optimal solution obtained with CVXPY **diamond2016cvxpy**, considering the absolute value of their ratio $|\Gamma(\mathbf{x}^+)/\Gamma(\mathbf{x}^*)|$. In the plot, the 90th percentile is shown, discarding hence 10% of outliers. Thus, whenever the ratio settles down to approximately 1, the nodes have reached the global minimum of the cost function. The results show that the power availability impacts the convergence speed: the more harvested energy P_{PV} is available, the quicker the algorithm reaches the minimum. This descends from the fact that a high energy availability leads to a rare activation of the max term in func-

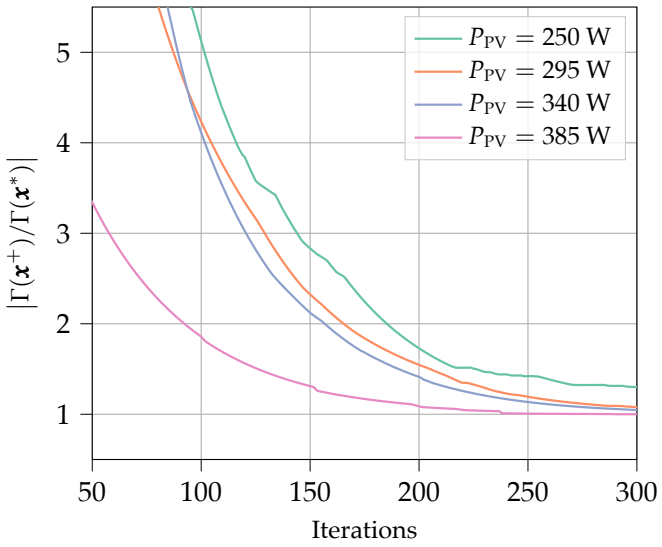


Figure 6.5: Ratio between the value of the cost at iteration m and the optimal cost computed with CVXPY (90th percentile). Job generation probability $p = 0.25$.

tion (6.18). When the max term returns 0 and the constraint (6.19) is not active, the optimum is simply given by $\mathbf{o}_i = \bar{\mathbf{w}}_i$, i.e., the selected action is to follow the vehicle movements. The nodes will be very fast in retrieving this particular solution, as the Lagrange multipliers associated with all the constraints remain null after the first two iterations, leading to accept the solution. A similar reasoning holds for the job generation probability that determines the load of the servers. Here, in the interest of space, we omit the associated plot as it is very similar to Fig. 6.5. Specifically, the convergence requires more iterations as p increases. In fact, an increase in the average load experienced by the servers activates the constraint (6.19), modifying the optimal solution or even activating the penalties $\hat{\delta}_i$. As it is known, the dual ascent is slow when being close to constraint boundaries. However, as a general result, the number of iterations required to converge even with complex initializations is between 200 and 500. The communication overhead can be evaluated considering that two communication rounds (of few bytes) are required per iteration (see Algorithm 6.2). Although this may actually appear to be a high number of exchanged messages, we remark that: i) the subsequent step of the proposed pipeline rounds the solution, and, in turn, it is not necessary to retrieve the exact optimum, but it is sufficient to obtain a decent cost value in the continuous domain; ii) we considered slots of $\tau = 3$ s, which is the amount of time available to make a migration decision. Longer time slots can be used, leaving more time to the decision process.

6.7.4 Rounding algorithm performance

To test the performance of the rounding Algorithm 6.3, the cost function (6.20) is evaluated with the obtained rounded solution $\mathbf{o}^r = \{\mathbf{o}_i^r \mid i \in \mathcal{N}\}$. The comparison is performed with the solution

given by each server i simply following the desired $\bar{\mathbf{w}}_i$, i.e., the solution corresponding to the "migrate" strategy. Specifically, the ratio between the cost values of the "migrate" strategy and the rounded solution is computed, considering the cases where it is energetically inefficient to follow the desired migrations. Indeed, in the other case $\mathbf{o}_i = \mathbf{o}_i^r = \bar{\mathbf{w}}_i$, for all servers, i.e., $\bar{\mathbf{w}}_i$ is the optimal solution and it is a feasible one in the discrete domain, thus the costs are equal. As an example, with prediction horizon $T = 5$, job generation probability $p = 0.3$, and $P_{\text{PV}} = 300$ W, the gain of using the proposed relax and round optimization procedure of EASE over the "migrate" strategy is on average 10 folds. More in the detail, the gain has a median of 3.8, the 10th percentile is 1.3, meaning that rarely a gain lower than 30% is observed, and the 90th percentile is 17. Hence, often, the rounding step of EASE induces a high gain over the blind "migrate" strategy from an energy perspective.

6.8 Concluding remarks

In this chapter, we proposed EASE, a novel strategy for online job scheduling in a MEC-enabled network co-powered by the grid and renewable energy resources, considering an IoV scenario. EASE tackles the problem of ensuring computing service continuity as the users move within the resources-constrained network area. It allows deciding whether to migrate the jobs following the UE, or to continue the execution on the MEC server where it started. This is achieved through the alternation of a local control optimization phase, to estimate future resources, and a distributed consensus step, to reach the migration agreement. The primary objective is the minimization of the carbon footprint at the network side, guaranteeing adequate QoS to the moving users. Using EASE leads to energy efficiency improvements of up to 10% over heuristic strategies, getting close to carbon neutrality in a wide range of contexts.

7 Concluding remarks

Future wireless networks are expected to combine *communication*, *computing* and *sensing* services. The result of their joint implementation is two-fold. At first, the utilization of wireless transmissions as a source of opportunity for environmental sensing will provide the users with monitoring solutions at no additional costs. On the other hand, the integration of environmental information into the network management processes will allow reducing the energy consumption – while guaranteeing adequate quality of experience (QoE) to the users – through a better exploitation of the available resources. The high potential of this integration motivated several studies that brought to the development of advanced algorithms enabling both *communication-assisted sensing* and *sensing-assisted communication* services. Moreover, the new opportunities attracted the attention of the standardization entities – IEEE for Wi-Fi networks and 3GPP for cellular networks – that envision the release of new standards providing support for joint communication and sensing applications. However, several challenges are still open for solutions and require further investigation by the scientific community. With this thesis, we made our contribution by designing novel algorithms that go in the direction of enabling such paradigm change in future indoor and outdoor wireless networks.

In Part I we focused on *communication-assisted sensing* applications. We targeted indoor spaces where Wi-Fi communications are ongoing and we used Wi-Fi signals as a bridge to gain information about the propagation environment. Specifically, we exploited the channel frequency response (CFR) that is continuously estimated at each Wi-Fi receiver to compensate channel impairments and properly decode the received data. We proposed two algorithms with different sensing targets. The former relies on CFR variations to detect the presence of a person inside the monitored area and recognize the movements she/he is performing. The objective of the latter algorithm is instead the identification of the Wi-Fi devices that are transmitting data in range, thus enabling spectrum access monitoring applications.

The *sensing-assisted communication* aspect was tackled in Part II of the thesis. Specifically, we fo-

cused on urban scenarios where cellular networks provide connectivity and computing support to fifth generation (5G) enabled vehicles. We devised two algorithms that use the side information about the vehicles' movements to efficiently allocate computing resources. The algorithms leverage a learning based prediction tool that we designed to track vehicles' mobility in an urban constrained scenario. The proposed resource allocation algorithms are beneficial to both mobile network operators and users in terms of energy saving, better resource usage and increased QoE.

To summarize, we discussed through practical use cases the benefits that *wireless communications* and *sensing applications* grant to each other. This paves the way for their joint integration and exploitation in future *environment aware networks*.

7.1 Future research directions

Despite recent advancements, there are many open issues to be solved to effectively implement joint communication and sensing systems. *First*, the two functions have different requirements in terms of physical channel access. Specifically, the communication functionality requires spectrum resources only when transmitting/receiving data, while sensing can be performed accurately only by providing the device with regular access to the wireless channel, i.e., by sounding the channel at regular time intervals. This requires improving the channel access schemes considering the service – connectivity or sensing – specific needs. *Second*, the increasing number of wirelessly connected devices operating on the licensed and unlicensed portions of the radio spectrum – while implementing different standards – demands for sensing algorithms that are robust to interference from other transmitting sources. Advanced spectrum access policies based on radio fingerprinting of the transmitting devices will be of paramount importance to guarantee fair access to the radio resources. *Third*, existing studies do not consider the joint exploitation of different portions of the radio spectrum – low and high frequency bands – for sensing and communication. In particular, the integration of sensing data at different frequencies is expected to lead to an increased sensing performance, while the functional splitting of communication and sensing on the two portions of the spectrum in an environment-aware manner will allow exploiting the available resources efficiently, avoiding unwanted interferences.

One never notices what has been done.
One can only see what remains to be done.
Marie Skłodowska Curie

References

[1] "Global economic value of Wi-Fi® 2021 – 2025," Wi-Fi alliance, Tech. Rep., 2021.

[2] U. Challita, H. Ryden, and H. Tullberg, "When Machine Learning Meets Wireless Cellular Networks: Deployment, Challenges, and Applications," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 12–18, 2020.

[3] F. Adib and D. Katabi, "See through walls with WiFi!" *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, 2013.

[4] Y. Ma, G. Zhou, and S. Wang, "WiFi sensing with channel state information: A survey," *ACM Computing Surveys*, vol. 52, no. 3, 2019.

[5] C. Wu, F. Zhang, B. Wang, and K. J. Ray Liu, "Mmtrack: Passive multi-person localization using commodity millimeter wave radio," in *Proceedings of IEEE INFOCOM*, 2020.

[6] Y. Ren, J. Lu, A. Beletchi, Y. Huang, I. Karmanov, D. Fontijne, C. Patel, and H. Xu, "Hand gesture recognition using 802.11ad mmwave sensor in the mobile device," in *Proceedings of IEEE WCNCW*, 2021.

[7] F. Gringoli, M. Schulz, J. Link, and M. Hollick, "Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets," in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH)*, Los Cabos, Mexico, Oct. 2019.

[8] W. alliance. "The worldwide network of companies that brings you Wi-Fi®." (), [Online]. Available: `https://www.wi-fi.org`. (accessed: September 18, 2021).

[9] "The economic value of Wi-Fi®: A global view (2021 – 2025)," Wi-Fi alliance, Tech. Rep., 2021.

[10] Federal Communications Commission (FCC), *FCC Opens 6 GHz Band to Wi-Fi and Other Unlicensed Uses*, `https://www.fcc.gov/document/fcc-opens-6-ghz-band-wi-fi-and-other-unlicensed-uses`, 2020.

[11] O. J. of the European Union. "Commission implementing decision (EU) 2021/1067 of 17 june 2021 on the harmonised use of radio spectrum in the 5 945-6 425 MHz frequency band for the implementation of wireless access systems including radio local area networks (WAS/RLANs)." (), [Online]. Available: `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2021:232:FULL&from=EN`. (accessed: September 18, 2021).

[12] I. of Electrical and E. Engineers. "Official IEEE 802.11 working group project timelines - 2021-08-05." (), [Online]. Available: `https://www.ieee802.org/11/Reports/802.11_Timelines.htm`. (accessed: September 18, 2021).

[13] IEEE, "IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN," *IEEE Std 802.11ax-2021 (Amendment to IEEE Std 802.11-2020)*, 2021.

[14] C. Deng, X. Fang, X. Han, X. Wang, L. Yan, R. He, Y. Long, and Y. Guo, "IEEE 802.11be Wi-Fi 7: New challenges and opportunities," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, 2020.

[15] E. Khorov, I. Levitsky, and I. F. Akyildiz, "Current status and directions of IEEE 802.11be, the future Wi-Fi 7," *IEEE Access*, vol. 8, 2020.

[16] IEEE, "IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Enhanced Throughput for Operations in License-exempt Bands above 45 GHz," *IEEE Std 802.11ay-2021 (Amendment to IEEE Std 802.11-2020 as amended by IEEE Std 802.11ax-2021)*, 2021.

[17] G. Naik, J.-M. Park, J. Ashdown, and W. Lehr, "Next generation Wi-Fi and 5G NR-U in the 6 GHz bands: Opportunities and challenges," *IEEE Access*, vol. 8, 2020.

[18] N. Patriciello, S. Lagén, B. Bojović, and L. Giupponi, "NR-U and IEEE 802.11 technologies coexistence in unlicensed mmWave spectrum: Models and evaluation," *IEEE Access*, vol. 8, 2020.

[19] F. Restuccia, "IEEE 802.11bf: Toward ubiquitous Wi-Fi sensing," *arXiv preprint arXiv:2103.14918*, 2021.

[20] B. Ma, W. Guo, and J. Zhang, "A survey of online data-driven proactive 5G network optimisation using machine learning," *IEEE Access*, vol. 8, pp. 35 606–35 637, 2020.

[21] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location aware communications for 5G networks: How location information can improve scalability, latency, and robustness of 5G," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 102–112, Nov. 2014.

[22] M. Koivisto, A. Hakkarainen, M. Costa, P. Kela, K. Leppanen, and M. Valkama, "High-efficiency device positioning and location-aware communications in dense 5G networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 188–195, 2017.

[23] P. Dinh, T. M. Nguyen, S. Sharafeddine, and C. Assi, "Joint location and beamforming design for cooperative UAVs with limited storage capacity," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 8112–8123, 2019.

[24] J. Talvitie, T. Levanen, M. Koivisto, T. Ihalainen, K. Pajukoski, and M. Valkama, "Positioning and location-aware communications for modern railways with 5G new radio," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 24–30, 2019.

[25] M. Chen, Y. Hao, L. Hu, K. Huang, and V. K. N. Lau, "Green and mobility-aware caching in 5G networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8347–8361, 2017.

[26] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service placement for mobile micro-clouds with predicted future cost," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1002–1016, 2017.

[27] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient mobility-aware task offloading for vehicular edge computing networks," *IEEE Access*, vol. 7, pp. 26 652–26 664, 2019.

[28] W. Zhan, C. Luo, G. Min, C. Wang, Q. Zhu, and H. Duan, "Mobility-aware multi-user offloading optimization for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3341–3356, 2020.

[29] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, 2012.

[30] "Requirements related to technical performance for IMT-Advanced radio interface(s)," International telecommunications union (ITU), Tech. Rep., 2008, ITU-R M.2134-0.

[31] "Detailed specifications of the terrestrial radio interfaces of international mobile telecommunications advanced (IMT-advanced)," International telecommunications union (ITU), Tech. Rep., 2019, ITU-R M.2012-4.

[32] "IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond," International telecommunications union (ITU), Tech. Rep., 2015, ITU-R M.2083-0.

[33] "Technical specification group services and system aspects; release 15 description; summary of Rel-15 work items (release 15)," 3GPP, Tech. Rep., Sep. 2019, TR 21.915 V15.0.0.

[34] "Detailed specifications of the terrestrial radio interfaces of international mobile telecommunications 2020 (IMT-2020)," International telecommunications union (ITU), Tech. Rep., 2021, ITU-R M.2150-0.

[35] "Technical specification group services and system aspects; release 16 description; summary of Rel-16 work items (release 16)," 3GPP, Tech. Rep., Sep. 2021, TR 21.916 V16.0.1.

[36] 3GPP, *Advanced plans for 5G*, `https://www.3gpp.org/news-events/2210-advanced_5g`, 2021.

[37] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, "5g evolution: A view on 5g cellular technology beyond 3gpp release 15," *IEEE Access*, vol. 7, pp. 127 639–127 651, 2019.

[38] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 55–61, 2020.

[39] L. Bonati, S. D'Oro, M. Polese, S. Basagni, and T. Melodia, *Intelligence and learning in o-ran for data-driven nextg cellular networks*, 2021. arXiv: 2012.01263 [cs.NI].

[40] A. Garcia-Saavedra and X. Costa-Perez, "O-ran: Disrupting the virtualized ran ecosystem," *IEEE Communications Standards Magazine*, pp. 1–8, 2021.

[41] "Mobile Edge Computing (MEC); Framework and Reference Architecture," ETSI, Tech. Rep., Dec. 2020, GS MEC 003 V2.2.1.

[42] "Multi-access Edge Computing (MEC); MEC 5G Integration," ETSI, Tech. Rep., Oct. 2020, GS MEC 031 V2.1.1.

[43] B. Dong, V. Prakash, F. Feng, and Z. O'Neill, "A review of smart building sensing system for better indoor environment control," *Energy and Buildings*, vol. 199, pp. 29–46, 2019.

[44] Y. Zheng, Y. Zhang, K. Qian, G. Zhang, Y. Liu, C. Wu, and Z. Yang, "Zero-effort cross-domain gesture recognition with Wi-Fi," in *Proceedings of ACM MobiSys*, 2019.

[45] B. Wang, Q. Xu, C. Chen, F. Zhang, and K. J. R. Liu, "The promise of radio analytics: A future paradigm of wireless positioning, tracking, and sensing," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 59–80, 2018.

[46] Y. He, Y. Chen, Y. Hu, and B. Zeng, "WiFi vision: Sensing, recognition, and detection with commodity MIMO-OFDM WiFi," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8296–8317, 2020.

[47] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos, "DeepSense: Device-free human activity recognition via autoencoder long-term recurrent convolutional network," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, 2018.

[48] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, W. Xu, and L. Su, "Towards environment independent device free human activity recognition," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2018.

[49] Z. Shi, J. A. Zhang, Y. D. R. Xu, and Q. Cheng, "Environment-robust device-free human activity recognition with channel-state-information enhancement and one-shot learning," *IEEE Transactions on Mobile Computing*, 2020.

[50] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-Eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, Maui, Hawaii, USA, 2014.

[51] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2015.

[52] J. Ding and Y. Wang, "WiFi CSI-based human activity recognition using deep recurrent neural network," *IEEE Access*, vol. 7, pp. 174 257–174 269, 2019.

[53] N. Damodaran, E. Haruni, M. Kokhkharova, and J. Schäfer, "Device free human activity and fall recognition using WiFi channel state information (CSI)," *CCF Transaction on Pervasive Computing Interaction*, vol. 2, pp. 1–17, 2020.

[54] T. Tegou, A. Papadopoulos, I. Kalamaras, K. Votis, and D. Tzovaras, "Using auditory features for WiFi channel state information activity recognition," *SN computer science*, vol. 1, no. 3, 2020.

[55] J. K. Brinke and N. Meratnia, "Scaling activity recognition using channel state information through convolutional neural networks and transfer learning," in *Proceedings of the First International Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things*, ser. AIChallengeIoT'19, New York, NY, USA: Association for Computing Machinery, 2019.

[56] C. Xiao, D. Han, Y. Ma, and Z. Qin, "CsiGAN: Robust channel state information-based activity recognition with GANs," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 191–10 204, 2019.

[57] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proceedings of the Advances in Neural Information Processing Systems*, Barcelona, Spain, 2016.

[58] Y. Ma, S. Arshad, S. Muniraju, E. Torkildson, E. Rantala, K. Doppler, and G. Zhou, "Location- and person-independent activity recognition with WiFi, deep neural networks, and reinforcement learning," *ACM Trans. Internet Things*, vol. 2, no. 1, 2021.

[59] X. Wang, C. Yang, and S. Mao, "On CSI-based vital sign monitoring using commodity WiFi," *ACM Transactions on Computing for Healthcare*, vol. 1, no. 3, 12:1–12:27, 2020.

[60] Y. Zeng, D. Wu, R. Gao, T. Gu, and D. Zhang, "FullBreathe: Full human respiration detection exploiting complementarity of CSI phase and amplitude of WiFi signals," *Proceedings of ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, 2018.

[61] D. Zhang, Y. Hu, Y. Chen, and B. Zeng, "BreathTrack: Tracking indoor human breath status via commodity WiFi," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3899–3911, 2019.

[62] N. Keerativoranan, A. Haniz, K. Saito, and J. I. Takada, "Mitigation of CSI temporal phase rotation with B2B calibration method for fine-grained motion detection analysis on commodity Wi-Fi devices," *Sensors (Basel)*, vol. 18, no. 11, p. 3795, 2018.

[63] A. Borhani and M. Pätzold, "RF-based human activity recognition: A non-stationary channel model incorporating the impact of phase distortions," in *Proceedings of the International Work-Conference on Artificial Neural Networks*, Gran Canaria, Spain, 2019.

[64] Y. Chen, X. Su, Y. Hu, and B. Zeng, "Residual carrier frequency offset estimation and compensation for commodity WiFi," *IEEE Transactions on Mobile Computing*, vol. 19, no. 12, pp. 2891–2902, 2019.

[65] X. Li, D. Zhang, Q. Lv, J. Xiong, S. Li, Y. Zhang, and H. Mei, "IndoTrack: Device-free indoor human tracking with commodity Wi-Fi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–22, 2017.

[66] K. Qian, C. Wu, Z. Zhou, Y. Zheng, Z. Yang, and Y. Liu, "Inferring motion direction using commodity Wi-Fi for interactive exergames," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Denver Colorado, USA, 2017.

[67] W. Li, R. J. Piechocki, K. Woodbridge, C. Tang, and K. Chetty, "Passive WiFi radar for human sensing using a stand-alone access point," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 1986–1998, 2021.

[68] S. Li, X. Li, Q. Lv, G. Tian, and D. Zhang, "WiFit: Ubiquitous bodyweight exercise monitoring with commodity Wi-Fi device," in *Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, Guangzhou, China, 2018.

[69] M. Muaaz, A. Chelli, A. A. Abdelgawwad, A. C. Mallofré, and M. Pätzold, "WiWeHAR: Multimodal human activity recognition using Wi-Fi and wearable sensing modalities," *IEEE Access*, vol. 8, pp. 164 453–164 470, 2020.

[70] H. Zhu, Y. Zhuo, Q. Liu, and S. Chang, "π-splicer: Perceiving accurate CSI phases with commodity WiFi devices," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2155–2165, 2018.

[71] A. Zubow, P. Gawlowicz, and F. Dressler, "On phase offsets of 802.11ac commodity WiFi," *arXiv preprint arXiv:2005.03755*, 2020.

[72] D. Zhang, Y. Hu, Y. Chen, and B. Zeng, "Calibrating phase offsets for commodity WiFi," *IEEE Systems Journal*, vol. 14, no. 1, pp. 661–664, 2020.

[73] M. Schulz, D. Wegemer, and M. Hollick. "Nexmon: The C-based firmware patching framework." (2017), [Online]. Available: `https://nexmon.org`.

[74] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[75] J. Pegoraro, F. Meneghello, and M. Rossi, "Multiperson continuous tracking and identification from mm-Wave micro-Doppler signatures," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–16, 2020.

[76] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," *CoRR*, 2016. arXiv: 1602.07261.

[77] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 41, no. 1, p. 53, 2011.

[78] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, New York, NY, USA, 2015.

[79] M. Schulz, D. Wegemer, and M. Hollick, *NexMon: A cookbook for firmware modifications on smartphones to enable monitor mode*, arXiv:1601.07077, Dec. 2015.

[80] M. Schulz, J. Link, F. Gringoli, and M. Hollick, "Shadow Wi-Fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over Wi-Fi," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, New York, NY, USA, 2018.

[81] T. Zheng, Z. Sun, and K. Ren, "FID: Function Modeling-based Data-Independent and Channel-Robust Physical-Layer Identification," in *Proc. of IEEE INFOCOM*, 2019.

[82] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a Hybrid RF Fingerprint Extraction and Device Classification Scheme," *IEEE Internet of Things Journal*, vol. 6, no. 1, 2019.

[83] F. Xie, H. Wen, Y. Li, S. Chen, L. Hu, Y. Chen, and H. Song, "Optimized coherent integration-based radio frequency fingerprinting in internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 5, 2018.

[84] Y. Xing, A. Hu, J. Zhang, L. Peng, and G. Li, "On Radio Frequency Fingerprint Identification for DSSS Systems in Low SNR Scenarios," *IEEE Communications Letters*, vol. 22, no. 11, 2018.

[85] K. Sankhe, M. Belgiovine, F. Zhou, S. Riyaz, S. Ioannidis, and K. Chowdhury, "ORACLE: Optimized Radio clAssification through Convolutional neuraL nEtworks," in *Proc. of IEEE INFOCOM*, 2019.

[86] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Fingerprinting Wi-Fi Devices Using Software Defined Radios," in *Proc. of ACM WiSec*, 2016.

[87] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, S. Ioannidis, K. Chowdhury, and T. Melodia, "Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting," in *Proc. of IEEE INFOCOM*, 2020.

[88] E. Johnson, "Physical Limitations on Frequency and Power Parameters of Transistors," in *Proc. of 1958 IRE International Convention Record*, 1966.

[89] E. H. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtilä, "IEEE 802.11ac: Enhancements for very high throughput WLANs," in *Proc. of IEEE PIMRC*, 2011.

[90] E. Khorov, A. Kiryanov, A. Lyakhov, and G. Bianchi, "A tutorial on IEEE 802.11ax high efficiency WLANs," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, 2018.

[91] C. Deng, X. Fang, X. Han, X. Wang, L. Yan, R. He, Y. Long, and Y. Guo, "IEEE 802.11be Wi-Fi 7: New challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, 2020.

[92] IEEE, "IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz," *IEEE Std 802.11ac-2013 (Amendment to IEEE Std 802.11-2012)*, 2013.

[93] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, "Device Fingerprinting to Enhance Wireless Security Using Nonparametric Bayesian Method," in *Proc. of IEEE INFOCOM*, 2011.

[94] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless Device Identification with Radiometric Signatures," in *Proc. of ACM MobiCom*, 2008.

[95] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device Fingerprinting in Wireless Networks: Challenges and Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, 2016.

[96] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms," in *Proc. of ACM MobiHoc*, 2019.

[97] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep Learning Convolutional Neural Networks for Radio Identification," *IEEE Communications Magazine*, vol. 56, no. 9, 2018.

[98] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep Learning for RF Device Fingerprinting in Cognitive Communication Networks," *IEEE JSTSP*, vol. 12, no. 1, 2018.

[99] R. Das, A. Gadre, S. Zhang, S. Kumar, and J. M. Moura, "A Deep Learning Approach to IoT Authentication," in *Proc. of IEEE ICC*, 2018.

[100] S. Gopalakrishnan, M. Cekic, and U. Madhow, "Robust Wireless Fingerprinting via Complex-Valued Neural Networks," *arXiv preprint arXiv:1905.09388*, 2019.

[101] S. D'Oro, F. Restuccia, and T. Melodia, "Can You Fix My Neural Network? Real-Time Adaptive Waveform Synthesis for Resilient Wireless Signal Classification," in *Proc. of IEEE INFOCOM*, 2021.

[102] E. Perahia and R. Stacey, *Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n*. Cambridge Univ. Press, 2008.

[103] A. Orebaugh, G. Ramirez, and J. Beale, *Wireshark & Ethereal network protocol analyzer toolkit*. Elsevier, 2006.

[104] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," *IEEE JSTSP*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[105] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Proc. of ACM NIPS*, 2017.

[106] D. J. ketchen and C. L. Shook, "The application of cluster analysis in strategic management: An analysis and critique," *Strategic Management Journal*, vol. 17, 1996.

[107] IEEE, *TGac Channel Model Addendum. Version 12. IEEE 802.11-09/0308r12*, 2010.

[108] M. Surridge, *5G innovations for new business opportunities*, Feb. 2017.

[109] *Feasibility study on new services and markets technology enablers (Release 14)*, TR 22.891 V14.1.0, 3GPP, Sep. 2016.

[110] *Study on positioning use cases (Release 16)*, TR 22.872 V16.1.0, 3GPP, Sep. 2018.

[111] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of cellular mobile radio localization methods: From 1G to 5G," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1124–1148, Dec. 2018.

[112] E. Björnson, L. Sanguinetti, H. Wymeersch, J. Hoydis, and T. L. Marzetta, "Massive MIMO is a reality - What is next?: Five promising research directions for antenna arrays," *Digital Signal Processing*, vol. 94, pp. 3–20, 2019.

[113] S. Michaelis, N. Piatkowski, and K. Morik, "Predicting next network cell IDs for moving users with discriminative and generative models.," in *Proceedings of the Mobile Data Challenge by Nokia Workshop in Conjunction with International Conference on Pervasive Computing*, Newcastle, UK, Jun. 2012.

[114] A. Hadachi, O. Batrashev, A. Lind, G. Singer, and E. Vainikko, "Cell phone subscribers mobility prediction using enhanced Markov chain algorithm," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Dearborn, MI, USA, Jun. 2014.

[115] A. A. Hasbollah, S. H. Ariffin, N. E. Ghazali, K. M. Yusuf, and H. Morino, "Handover algorithm based VLP using mobility prediction database for vehicular network," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 4, p. 2477, 2018.

[116] D. S. Wickramasuriya, C. A. Perumalla, K. Davaslioglu, and R. D. Gitlin, "Base station prediction and proactive mobility management in virtual cells using recurrent neural networks," in *Proceedings of the 18th IEEE Wireless and Microwave Technology Conference (WAMICON)*, Cocoa Beach, FL, USA, Apr. 2017.

[117] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[118] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014.

[119] M. F. Rahman, W. Liu, S. B. Suhaim, S. Thirumuruganathan, N. Zhang, and G. Das, "Density based clustering over location based services," in *Proceedings of the 33rd IEEE International Conference on Data Engineering (ICDE)*, San Diego, CA, USA, Apr. 2017.

[120] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 2011.

[121] SUMO. "TAPAS Cologne scenario." (2021), [Online]. Available: `https://sumo.dlr.de/docs/Data/Scenarios/TAPASCologne.html`.

[122] J. Kim, H. Chung, G. Noh, B. Hui, I. Kim, Y. Choi, and Y. Han, "Field trial of millimeter-wave-based MHN system for vehicular communications," in *Proceedings of the 12th European Conference on Antennas and Propagation (EuCAP)*, London, UK, Jan. 2018.

[123] J. Ko, Y. Cho, S. Hur, T. Kim, J. Park, A. F. Molisch, K. Haneda, M. Peter, D. Park, and D. Cho, "Millimeter-wave channel measurements and analysis for statistical spatial channel model in in-building and urban environments at 28 GHz," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5853–5868, 2017.

[124] Z. Becvar and P. Mach, "Adaptive hysteresis margin for handover in femtocell networks," in *Proceedings of the 6th International Conference on Wireless and Mobile Communications*, Valencia, Spain, Sep. 2010.

[125] J. Plachy, Z. Becvar, and E. Calvanese Strinati, "Dynamic resource allocation exploiting mobility prediction in mobile edge computing," in *Proceedings of the IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Valencia, Spain, Sep. 2016.

[126] P. A. Frangoudis and A. Ksentini, "Service migration versus service replication in multi-access edge computing," in *Proceedings of the 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Limassol, Cyprus, Jun. 2018.

[127] F. Zhang, G. Liu, B. Zhao, X. Fu, and R. Yahyapour, "Reducing the network overhead of user mobility-induced virtual machine migration in mobile edge computing," *Software: Practice and Experience*, vol. 49, no. 4, pp. 673–693, 2018.

[128] I. Farris, T. Taleb, M. Bagaa, and H. Flick, "Optimizing service replication for mobile delay-sensitive applications in 5g edge network," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Paris, France, May 2017.

[129] V. Di Valerio and F. Lo Presti, "Optimal virtual machines allocation in mobile femto-cloud computing: An MDP approach," in *Proceedings of the IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Istanbul, Turkey, Apr. 2014.

[130] *Handover procedures*, TS 23.009 V12.0.0, 3GPP, Sep. 2014.

[131] Y. Wang, S. Cang, and H. Yu, "A survey on wearable sensor modality centred human activity recognition in health care," *Expert Systems with Applications*, vol. 137, pp. 167–190, 2019.

[132] A. Aissioui, A. Ksentini, A. M. Gueroui, and T. Taleb, "On enabling 5G automotive systems using follow me edge-cloud concept," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5302–5316, 2018.

[133] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018.

[134] "Multi-access Edge Computing (MEC); V2X Information Service API," ETSI, Tech. Rep., Apr. 2020, GS MEC 030 V2.1.1.

[135] M. J. Neely, *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool Publishers, 2010.

[136] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends® in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[137] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[138] M. Zakarya, L. Gillam, A. A. Khan, and I. U. Rahman, "Perficientcloudsim: A tool to simulate large-scale computation in heterogeneous clouds.," *The Journal of Supercomputing*, 2020.

[139] Microsoft. "Sizes for virtual machines in azure." (), [Online]. Available: `https://docs.microsoft.com/en-us/azure/virtual-machines/sizes`. (accessed: 13.10.2020).

[140] I. Labriji, F. Meneghello, D. Cecchinato, S. Sesia, E. Perraud, E. C. Strinati, and M. Rossi, "Mobility aware and dynamic migration of MEC services for the internet of vehicles," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 570–584, 2021.

[141] K. Kaur, S. Garg, G. Kaddoum, S. H. Ahmed, and M. Atiquzzaman, "Keids: Kubernetes-based energy and interference driven scheduler for industrial IoT in edge-cloud ecosystem," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4228–4237, 2020.

[142] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.

[143] M. Feng, M. Krunz, and W. Zhang, "Joint task partitioning and user association for latency minimization in mobile edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8108–8121, 2021.

[144] Z. Rejiba, X. Masip-Bruin, and E. Marín-Tordera, "A survey on mobility-induced service migration in the fog, edge, and related computing paradigms," *ACM Comput. Surv.*, vol. 52, no. 5, Sep. 2019.

[145] T. V. Doan, G. T. Nguyen, H. Salah, S. Pandi, M. Jarschel, R. Pries, and F. H. P. Fitzek, "Containers vs virtual machines: Choosing the right virtualization technology for mobile edge cloud," in *Proceedings of thw IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 46–52.

[146] S. Ramanathan, K. Kondepu, M. Razo, M. Tacca, L. Valcarenghi, and A. Fumagalli, "Live migration of virtual machine and container based mobile core network components: A comprehensive study," *IEEE Access*, vol. 9, pp. 105 082–105 100, 2021.

[147] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, 2018.

[148] L. Ma, S. Yi, N. Carter, and Q. Li, "Efficient live migration of edge services leveraging container layered storage," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2020–2033, 2019.

[149] C. Campolo, A. Iera, A. Molinaro, and G. Ruggeri, "MEC support for 5G-V2X use cases through docker containers," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2019.

[150] A. Rago, G. Piro, G. Boggia, and P. Dini, "Anticipatory allocation of communication and computational resources at the edge using spatio-temporal dynamics of mobile users," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2021.

[151] A. Dalgkitsis, P.-V. Mekikis, A. Antonopoulos, and C. Verikoukis, "Data driven service orchestration for vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4100–4109, 2021.

[152] W. Zhan, C. Luo, G. Min, C. Wang, Q. Zhu, and H. Duan, "Mobility-aware multi-user offloading optimization for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3341–3356, 2020.

[153] Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo, and X. Shen, "A joint service migration and mobility optimization approach for vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9041–9052, 2020.

[154] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Found. Trends Signal Process.*, vol. 11, no. 3-4, 2017.

[155] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proceedings of the 20th international symposium on High performance distributed computing*, 2011, pp. 171–182.

[156] Standard Performance Evaluation Corporation. "SPECpower Results." (2021), [Online]. Available: `https://www.spec.org/power_ssj2008/results/`.

[157] J. B. Rawlings and D. Q. Mayne, *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

List of publications

Journals

[158] G. Perin, F. Meneghello, R. Carli, L. Schenato, and M. Rossi, "Distributed, predictive and energy-aware job scheduling in urban MEC for the IoV," *IEEE Transaction on Vehicular Technology*, 2021, *under submission*.

[159] F. Meneghello, D. Garlisi, N. Dal Fabbro, I. Tinnirello, and M. Rossi, "Environment and person independent activity recognition with a commodity IEEE 802.11ac access point," *IEEE Transaction on Mobile Computing*, 2021, *under submission*.

[160] I. Labriji, F. Meneghello, D. Cecchinato, S. Sesia, E. Perraud, E. C. Strinati, and M. Rossi, "Mobility aware and dynamic migration of mec services for the Internet of Vehicles," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 570–584, 2021.

[161] J. Pegoraro, F. Meneghello, and M. Rossi, "Multiperson continuous tracking and identification from mm-Wave micro-Doppler signatures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 4, pp. 2994–3009, 2020.

[162] F. Meneghello, M. Rossi, and N. Bui, "Smartphone identification via passive traffic fingerprinting: A sequence-to-sequence learning approach," *IEEE Network*, vol. 34, no. 2, pp. 112–120, 2020.

[163] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182–8201, 2019.

Conferences

[164] F. Meneghello, M. Rossi, and F. Restuccia, "DeepCSI: Robust Wi-Fi radio fingerprinting through MU-MIMO CSI feedback deep learning," *submitted - venue not disclosed for double blind review policy*.

[165] F. Meneghello, D. Cecchinato, and M. Rossi, "Mobility prediction via sequential learning for 5G mobile networks," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Thessaloniki, Greece, Oct. 2020.

[166] J. Pegoraro, D. Solimini, F. Matteo, E. Bashirov, F. Meneghello, and M. Rossi, "Deep learning for accurate indoor human tracking with a mm-Wave radar," in *Proceedings of the IEEE Radar Conference (RadarConf20)*, Florence, Italy, Sep. 2020.

Book Chapter

[167] F. Meneghello, G. Perin, and M. Rossi, *"Machine Learning Techniques for Context Extraction and User Profiling in 5G Mobile Systems,"* book chapter in *"CNIT technical report 07- Machine Learning and 5G/6G Networks: Interplay and Synergies"*, S. Barbarossa and A. Zanella, Eds. Texmat, Jul. 2021.

Acknowledgments

On Thursday, September 30, 2021, I submitted these pages, my Ph.D. thesis. I didn't sleep too much the night before that day. I felt excited about what was going to happen. Nothing amazing nor scary – apparently – as I was only required to upload the dissertation on the university website. So what was I excited about? At that moment, I realized that in two days something would have changed. While during the last months of my Ph.D. I was too busy to think about that, once "the day" arrived I could not postpone the thought. I'm still me, Francesca, the young master graduate that decided to enroll in the Ph.D. program, but three years older and much more self-aware. This thesis is the result of intensive years – that allowed me to grow both professionally and personally – and the journey that brought me to this point was not always easy. My supervisor told me that from the beginning. I was quite prepared for days of full satisfaction followed by other days of discouragement. But – more than that – I was very lucky to have shared this adventure with special people who made it unforgettable. Now is the right time to acknowledge all of them for the time they gifted me.

At first, I am really thankful to my supervisor, Prof. Michele Rossi, who was always available when I needed his support. Thanks for the precious advice, the stimulating conversations and for always guiding me thinking at the best for me as a researcher. Thanks for the different opportunities you gave me to collaborate with other research teams around the globe. Finally, thanks for having trusted me as a laboratory assistant for the HDA course for three years, giving me an additional growth opportunity. I couldn't have had a better mentor for my doctoral program.

Next, I would like to thank the other SIGNET faculty members, especially Prof. Andrea Zanella, Prof. Michele Zorzi and Prof. Leonardo Badia, the director of the Department of Information Engineering, Prof. Gaudenzio Meneghesso, and the director of the Ph.D. school, Prof. Andrea Neviani, for all the studying and teaching opportunities they gave me over these years. Another thank goes to the technical and administrative staff of the Department of Information Engineering, for having assisted me with electronics and shipments. Also, thanks to all the people who contributed to making our department a good workplace.

Moving abroad from the department, I thank Prof. Ilenia Tinnirello and Dr. Domenico Garlisi from the University of Palermo, Prof. Francesco Restuccia from Northeastern University, Dr. Joerg Widmer and Alejandro Blanco Pizarro from IMDEA networks, for our collaborations that allowed me to delve into different research topics. Unfortunately, the exchanges have been done remotely only due to the pandemic situation but I am thankful for having have had at least that possibility. I would also extend an additional acknowledgment to Prof. Francesco Restuccia and Dr. Joerg Widmer for having revised this doctoral thesis, providing me with suggestions for the improvement of the presentation of the work.

A big thanks goes to the ICTea team, Matteo Drago, Marco Giordani, Federico Mason, Alberto Signori and Tommaso Zugno. We started as colleagues and we became soon friends, sharing

fantastic adventures and always supporting each other during these years.

Thanks to Silvia Zampato, Marco Canil, Jacopo Pegoraro and Giovanni Perin for being part of the amazing Rossi's team. Thanks to all the other SIGNET - actual and former - members for the good time we spent together (before the pandemic): Martina Capuzzo, Filippo Campagnaro, Davide Cecchinato, Matteo Gadaleta, Roberto Francescon, Mattia Lecci, Davide Magrin, Michele Polese, Mattia Rebato, Paolo Testolina, Davide Zordan, Daniel Zucchetto. Thanks to Laura Crosara for having included me in her project about women in engineering and Elena Camuffo for having been part of it with me. Thanks to Chiara Graziani for the last-minute proofreading of some parts of the thesis. Thanks to the students that worked with me for their thesis: Nicolò Dal Fabbro, Antonio Cusano, Gianmaria Ventura, Quang Minh Pham, Marco Tronu. Thanks to all the Ph.D. students I met during the SSIE summer school at Bressanone and the other events of the Italian telecommunication group – in particular the team from Politecnico of Bari, Arcangela Rago, Vittoria Musa, Paolo Benedetti – for the nice exchanges and the social moments we had together.

Last – but absolutely not least – my family and my friends. A huge thank goes to my parents Fabrizia and Sisto, my siblings Anna and Stefano, my grandmother Maria, my great-aunt Luigina, for always being by my side and for having supported me through these years. Thanks to the rest of the supporting team, Bertilla, Barbara, Sergio, Leonardo, Alessandro, Anna, Annalisa, Elda, Ottorino, Giuseppe, Antonio, Maria Teresa, for always believing in me and for having offered me some relaxing opportunities. Thanks to all my fantastic friends, especially Silvia Bolcato, Manuela Moretto and Francesco Rinaldi – each one of you know how much important you are for me – and Sebastiano Borin, Matteo Calore, Marco Dal Grande, Riccardo Ferrari, Gabriele Lain, Francesco Lanza, Marino Zanconato.

I am grateful for having met you all in my life.

Errata Corrige

Chapter 2.2.3, page 18.

[**Qian2017_2**] refers to publication [K. Qian, C. Wu, Z. Yang, Y. Liu, and K. Jamieson, "Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi," in Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Chennai, India, 2017.]

Chapter 6.2, page 97.

[**perin2021towards**] refers to publication [G. Perin, M. Berno, T. Erseghe, and M. Rossi, "Towards sustainable edge computing through renewable energy resources and online, distributed and predictive scheduling," IEEE Transactions on Network and Service Management, 2021, (early access).]

Chapter 6.6.2, page 107.

[**bertsekas1999nonlinear**] refers to publication [D. P. Bertsekas, Nonlinear Programming, 2nd ed. Athena Scientific, 1999.]

Chapter 6.7.3, page 116.

[**diamond2016cvxpy**] refers to publication [S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," Journal of Machine Learning Research, vol. 17, no. 83, pp. 1–5, 2016.]