

Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN  
INGEGNERIA ELETTRONICA, TELECOMUNICAZIONI E  
TECNOLOGIE DELL'INFORMAZIONE

Ciclo 34

**Settore Concorsuale:** 09/F2 - TELECOMUNICAZIONI

**Settore Scientifico Disciplinare:** ING-INF/03 - TELECOMUNICAZIONI

WIRELESS NETWORK ANALYTICS FOR NEXT GENERATION SPECTRUM  
AWARENESS

**Presentata da:** Enrico Testi

**Coordinatore Dottorato**

Aldo Romani

**Supervisore**

Andrea Giorgetti

**Esame finale anno 2022**

ALMA MATER STUDIORUM  
UNIVERSITY OF BOLOGNA

---

PH.D. PROGRAMME  
ELECTRONICS, TELECOMMUNICATIONS, AND  
INFORMATION TECHNOLOGIES ENGINEERING  
(ETIT)

SSD ING/INF 03  
SC 09/F2 Telecomunicazioni

WIRELESS NETWORK ANALYTICS FOR  
NEXT GENERATION SPECTRUM  
AWARENESS

Ph.D. Thesis

*Ph.D. candidate*  
ENRICO TESTI

*Supervisor*  
Prof. Ing.  
ANDREA GIORGETTI

*Ph.D. coordinator*  
Prof. Ing.  
ALDO ROMANI

*Co-Supervisor*  
Chiar.mo Prof. Ing.  
MARCO CHIANI

---

XXXIV° CYCLE



## **KEYWORDS**

Wireless Network Analytics

Spectrum Awareness

Physical Layer Security

RF Sensing

Machine Learning



*“If I had an hour to solve a problem, I’d spend  
55 minutes thinking about the problem and  
5 minutes thinking about solutions.”*  
A. Einstein



# Acknowledgments

This work was supported by MIUR under the program “Dipartimenti di Eccellenza (2018-2022) - Precise-CPS.”



# Abstract

The importance of networks, in their broad sense, is rapidly and massively growing in modern-day society thanks to unprecedented communication capabilities offered by technology. The next generation wireless networks will face a scenario of ultra-densely connected objects requiring high data-rates, practically 100% reliable quality of service, and extremely low latency. In this context, the radio spectrum will be a primary resource to be preserved and not wasted. Moreover, using artificial intelligence (AI)-based technologies will make the networks more intelligent but, eventually, more vulnerable to malicious users. Therefore, the need for intelligent and automatic systems for in-depth spectrum analysis and monitoring will pave the way for a new set of opportunities and potential challenges.

This thesis proposes a novel framework for automatic spectrum patrolling and the extraction of wireless network analytics. It aims to enhance the physical layer security of next generation wireless networks through the extraction and the analysis of dedicated analytical features. The framework consists of a spectrum sensing phase, carried out by a patrol composed of numerous radio-frequency (RF) sensing devices, followed by the extraction of a set of wireless network analytics. The methodology developed is blind, allowing spectrum sensing and analytics extraction of a network whose key features (i.e., number of nodes, physical layer signals, medium access protocol (MAC) and routing protocols) are unknown. Because of the wireless medium, over-the-air signals captured by the sensors are mixed; therefore, blind source separation (BSS) and measurement association are used to estimate the number of sources and separate the traffic patterns. After the separation, we put together a set of methodologies for extracting useful features of the wireless network. Firstly,

a method for locating uncooperative wireless network nodes using power measurements collected by sensors is proposed. In particular, received signal strength (RSS) is extracted from the separated signals, and localization is performed through conventional least squares (LS) and maximum likelihood estimation (MLE) techniques. Then, the logical topology of the wireless network is inferred. In particular, we detect directed data flows among nodes by identifying causal relationships between the separated transmitted patterns. We compare causal inference methods that use the time series of the separated traffic profiles (i.e., Granger causality (GC), transfer entropy (TE), and conditional transfer entropy (CTE)) with a novel solution based on a neural network (NN) that exploits distilled time-based features. Finally, we propose a machine learning (ML)-based methodology for the classification of application-level traffic patterns of wireless network users.

The whole framework is validated on an ad-hoc wireless network accounting for MAC protocol, packet collisions, nodes mobility, the spatial density of sensors, and channel impairments, such as path-loss, shadowing, and noise. The numerical results obtained by extensive and exhaustive simulations show that the proposed framework is consistent and can achieve the required performance.

Based on the evolution of wireless networks, the importance of security, and ongoing research trends, we will soon witness an increase in interest in the proposed framework.

# Contents

<b>Mathematical Notation</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Spectrum Patrolling . . . . .	5
1.2 Wireless Network Analytics . . . . .	7
1.3 Aims of this Work . . . . .	9
1.3.1 Blind Source Separation . . . . .	9
1.3.2 Transmitters Localization . . . . .	10
1.3.3 Topology inference . . . . .	12
1.3.4 Traffic Classification . . . . .	13
1.4 Document Organization . . . . .	14
<b>2 Scenario and Physical Layer Model</b>	<b>15</b>
2.1 Cases of Interest . . . . .	15
2.2 Scenario . . . . .	16
2.3 Physical Layer Model . . . . .	17
2.3.1 Thermal Noise . . . . .	19
2.3.2 Shadowing . . . . .	20
<b>3 Blind Source Separation</b>	<b>21</b>
3.1 Problem Statement . . . . .	21
3.1.1 Independence and non-Gaussianity . . . . .	23
3.2 Whitening, Node Counting, and Dimensionality Reduction . . . . .	26
3.2.1 Whitening . . . . .	26

---

3.2.2	Node Counting . . . . .	27
3.2.3	Dimensionality Reduction . . . . .	29
3.3	Independent Component Analysis . . . . .	30
3.4	Association Problem . . . . .	31
3.5	Spatial Filtering . . . . .	33
3.6	Excision Filter . . . . .	34
<b>4</b>	<b>Multiple Transmitter Localization</b>	<b>37</b>
4.1	Problem Statement . . . . .	37
4.1.1	Existing Works . . . . .	38
4.2	Received Signal Strength Extraction . . . . .	39
4.3	Localization . . . . .	40
4.3.1	Least Squares . . . . .	41
4.3.2	Maximum Likelihood . . . . .	42
4.3.3	Particle Simulation . . . . .	42
4.4	Cramèr-Rao Lower Bound . . . . .	43
<b>5</b>	<b>Topology Inference</b>	<b>47</b>
5.1	Problem Statement . . . . .	47
5.1.1	Existing works . . . . .	47
5.2	Causality Inference . . . . .	50
5.2.1	Granger Causality . . . . .	50
5.2.2	Transfer Entropy . . . . .	52
5.3	Causality Detection with Machine Learning . . . . .	53
5.3.1	Feature Extraction . . . . .	54
5.3.2	Causality Detection . . . . .	55
<b>6</b>	<b>Traffic Classification</b>	<b>59</b>
6.1	Problem Statement . . . . .	59
6.1.1	Existing Works . . . . .	59
6.2	Features Extraction . . . . .	61
6.3	Survey of ML Classifiers . . . . .	62

---

<b>7</b>	<b>Framework Validation</b>	<b>65</b>
7.1	Simulation Setup . . . . .	65
7.2	Blind Source Separation . . . . .	66
7.2.1	Parameter Setting and Figures of Merit . . . . .	66
7.2.2	Reconstruction Error vs. Position Uncertainty . . . . .	67
7.3	Multiple Transmitter Localization . . . . .	68
7.3.1	Parameter Setting and Figures of Merit . . . . .	68
7.3.2	Number of Sensors and Shadowing . . . . .	70
7.3.3	Computational Complexity Analysis . . . . .	73
7.4	Topology Inference . . . . .	73
7.4.1	Parameter Settings and Figures of Merit . . . . .	73
7.4.2	Topology Inference and Number of Nodes . . . . .	76
7.4.3	Impact of shadowing . . . . .	76
7.4.4	Impact of nodes mobility . . . . .	77
7.4.5	Computational Complexity Analysis . . . . .	78
7.5	Traffic Classification . . . . .	80
7.5.1	Parameter Settings and Figures of Merit . . . . .	80
7.5.2	Accuracy vs. number of RF sensors . . . . .	82
7.5.3	Accuracy vs. observation window . . . . .	83
<b>8</b>	<b>Reinforcement Learning-Based UAVs Patrolling</b>	<b>85</b>
8.1	Problem Statement . . . . .	85
8.2	Multi-Agent Q-Learning . . . . .	87
8.2.1	Update Rule . . . . .	87
8.2.2	Rewards . . . . .	88
8.3	Algorithm Validation . . . . .	89
8.3.1	Rewards . . . . .	90
8.3.2	GDOP . . . . .	91
8.3.3	Localization Error . . . . .	91
8.3.4	Conclusions . . . . .	92
<b>9</b>	<b>Conclusions</b>	<b>95</b>
	<b>List of Tables</b>	<b>99</b>

List of Figures	101
Acronyms	107
Bibliography	111

# Mathematical Notation

Throughout the thesis, capital boldface letters denote matrices, lowercase bold letters denote vectors,  $(\cdot)^\top$  stands for transposition,  $(\cdot)^{-1}$  indicates the inverse operator,  $\text{trace}(\cdot)$  is the trace operator,  $\|\cdot\|_p$  is the  $l_p$ -norm,  $|\cdot|$  is the module operator,  $\odot$  stands for the element-wise product. With  $v_{i,j}$ ,  $\mathbf{v}_{i,:}$ , and  $\mathbf{v}_{:,j}$ , we represent, respectively, the element, the  $i$ th row, and the  $j$ th column of the matrix  $\mathbf{V}$  (when unambiguous, the  $i$ th row of  $\mathbf{V}$  is  $\mathbf{v}_i$ ), and with  $\mathbf{v}_{i,j:k}$  we select the elements between the  $j$ th and the  $k$ th entry of the  $i$ th row of  $\mathbf{V}$ , extremes included.  $\mathbf{I}_N$  indicates the  $N \times N$  identity matrix. We use  $\mathcal{N}(\mu, \sigma^2)$  to denote a real Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ ,  $\mathcal{U}(a, b)$  to denote a uniform distribution between  $a$  and  $b$ ,  $\mathbb{E}\{\cdot\}$  to denote the expectation operator,  $\langle \cdot \rangle$  to indicate the sample mean operator and  $\mathbb{1}_{\{\mathcal{A}\}}$  is the indicator function equal to one when  $\mathcal{A}$  is true and zero otherwise.



# Chapter 1

## Introduction

In future 6G communication systems, billions of wireless devices (sensors, connected vehicles, smart infrastructures, etc.) will overwhelmingly demand radio spectrum resources to ensure high communication reliability. Consequently, strict regulation and real-time monitoring of radio spectrum usage will be of paramount importance. For this reason, a considerable research effort is trying to exploit such a precious resource to push technology towards the limits of communication to densify networks, increase bandwidth, reduce latency, and increase reliability [1, 2].

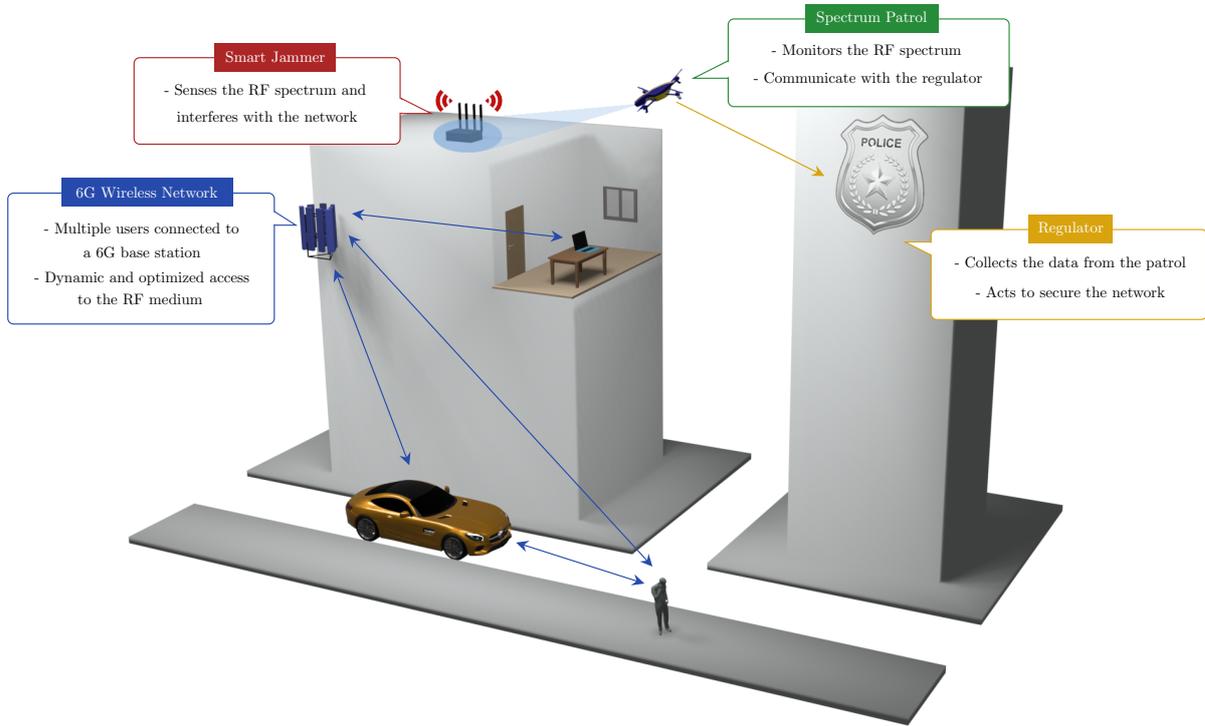
Next generation wireless networks will have to provide extremely high data rates and radically new applications, which require a new wireless radio technology paradigm. As reported by IBM, by 2025, an estimated 75 billion connected internet of things (IoT) devices are expected to generate over 600 zettabytes (or 600 trillion of bytes) of data [3, 4]. Current communication systems, designed on conventional communication theories, significantly limit further performance improvements and lead to severe limitations. Presumably, 6G will continue to benefit from the technological advancements of 5G, but new strategies will undoubtedly be needed to make the next step-change. Compliance with these needs requires intelligent adaptive learning and decision making devices. Future smart 6G mobile terminals are expected to autonomously access the free portions of the spectrum with the aid of sophisticated spectral efficiency learning and inference tools to enhance the

quality of the communications.

Thanks to such remarkable advances, 6G systems will become the nervous systems of our society.

However, it seems that a small effort has been made to ensure the resiliency of such a nervous system. This aspect is even more exacerbated by the upcoming revolution of artificial intelligence (AI), making our devices smart and efficient on one side but much more vulnerable on the other [5–9]. Such vulnerabilities can lead to different forms of unauthorized uses, such as lower-tier devices accessing spectrum reserved for higher-tier devices, unauthorized devices accessing licensed spectrum using software radios, jammers, or denial of service attacks. Techniques must be developed to detect such unauthorized access. In this sense, large-scale spectrum patrolling is emerging as a critical aspect for detecting improper spectrum usage in a technology-dependent society. In this context, more in-depth knowledge of how a network uses the wireless medium and, thus, the network’s structure, may contribute to developing a much more effective spectrum monitoring [10].

In 6G, cognitive radio (CR) will unlock its full potential, enabling an automated society of heterogeneous mobile systems and networks [11,12]. However, this paradigm raises security problems, e.g., jamming, because malicious secondary users (SUs) can sense the spectrum and interfere with different signals at different time instants on different channels. In fact, considering the jamming problem, it is often assumed that attackers adopt a strategy that does not change with time. Suppose the jammers are also equipped with CR technology. In that case, it is highly predictable that such outlaws may adapt their attack strategy according to the dynamics of the environment as well as to the strategy of the legitimate SUs. For this reason, it is expected that 5G and 6G technologies will cause the birth of a new category of smart jammers, enhanced with AI, that will be able to learn the mechanisms of the wireless networks to devise and deploy advanced jamming tactics [13–20].

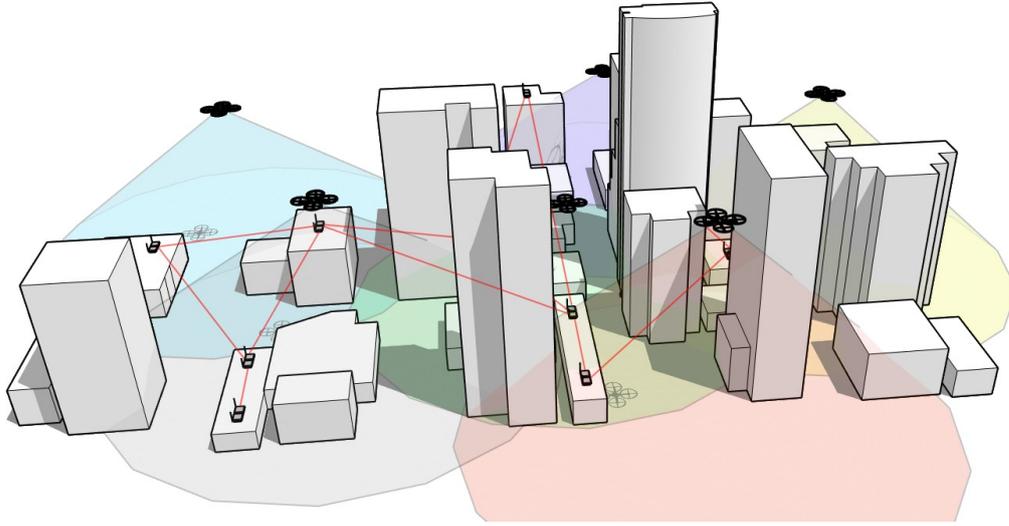


**Figure 1.1:** An illustration of a next generation wireless network interfered by a smart jammer. A spectrum patrol, e.g., an unmanned aerial vehicle (UAV), senses the radio-frequency (RF) medium and sends the data to the authority or regulator. The regulator extracts a set of analytics from the data to acquire as much information as possible about the scenario and eventually detect the jammer.

## 1.1 Spectrum Patrolling

A mechanism of spectrum patrolling can be adopted to enhance the security of next generation CR technology [21]. Fig. 1.1 depicts a wireless network scenario in which a spectrum patrol, represented by a UAV, collects the data about the RF scene. Such data is then forwarded to the authority that extracts a set of wireless network analytics, acting as a fusion center. The analytics are then used, e.g., to detect the anomalies and the malicious users (e.g., the jammer) that are interfering with the network.

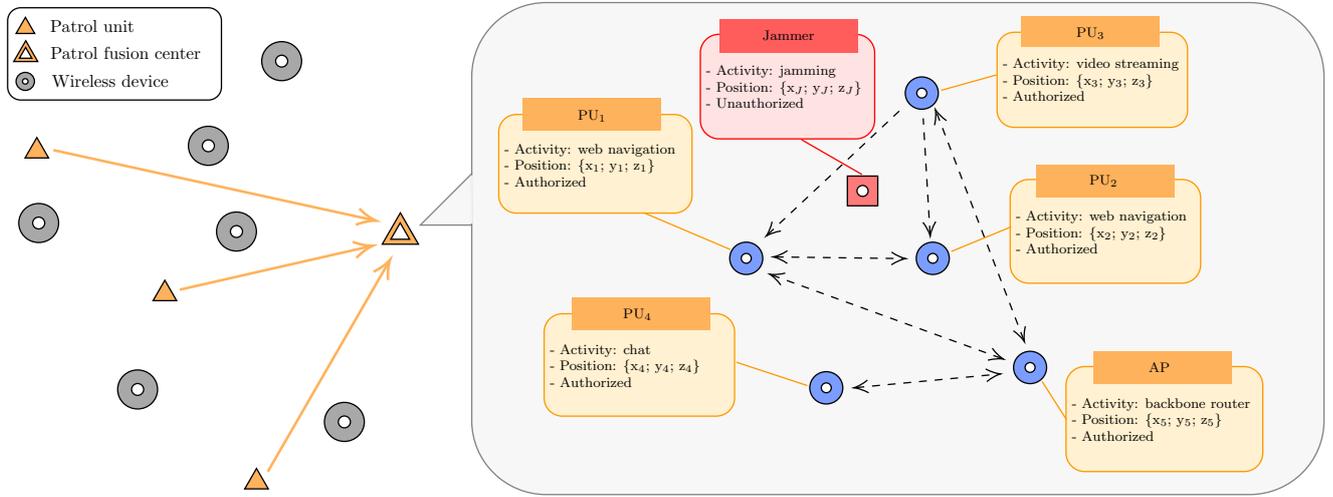
The patrol can be either a dedicated device or a network provided by the authority (as depicted in Fig. 1.2) or a crowdsourcing-based system. The users periodically sense the spectrum and send distilled data to operators or the regulator [21]. Such data can be processed to extract a set of analytics



**Figure 1.2:** An example of urban scenario where a cloud of RF sensors aboard UAVs monitors a wireless network to extract analytics. Alternatively, sensors could be deployed in fixed positions within the network landscape.

that provide the authority with a complete picture of the RF scene. This enables the regulator to detect the presence of anomalies or malicious users in the observed network and act to secure it.

While we require communications with almost 100% reliability, our shared resource, the spectrum, is still monitored by regulators with lab-grade spectrum analyzers whose huge cost and thirst for energy make spectrum monitoring ineffective, slow, outdated, sporadic and extremely sensitive to the scalability problem. We believe that everything around us will be amazingly intelligent in the next decade, giving rise to what is conceptualized as internet of intelligent things (IoIT) [2, 22]. The high number of intelligent devices, enhanced by AI capabilities, will act as an extensive low-cost heterogeneous spectrum sensors network. Accordingly, large-scale crowdsourcing paradigms will lead to a revolution in spectrum sensing and monitoring, opening the way to a new branch of distributed crowdsourcing-based methodologies. This revolution will enormously increase the effectiveness of spectrum patrolling across the environment, making this tool a precious resource [21].



**Figure 1.3:** An example of a spectrum patrolling scenario composed of the nodes of a wireless network, many patrol units equipped with RF sensors, and a fusion center. The patrol units collect the over-the-air power profiles transmitted by the nodes and send them to the fusion center. The fusion center, which can also be an authority (or regulator) extracts the wireless network analytics and gets a detailed description of the analyzed network. Such information can be used to secure the network from malicious users (i.e., jammers) and optimize spectrum usage.

## 1.2 Wireless Network Analytics

The extrapolation and the collection of non-trivial features of the wireless network, from the physical to the application levels, will be crucial in detecting anomalies (e.g., jammers and unauthorized spectrum utilizers), optimizing the communications and reusing the spectrum [23,24]. For example, Fig. 1.3 shows how a patrol composed of a set of RF sensors transmits all the acquired data to a fusion center in charge of merging and processing all the observations. The fusion center extracts the wireless network analytics and gets a detailed, comprehensive view of the observed network. Such information can enhance wireless network's security and optimize spectrum usage. In this section, we would like to propose a set of characteristics that will hopefully allow the patrol to orchestrate and protect a next generation wireless network and its users.

**Multiple Transmitter Localization** The first crucial information that comes to mind when thinking about a patrol monitoring a network is the

position of its nodes. Considering the imposing will of security and privacy preservation, it is strongly believed that multiple non-collaborative transmitter localization will gain attention in the upcoming years. This analytic allows the authority to localize the possible sources of interference. Once all the other analytics are extracted, the complete knowledge about the observed network can be used to identify the malicious users and remove them.

**Topology Inference** How can the patrol protect the wireless network without knowing its logical structure? Indeed, the possibility of reconstructing a network's structure from a few observed quantities at some nodes (or at the edges) with little, if not zero, prior knowledge, is one tricky task that will enhance the capability of spectrum patrolling. If the problem appears rather complicated for a wired network, it can be even more challenging in a wireless scenario because of interference, path-loss, shadowing, fading, and the so-called hidden terminal problem. While connectivity between the nodes could be inferred based upon the distance between them, since many nodes can be within the range of others, guessing which ones are communicating might better rely on their activity patterns.

**Traffic Classification** On large-scale network infrastructures, identifying malicious use of resources and orchestrating security operations such as fire-walling and filtering unwanted traffic is paramount. In such a scenario, in-depth knowledge of the composition of traffic and the identification of trends in application usage may help CRs improve network design and provisioning. It is easy to think that the patrol should be provided with a smart control unit that classifies the activity of the nodes to detect anomalies within the monitored network. These activities can span across the whole ISO-OSI stack, from the physical access to the channel to the application level. Understanding the type of traffic generated by the users of a wireless network is crucial for identifying malicious transmitters. Hence, the traffic profiles generated by a jammer can be spotted thanks to the recognition of an unusual transmission pattern.

## 1.3 Aims of this Work

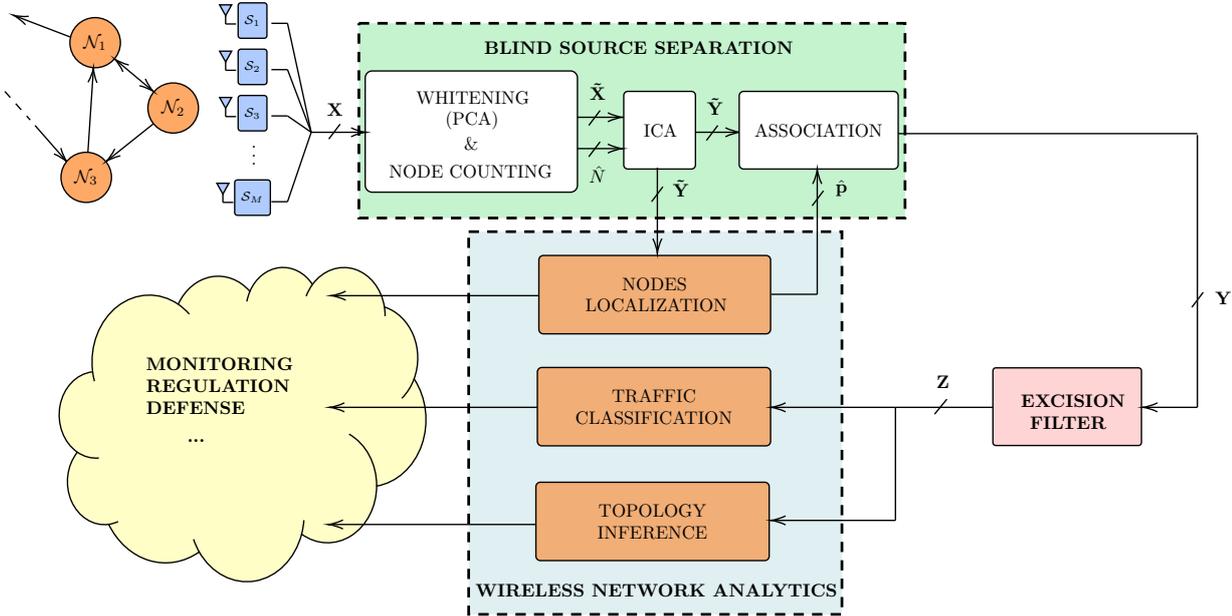
This work proposes a novel framework for extracting a set of analytics from a target wireless network in stealth mode by observing over-the-air spatial and temporal spectrum usage through RF sensors (i.e., the spectrum patrol). The methodology developed is blind, allowing the analysis of a network whose key features (i.e., number of nodes, physical layer signals, and medium access protocol (MAC) and routing protocols) are unknown. Fig. 1.4 shows the complete logical structure of the proposed framework. Because of the wireless medium, over-the-air signals captured by the sensors are mixed; therefore, blind source separation (BSS) and measurement association are used to separate traffic patterns. Then, the separated traffic profiles are used to extract the network analytics. The main stages of the research carried out during the Ph.D. period are briefly described in the following.

### 1.3.1 Blind Source Separation

Each RF sensor that composes the patrol collects a mixture of over-the-air received powers from all the nodes of the observed network. All the collected mixtures are sent to a fusion center, which can be either part of the patrol or the authority supervising the area. The set of acquired mixtures is processed to separate the traffic profiles generated by each wireless network node. This reconstruction phase lays the foundation for the extraction procedure of the analytics.

#### Research Questions

A signal separation stage based on BSS, which involves fast independent component analysis (F-ICA) and principal component analysis (PCA), has been proposed. As depicted in Fig. 1.4, the procedure consists of estimating the number of unknown sources, a preliminary signal conditioning, and the unmixing of the traffic profiles. In this sense, a novel and efficient solution to the permutation problem has been proposed. Such a solution requires a coarse estimate of the position of the network node. Therefore, multiple



**Figure 1.4:** A logical block schematic representation of the proposed framework. The spectrum patrol collects the over-the-air traffic profiles generated by the wireless network nodes. Then, BSS is performed to unmix the received signals. The last stage is the extraction of network analytics.

transmitter localization can be performed before the measurement association. The research on BSS tackles the following research questions.

- *Q1 - Is it possible to extract information about a network via RF sensors that collect only received powers?*
- *Q2 - Is it possible to estimate the number of transmitters?*
- *Q3 - Is it possible to separate over-the-air signals to get the transmitted traffic profiles of each node?*

### 1.3.2 Transmitters Localization

There have been several fundamental contributions on localization over the past decade: numerous algorithms have been developed and tested exploiting different wireless signals, such as WiFi and ultra wideband (UWB), the problem of multi-target localization using groups of cooperating sensors has

also been widely investigated, also for scalable and distributed localization algorithms. However, most of the works found in the literature assume that the targets can exchange information with the anchors, or patrol, thus contributing to the localization process. Only a few works tackle the localization problem without active interaction between the targets and the sensors. A minimal subset of them considers the most common channel impairments (i.e., shadowing, multiple paths) in the localization process.

### Research Questions

After the unmixing, the received signal strength (RSS)-based localization of multiple unknown targets is performed. The innovative proposed methodology is based on a filtering procedure for the extraction of the RSS from the unmixed sequences and maximum likelihood estimation (MLE)-based localization. Considering that this particular problem has not been widely investigated yet, the proposed solution has been compared with the only alternative found in the literature. Moreover, an expression of the Cramèr-Rao lower bound (CRLB) for the specific problem has been derived analytically and used as a benchmark to evaluate the algorithm's performance. The transmitter localization part tackles the following research questions.

- *Q4 - Is the RSS-based localization of multiple unknown transmitters possible?*
- *Q5 - Is the proposed approach better than the current state-of-the-art?*
- *Q6 - How many sensors are needed to perform the localization?*
- *Q7 - How many transmitters can be localized simultaneously?*
- *Q8 - What is the impact of channel impairments on localization performance?*

The non-collaborative transmitter localization problem is approached from a different point of view in Chapter 8. In that case, the objective is to propose an automatic navigation system for a spectrum patrol composed of a fleet of

UAVs. The proposed solution allows the fleet to navigate a scenario and find the best spatial configuration to localize a wireless transmitter. The patrol can use this system in synergy with the network analytics tool to localize a malicious user (i.e., jammer) and secure a wireless network.

### 1.3.3 Topology inference

Recently, the topic of wireless network topology inference has gained a lot of prominence due to the evolution of the networks toward an effective CR paradigm. Therefore, various approaches are proposed in the literature to extract the topology of a network observing only the power profiles transmitted (and received) by its nodes. In this sense, some researchers are exploiting the concept of causality, with all its philosophical background, to make a logical association between network nodes and infer its topology. There is vast literature in the field of causality, a topic that was not born with any connection with the world of telecommunications. Since the concept of causality has a philosophical origin, there is not only one correct analytical model to express it. At the moment, many researchers are diving into the numerous possible model formulations for detecting causality between two events, looking for a methodology that can help infer the topology of a wireless network. This is currently an open topic and it will certainly be widely investigated in the upcoming years.

#### Research Questions

The second analytic that is extracted is the wireless network topology. An innovative technique for the topology inference that exploits the concept of causality and is based on machine learning (ML) has been proposed. In particular, this technique involves a feature extraction phase in which statistical features related to the inter-arrival time of the packets transmitted by the network nodes are obtained from the unmixed data. Then, a binary classifier is properly trained and used to detect the presence of causality between the traffic profiles generated by the nodes. This allows the identification of the possible logical links between network nodes. The topology inference part

tackles the following research questions.

- *Q9 - Can the logical topology of a wireless network be inferred from external?*
- *Q10 - Can the causality inference approach be improved?*
- *Q11 - How many sensors are needed to reach a prescribed performance?*
- *Q12 - What is the impact of channel impairments on topology inference?*

### 1.3.4 Traffic Classification

While traffic classification in wired networks has been extensively investigated, very few works address the problem in wireless systems, although the emergence of CR technology makes this aspect rather important. More in-depth knowledge of how a network uses the wireless medium and, thus, classifying its users' activities may contribute to the development of effective spectrum sharing strategies. In this context, it is desirable to automatically recognize the user-level application generating a given stream of packets from direct observation of the RF scene [25, 26]. The impact of AI in such a particular and delicate task has to be considered. Recently, new ML-based approaches for traffic classification in non-collaborative wireless networks using low-cost RF sensors have been proposed. In these works, the authors were able to classify different types of applications, i.e., streaming, chat, browsing. This race to the use of AI is dictated by the fact that this particular problem can be very easily formulated in a very suitable way for the ML methodologies. The ferment of the ML community in this area is already making itself heard loudly, and this trend will only grow in the near future.

#### Research Questions

The last information extracted about the network is the type of traffic generated by the nodes. In fact, a novel ML-based methodology for the classification of the application-level traffic generated by the nodes of a wireless

network has been proposed. After extracting distilled time-based features from the unmixed data, a multi-class classification is performed. In particular, the algorithms can distinguish between three types of traffic: video streaming, web navigation, and chat. In this regard, in the numerical results, four well-known ML classifiers have been tested and compared: PCA, kernel principal component analysis (KPCA), support vector machine (SVM), and neural network (NN). The traffic classification part tackles the following research questions.

- *Q13 - Can the application-level traffic generated by the nodes of a wireless network be classified from external?*
- *Q14 - What is the impact of channel impairments on traffic classification?*
- *Q15 - How long do we need to observe the network to perform the classification?*
- *Q16 - How many sensors are needed to reach a prescribed performance?*

## 1.4 Document Organization

This document is organized as follows. Chapter 2 introduces the scenario and the system model. In particular, the propagation issues and channel impairments of a realistic scenario are modeled. Chapter 3 provides an overview of the blind source separation problem and describes the proposed solutions. In Chapter 4, the multiple transmitter localization problem is tackled. Chapter 5 introduces the topology inference problem and details the proposed ML-based solution. The traffic classification problem is tackled in Chapter 6. Chapter 7 provides the validation of the proposed framework through extensive numerical results. In Chapter 8, a navigation system for a spectrum patrol composed of a fleet of UAVs is studied. Finally, conclusions are drawn in Chapter 9, followed by the complete list of publications made during this Ph.D.

# Chapter 2

## Scenario and Physical Layer Model

### 2.1 Cases of Interest

This section identifies three cases of interest that help us underline the importance of spectrum patrol and network analytics extraction. In particular, we sketch three scenarios in which the adoption of a spectrum patroller enhances networks security.

#### Cognitive Radio Networks

In a general CR scenario, a primary wireless network might wish to know if another network uses the same spectrum (legitimately or not). In this case, the wireless nodes can schedule a periodic sensing phase to sense the RF medium [27–29]. Once the primary network collects the RF spectrum samples, it can detect an eventual unknown adversarial network and extract some key information about it. Once this information about the adversarial network is extracted, the primary network can make decisions about the spectrum usage, perform communication optimization based on the adversarial network’s behavior, or notify spectrum regulators about violations by non-legitimate communications. In this scenario, the time spent to sense the spectrum reduces the primary network’s throughput while gaining insight

into the spectrum usage.

### Defense Scenario

In a defense scenario, a network of RF sensors can be deployed in an unknown environment to detect and extract information about an adversarial network's structure. The sensors can collect over-the-air received power profiles and perform an in-depth analysis of the adversarial network without the need to be part of it. The extraction of key information about the adversarial network might help, e.g., detecting and studying an adversarial tactical network that exchanges data between soldiers. Such operation enhances the security of the soldiers and might prevent them from taking potentially fatal risks.

### Industrial Security

In an industrial scenario, a set of machines (i.e., an assembly line) communicating and synchronizing their tasks through a wireless connection might be highly vulnerable to jamming attacks. A malicious transmitter might disrupt the inter-machine communications and slow down or stop the production process, causing inconveniences and substantial money losses.

A spectrum patrol, deployed in the production environment, can sense the spectrum and search for the jammer to secure the communications.

## 2.2 Scenario

Let us consider a scenario, depicted in Fig. 2.1, with a non-collaborative wireless network of  $N$  nodes (the network in the following) and a network of  $M$  RF sensors (the sensors in the following) randomly deployed on a two-dimensional landscape. Without loss of generality, the numerical results are derived considering the position  $(\tilde{x}_m, \tilde{y}_m)$  of the  $m$ th sensor uniformly distributed, i.e.,

$$\tilde{x}_m, \tilde{y}_m \sim \mathcal{U}(-L/2, L/2), \quad (2.1)$$

where  $L$  is the side length of a squared landscape. We assume that the technical specifications of the network (i.e., number of nodes, physical layer signals, and MAC and routing protocols) are unknown.

Since no interaction is expected between the observed network and the sensors, all the subsequent tasks are performed without demodulating the received signals, so that a simple energy detector (ED) receiver suffices [28, 30, 31].

Each sensor can measure only the instantaneous received power over short time intervals and send such information to a fusion center that performs inference and network analytics. In this setting, the goal is to design a framework of automatic network analysis tools that exploit only features observable by the temporal evolution of the packets flow between the devices.

## 2.3 Physical Layer Model

Let us consider the equivalent low-pass representation of the signal received by the  $m$ th sensor

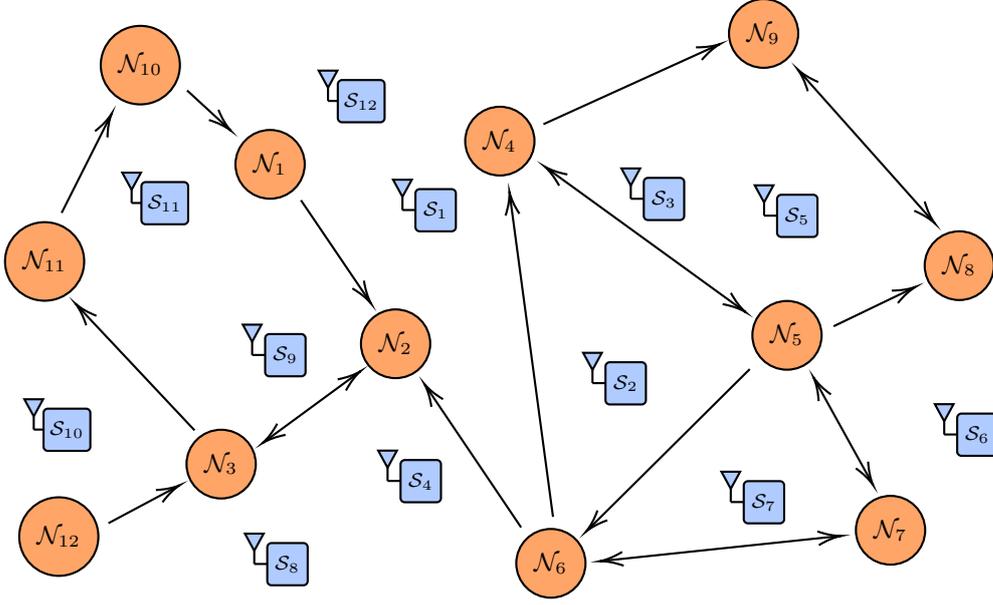
$$r_m(t) = \sum_{n=1}^N q_n(t)g_{m,n} + \nu_m(t) \quad (2.2)$$

where  $q_n(t)$  is the signal transmitted by node  $n$ ,  $g_{m,n}$  is the channel gain between node  $n$  and sensor  $m$  due to path-loss (which in turn depends on the distance  $d_{m,n}$  between the two and line-of-sight (LOS)/non-line-of-sight (NLOS) condition), antenna gains at the nodes and the sensors, and the carrier frequency, and  $\nu_m(t)$  is the additive white Gaussian noise (AWGN) with independent, identically distributed (i.i.d.) real and imaginary parts, each with two-sided power spectral density  $N_0$ .

For the sake of extracting the wireless network analytics, we seek to collect the transmitted power profiles for each node. Let us define such profile at node  $n$  as a vector of  $K$  samples  $\mathbf{p}_n = (p_{n,1}, p_{n,2}, \dots, p_{n,K})^T$ , whose elements

$$p_{n,k} = \frac{1}{T_b} \int_{(k-1)T_b}^{kT_b} |q_n(t)|^2 dt \quad k = 1, \dots, K \quad (2.3)$$

correspond to the transmitted power calculated over short intervals of du-



**Figure 2.1:** A cloud of randomly distributed RF sensors (in blue) across the wireless network landscape (in orange).

ration  $T_b$  such that  $T_{ob} = KT_b$ . Then, let us collect the transmitted power profiles in the matrix

$$\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)^T \in \mathbb{R}^{N \times K}. \quad (2.4)$$

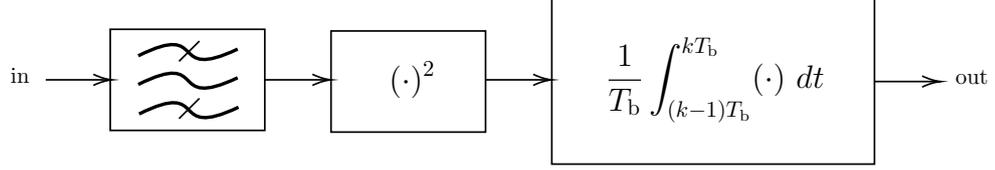
Similarly, the output of the RF sensor  $m$  is a vector,

$$\mathbf{x}_m = (x_{m,1}, x_{m,2}, \dots, x_{m,K})^T, \quad (2.5)$$

whose samples correspond to the received power within  $T_b$ , i.e.,

$$x_{m,k} = \frac{1}{T_b} \int_{(k-1)T_b}^{kT_b} |r_m(t)|^2 dt \quad k = 1, \dots, K. \quad (2.6)$$

The samples (2.6) are obtained by an ED, as depicted in Fig. 2.2, composed of a bandpass zonal filter with bandwidth  $W$  (the same of the transmitted signals), followed by a square-law device and an integrator with finite integration time  $T_b$  [28, 30]. Collecting the measured powers of all sensors in the



**Figure 2.2:** An illustration of the ED: bandpass zonal filter followed by a square-law device and finite time integrator.

matrix

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M)^T \in \mathbb{R}^{M \times K},$$

it is possible to relate  $\mathbf{X}$  with  $\mathbf{P}$  by

$$\mathbf{X} = \mathbf{HP} + \mathbf{N}, \quad (2.7)$$

where  $\mathbf{H} \in \mathbb{R}^{M \times N}$  is the matrix of power gains  $h_{m,n} = |g_{m,n}|^2$ , and  $\mathbf{N} \in \mathbb{R}^{M \times K}$  is the matrix of noise power samples at the output of the ED.

In deriving (2.7) we consider that the signals emitted by the network nodes are mutually uncorrelated and uncorrelated with the noise.

Note that matrix  $\mathbf{X}$  contains the traffic patterns of the wireless network, i.e., the matrix  $\mathbf{P}$  implicitly. However, extracting the desired packet flows is quite challenging because of multiple access interference, packet collisions, and physical layer impairments such as propagation and noise.

### 2.3.1 Thermal Noise

The equivalent low-pass representation of the signal in (2.2) contains the AWGN with i.i.d. real and imaginary parts, each with two-sided power spectral density  $N_0$ . If we consider the noise power samples at the output of the ED we have

$$n_{m,k} = \frac{1}{T_b} \int_{(k-1)T_b}^{kT_b} |\nu_m(t)|^2 dt, \quad (2.8)$$

where  $n_{m,k}$  is an element of  $\mathbf{N} \in \mathbb{R}^{M \times K}$ . Thus,  $n_{m,k}$  is a sum of independent squared standard normal random variables (r.v.s). Therefore, in general, the

elements of  $\mathbf{N}$  are i.i.d. central chi-squared r.v.s with a number of degrees of freedom  $N_{\text{DOF}} = 2WT_b$  [30]. The intervals duration,  $T_b$ , is chosen such that  $N_{\text{DOF}} = 2WT_b$  is an integer. It has been proved that, if  $N_{\text{DOF}}$  is large enough, by the central limit theorem, the chi-squared distribution can be approximated with a normal random variable. Thus, we can approximate the distribution of the elements of  $\mathbf{N}$  as

$$n_{m,k} \sim \mathcal{N}(\sigma_N^2, 2\sigma_N^4/N_{\text{DOF}}), \quad (2.9)$$

where  $\sigma_N^2 = 2N_0W$  [28, 30].

### 2.3.2 Shadowing

To properly assess the performance of the proposed frameworks we believe that shadowing has to be taken into account. In fact, in a real scenario, the impact of obstructions in the propagation environment might strongly affect the proposed methodology's capability of extracting wireless network analytics. The channel gain consists of two components

$$h_{m,n} = h'_{m,n} e^{\sigma_S s_{m,n}}, \quad (2.10)$$

where  $h'_{m,n}$  is the path gain, and  $s_{m,n} \sim \mathcal{N}(0, 1)$  are i.i.d. Gaussian r.v.s to model log-normal shadowing with intensity  $\sigma_S$  [32].

The shadowing parameter is usually expressed as the standard deviation of the channel loss in deciBel by

$$\sigma_S(\text{dB}) = \frac{10}{\ln 10} \sigma_S. \quad (2.11)$$

In Chapter 7, the impact of shadowing on the performance of the proposed algorithms is evaluated. In particular, the novel methodologies are tested both in strong and mild shadowing regimes and compared to the solutions found in the literature.

# Chapter 3

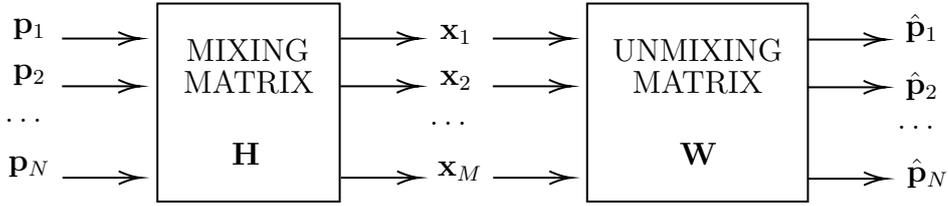
## Blind Source Separation

The analytics extraction requires the temporal characterization of the transmitted packets for each node of the wireless network. Therefore, a reconstruction of the temporal traffic profiles,  $\mathbf{P}$ , as if they were measured at each node, is needed. However, because of the wireless medium, sensors observe a mixture of the signals emitted by the nodes, (2.7), and an unmixing operation is required to extract  $\mathbf{P}$  [33,34].

In literature, various methods for separating mixed signals have been proposed, e.g., matrix factorization [35] and tensor decomposition [36,37], to name a few. In this work, we adopt an approach based on the combination of PCA and independent component analysis (ICA) techniques [38], and we compare it with a more straightforward approach named spatial filtering (SF). We then propose a novel solution to the measurement association problem. Since the unmixing operation is not perfect because of noise and shadowing, the output  $\mathbf{Y}$  of BSS contains residual transmitted power profiles from other nodes (crosstalk), which is removed by an excision filter.

### 3.1 Problem Statement

BSS recovers the source signals  $\mathbf{P}$  in (2.7), from a set of observed quantities  $\mathbf{X}$ , when the mixing matrix  $\mathbf{H}$  is unknown [33]. Let us consider the mixing model given by (2.7)



**Figure 3.1:** A block scheme of the blind source separation process.

$$\mathbf{X} = \mathbf{H}\mathbf{P} + \mathbf{N}, \quad (3.1)$$

where  $\mathbf{P} \in \mathbb{R}^{N \times K}$ ,  $\mathbf{H} \in \mathbb{R}^{M \times N}$ ,  $\mathbf{N} \in \mathbb{R}^{M \times K}$ ,  $\mathbf{X} \in \mathbb{R}^{M \times K}$ ,  $N$  is the number of sources, and  $M$  is the number of observations. The goal of the BSS is to find the unmixing matrix  $\mathbf{W} \in \mathbb{R}^{M \times N}$ , as depicted in Fig. 3.1, such that

$$\hat{\mathbf{P}} = \mathbf{W}^T \mathbf{X}, \quad (3.2)$$

where  $\hat{\mathbf{P}}$  is an estimate of  $\mathbf{P}$ . In other words, we aim to find the weights  $w_{i,j}$  that separate the components in  $\mathbf{P}$ .

Let us make some assumptions:

- The mixture signal is zero mean. This means that each row of  $\mathbf{X}$  is zero mean. To satisfy this assumption, it might be necessary to center the observations before the separation process.
- The sources have non-Gaussian distributions. This assumption is further detailed in Section 3.1.1.
- The number of independent observers is larger than (or equal to) the number of sources, i.e.,

$$M \geq N. \quad (3.3)$$

Thus, in this setting the problem is *overdetermined*.

In literature, there are numerous solutions for the BSS problem in different scenarios with various types of source signals. In most cases, the complexity of the problem requires an iterative approach that estimates the sources.

In this work, we adopted a methodology based on ICA. ICA is a data processing method that finds statistically independent and non-Gaussian components from data.

### 3.1.1 Independence and non-Gaussianity

Let us recall some definitions.

**Independence.** Two random variables  $y_1$  and  $y_2$  are said to be independent if

$$p(y_1, y_2) = p(y_1)p(y_2). \quad (3.4)$$

**Non-correlation.** Two random variables  $y_1$  and  $y_2$  are said to be uncorrelated if

$$\mathbb{E}[y_1^2, y_2^2] = 0. \quad (3.5)$$

Independence implies non-correlation but not viceversa. One of the assumptions made for BSS is that the sources are non-Gaussian. This is because, otherwise, ICA cannot resolve the independent directions due to symmetries. The joint density of unit variance Gaussians is symmetric, so it does not contain information about the directions of the unmixing matrix. Thus, the unmixing matrix cannot be estimated in that case.

**Example.** Let us consider  $N = 2$  source signals, rows of  $\mathbf{P}$ , such that

$$\mathbf{p}_1, \mathbf{p}_2 \sim \mathcal{N}(0, \mathbf{I}_N), \quad (3.6)$$

where  $\mathbf{I}$  is a  $2 \times 2$  identity matrix. We then observe

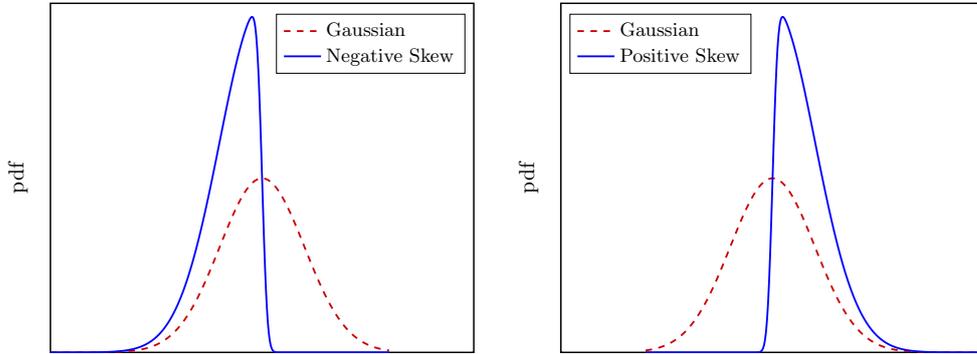
$$\mathbf{X} = \mathbf{A}\mathbf{P}, \quad (3.7)$$

where  $\mathbf{A}$  is the mixing matrix, and  $\mathbf{X}$  follows a Gaussian distribution with zero mean and covariance given by

$$\mathbb{E}[\mathbf{X}\mathbf{X}^T] = \mathbb{E}[\mathbf{A}\mathbf{P}\mathbf{P}^T\mathbf{A}^T] = \mathbf{A}\mathbf{A}^T. \quad (3.8)$$

Thus,

$$\mathbf{X} \sim \mathcal{N}(0, \mathbf{A}\mathbf{A}^T). \quad (3.9)$$



**Figure 3.2:** Examples of positively and negatively skewed distributions.

Now, let  $\mathbf{R}$  be an arbitrary orthogonal matrix such that  $\mathbf{R}\mathbf{R}^\top = \mathbf{I}_N$ . We now define the matrix  $\mathbf{A}'$  as

$$\mathbf{A}' = \mathbf{A}\mathbf{R}. \quad (3.10)$$

If the data has been mixed with  $\mathbf{A}'$  instead of  $\mathbf{A}$ , we would observe  $\mathbf{X}' = \mathbf{A}'\mathbf{P}$  which follows a Gaussian distribution with zero mean and covariance given by

$$\mathbb{E}[\mathbf{X}'\mathbf{X}'^\top] = \mathbb{E}[\mathbf{A}'\mathbf{P}\mathbf{P}^\top\mathbf{A}'^\top] = \mathbf{A}'\mathbf{A}'^\top = \mathbf{A}'\mathbf{R}\mathbf{R}^\top\mathbf{A}'^\top = \mathbf{A}\mathbf{A}^\top. \quad (3.11)$$

Thus,

$$\mathbf{x}' \sim \mathcal{N}(0, \mathbf{A}\mathbf{A}^\top) \quad (3.12)$$

and this implies that  $\mathbf{A}$  is an arbitrary rotational component that cannot be determined from the data.

*Then, why does ICA need a measure of non-Gaussianity to separate the components?*

The central limit theorem states that the distribution of the sum of independent random variables, which itself is a random variable, tends toward a Gaussian distribution as the number of terms in the sum increases. According to this, a mixture signal is “more” Gaussian than each of the sources since it is a linear combination of them. Therefore, if we want to estimate

the sources, we aim to find the components  $\hat{\mathbf{y}}$  that are as less Gaussian as possible. The optimal  $\mathbf{W}$  is the matrix that maximizes the non-Gaussianity of  $\hat{\mathbf{y}} = \mathbf{W}^T \mathbf{X}$ , since this will make  $\hat{\mathbf{y}} \approx \mathbf{y}$ .

*But how do we measure non-Gaussianity?*

### Skewness

Skewness characterizes the asymmetry of a distribution around its mean, as depicted in Fig. 3.2

- A distribution with positive skewness has a tail pulled in the positive direction.
- A negatively skewed distribution has a tail pulled in the negative direction.

### Kurtosis

Kurtosis measures the peakedness or the flatness of a distribution with respect to a normal distribution, as shown in Fig. 3.3.

- If kurtosis is zero, the distribution is Gaussian or *mesokurtic*.
- If kurtosis is positive, the distribution is called *supergaussian* or *leptokurtic*.
- If kurtosis is negative, the distribution is called *subgaussian* or *platykurtic*.

Both these characteristics can be used to measure the non-Gaussianity of a distribution. Kurtosis also has the advantage of being computationally cheap.

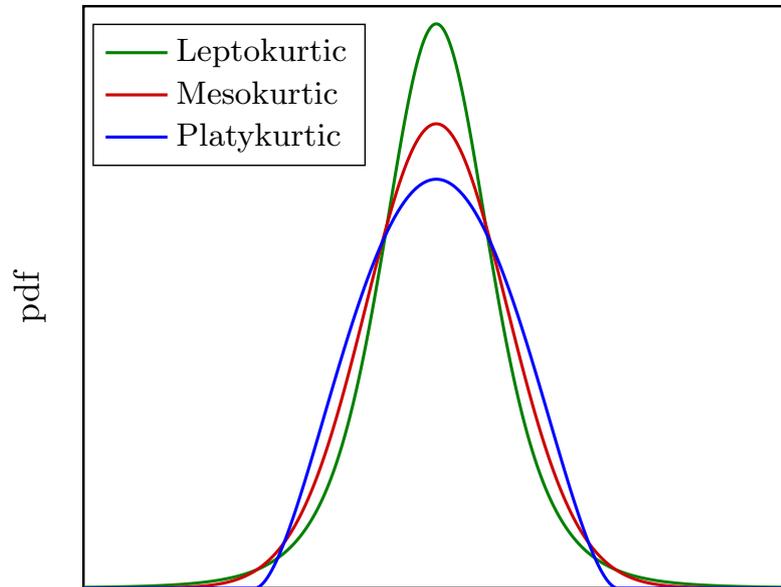


Figure 3.3: Shape of a distribution with respect to its kurtosis.

## 3.2 Whitening, Node Counting, and Dimensionality Reduction

Signal separation can be effective if a pre-processing stage manipulates the data so that there are  $N$  mixtures centered and whitened at its output [33]. Since the number of network nodes  $N$  is unknown, this stage has to estimate the number of sources.

### 3.2.1 Whitening

Firstly, we center the mixtures subtracting the row-wise mean from  $\mathbf{X}$ . Then, we apply a linear transformation to the observations to make their components uncorrelated with unit variance. The whitened signal,  $\tilde{\mathbf{X}}$ , is written as

$$\tilde{\mathbf{X}} = \mathbf{Q}\mathbf{X}, \quad (3.13)$$

where  $\mathbf{Q}$  is the whitening matrix, and depends on the linear transformation chosen for the whitening. In this work PCA has been used.

Starting from the sample covariance matrix of the observations

$$\mathbf{\Sigma} = \frac{1}{K} \mathbf{X} \mathbf{X}^T, \quad (3.14)$$

we perform the eigenvalue decomposition

$$\mathbf{\Sigma} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T, \quad (3.15)$$

where  $\mathbf{U}$  is the orthogonal matrix containing the eigenvectors, and  $\mathbf{\Lambda}$  is a diagonal matrix of the eigenvalues,  $\Lambda_i$ , with  $i = 1, \dots, M$ , sorted in descending order. Thus, the whitening matrix is

$$\mathbf{Q} = \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{U}^T. \quad (3.16)$$

This process removes the linear correlation between the observations such that the decomposed vectors are orthogonal to each other but not necessarily independent.

We can now consider the complete mixing matrix

$$\tilde{\mathbf{H}} = \mathbf{Q} \mathbf{H} = \mathbf{\Lambda}^{-1/2} \mathbf{U}^T \mathbf{H}. \quad (3.17)$$

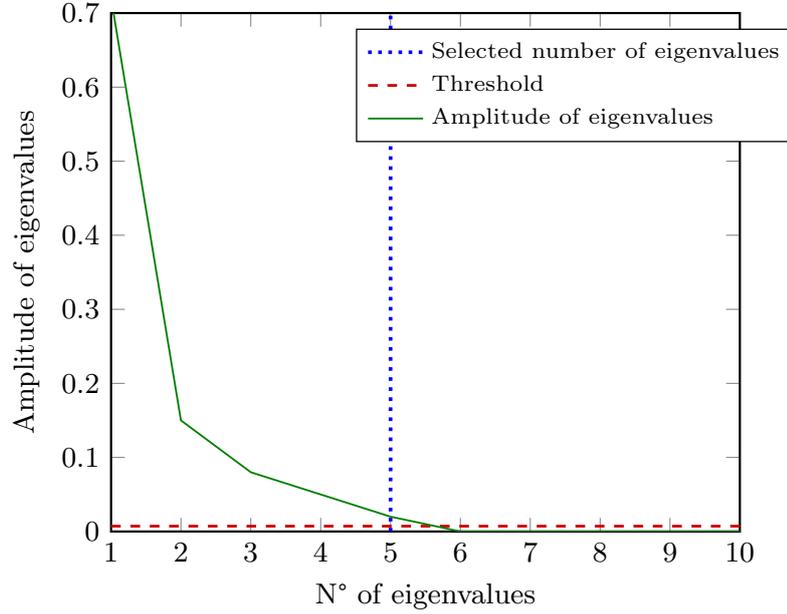
such that

$$\begin{aligned} \tilde{\mathbf{X}} &= \mathbf{\Lambda}^{-1/2} \mathbf{U}^T \mathbf{X} \\ &= \mathbf{\Lambda}^{-1/2} \mathbf{U}^T \mathbf{H} \mathbf{P} \\ &= \tilde{\mathbf{H}} \mathbf{P}. \end{aligned} \quad (3.18)$$

It has been proven that whitening makes the complete mixing matrix orthogonal [39].

### 3.2.2 Node Counting

To estimate the number of sources  $\hat{N}$  generating the mixture, we count the significant eigenvalues of the sample covariance matrix  $\mathbf{\Sigma}$ . This operation can



**Figure 3.4:** An illustration of the eigenvalue selection procedure. The green line shows the amplitude of the eigenvalues, the red dashed line represents the threshold, and the blue dotted line highlights the number of eigenvalues selected.

be performed adopting a model order selection approach based on minimum description length (MDL) criteria [40, 41], or using a threshold.

### Minimum Description Length

The MDL criterion consists of the following minimization problem

$$\hat{N} = \arg \min_{n \in \{1, \dots, M\}} \{\text{MDL}(n)\} \quad (3.19)$$

with

$$\begin{aligned} \text{MDL}(n) = & -\log \left( \frac{\prod_{i=n+1}^M \Lambda_i^{1/(M-n)}}{\frac{1}{M-n} \sum_{i=n+1}^M \Lambda_i} \right)^{(M-n)K} \\ & + \frac{1}{2} n(2M - n) \log K \end{aligned}$$

where  $n$  is the unknown model order.

### Threshold (Scree Plot)

Instead, using a threshold approach, the number of sources generating the mixture is given by the number of significant eigenvalues, i.e.,

$$\hat{N} = \sum_{i=1}^M \mathbb{1}_{\{\Lambda_i > \bar{\Lambda}\}} \quad (3.20)$$

with  $\bar{\Lambda} = w \cdot (\Lambda_1 - \Lambda_M)$ , where  $w \in [0, 1]$  is the eigenvalue selection parameter chosen, e.g., according to the scree plot approach [42].

In PCA, the eigenvalue selection threshold is often set on a percentage of the difference between the largest and the smallest eigenvalues. This approach is known as scree plot and is reported in the literature about PCA [42]. An example of a scree plot taken from our simulation setup is presented in Fig. 3.4. Because the smallest eigenvalues assume almost zero values, we selected a threshold  $w$  that ensures that the relevant eigenvalues are always considered and never discarded. In other words, the value of the parameter  $w$  has been chosen to obtain the best performances for the source counting procedure.

### 3.2.3 Dimensionality Reduction

The ICA method for BSS requires that the number of observations equals the number of sources (or components) to separate. We hypothesized that  $M \geq N$ , thus, we have to be sure that the dimensionality of the mixtures is reduced from  $M$  to  $N$ .

Since the number of sources,  $\hat{N}$ , has been estimated, we project the mixture onto the subspace spanned by the eigenvectors corresponding to the  $\hat{N}$  largest eigenvalues, reducing the dimensionality from  $M$  to  $\hat{N}$ . If the counting process is successful, the estimated number of sources will equal the real one, so that  $\hat{N} = N$ .

Considering the matrix  $\mathbf{\Lambda}$  in (3.16), defined as

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_M \end{bmatrix}, \quad (3.21)$$

dropping to zero the eigenvalues  $\lambda_k$  with  $\hat{N} < k \leq M$  we obtain

$$\tilde{\mathbf{\Lambda}} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{\hat{N}} \end{bmatrix}. \quad (3.22)$$

Then, we can define a new projection matrix,  $\tilde{\mathbf{Q}}$ , obtained from the first  $\hat{N}$  rows of  $\mathbf{Q}$ , so that the whitened mixture is

$$\tilde{\mathbf{X}} = \tilde{\mathbf{Q}}\mathbf{X}. \quad (3.23)$$

### 3.3 Independent Component Analysis

ICA is a data processing method that finds statistically independent and non-Gaussian components from data. In our setting, ICA is applied to  $\tilde{\mathbf{X}}$  to reconstruct the transmitted power profiles  $\mathbf{P}$  in (2.7). Its output is an unmixing matrix  $\mathbf{W} \in \mathbb{R}^{\hat{N} \times \hat{N}}$  such that

$$\tilde{\mathbf{Y}} = \mathbf{W}^T \tilde{\mathbf{X}} \quad (3.24)$$

where  $\tilde{\mathbf{Y}} \in \mathbb{R}^{\hat{N} \times K}$  is the matrix of the separated components. We propose Fast-ICA, an iterative algorithm with kurtosis as a measure of non-Gaussianity, and decorrelation based on the Gram-Schmidt orthonormalization [38, 39]. The estimation of the  $n$ -th column of the unmixing matrix requires the following steps:

- 1) Estimate  $\mathbf{w}_{:,n}$  using the update rule [38, eq. 20]

$$\mathbf{w}_{:,n} \leftarrow 4 \begin{pmatrix} \langle \tilde{\mathbf{x}}_{1,:} \odot (\tilde{\mathbf{w}}_{:,n}^T \tilde{\mathbf{X}})^3 \rangle \\ \dots \\ \langle \tilde{\mathbf{x}}_{\hat{N},:} \odot (\tilde{\mathbf{w}}_{:,n}^T \tilde{\mathbf{X}})^3 \rangle \end{pmatrix} - 3\tilde{\mathbf{w}}_{:,n}. \quad (3.25)$$

- 2) Subtract from  $\mathbf{w}_{:,n}$  its projections on the components  $\mathbf{w}_{:,i}$  with  $i = 1, \dots, n-1$ . That is

$$\mathbf{w}_{:,n} = \mathbf{w}_{:,n} - \sum_{i=1}^{n-1} (\mathbf{w}_{:,n}^T \mathbf{w}_{:,i}) \mathbf{w}_{:,i}. \quad (3.26)$$

- 3) Normalize  $\mathbf{w}_{:,n}$  as

$$\mathbf{w}_{:,n} = \frac{\mathbf{w}_{:,n}}{\|\mathbf{w}_{:,n}\|_2}. \quad (3.27)$$

- 4) Repeat from step (1) until the convergence condition is reached.

The complete iterative method is reported in Algorithm 1, where  $\epsilon_t$  is the termination parameter.

F-ICA algorithm has two main issues.

- 1) The separation process does not preserve the energy of the source signals. Luckily, the time characterization of the signals is sufficient for the extraction of wireless network analytics. Thus, information about the energy of the source signals is not needed.
- 2) The order of recovered signals is not preserved; thus,  $\mathbf{P}$  could be obtained from  $\tilde{\mathbf{Y}}$  through a permutation of the rows. In the next section, a novel solution to this issue, tailored for our scenario is proposed.

### 3.4 Association Problem

We propose two variants of an iterative method to associate the reconstructed sequences to the network nodes and measure the correctness of this matching.

**Algorithm 1:** Fast-ICA for BSS

---

**Input** : Whitened signals  $\tilde{\mathbf{X}}$ ,  $\epsilon_t > 0$ ,  $\hat{N}$   
**Output:** Unmixing matrix  $\mathbf{W}$

- 1  $\epsilon \leftarrow \infty$
- 2  $\mathbf{W} \leftarrow$  random initialization such that  $\|\mathbf{W}\|_1 = 1$
- 3  $\tilde{\mathbf{W}} \leftarrow \mathbf{W}$
- 4 **for**  $n$  from 1 to  $\hat{N}$  **do**
- 5     **while**  $\epsilon \geq \epsilon_t$  **do**
- 6          $\mathbf{w}_{i,n} \leftarrow 4 \langle \tilde{\mathbf{x}}_{i,:} \odot (\tilde{\mathbf{w}}_{:,n}^T \tilde{\mathbf{X}})^3 \rangle - 3 \tilde{\mathbf{w}}_{i,n}, \forall i = 1, \dots, \hat{N}$
- 7          $\mathbf{w}_{:,n} \leftarrow \mathbf{w}_{:,n} - \sum_{i=1}^{n-1} (\mathbf{w}_{:,n}^T \mathbf{w}_{:,i}) \mathbf{w}_{:,i}$
- 8          $\mathbf{w}_{:,n} \leftarrow \frac{\mathbf{w}_{:,n}}{\|\mathbf{w}_{:,n}\|_2}$
- 9          $\epsilon = \sum_{m=1}^{\hat{N}} |w_{m,n} - \tilde{w}_{m,n}|$
- 10          $\tilde{\mathbf{w}}_{:,n} \leftarrow \mathbf{w}_{:,n}$
- 11     **end**
- 12 **end**

---

**Variante 1.** For a given a permutation of the rows of the unmixed signal  $\tilde{\mathbf{Y}}$ , a path-loss law is applied to mimic the signal propagation to each sensor. This corresponds to a mixing operation but, this time, considering  $\tilde{\mathbf{Y}}$  as source signal, instead of  $\mathbf{P}$ . Subsequently, a row-by-row correlation between the real mixed matrix  $\mathbf{X}$  and the one obtained by mixing  $\tilde{\mathbf{Y}}$  is performed. The maximum of the correlation quantifies the correctness of the permutation. This operation is repeated for each possible permutation of the rows of  $\tilde{\mathbf{Y}}$  and the permutation with the highest correlation peak is chosen and used to get  $\mathbf{P}$ . This method works remarkably well but requires an exhaustive search among all the possible permutations, with complexity  $\mathcal{O}(N!)$ .

**Variante 2.** When dealing with large networks, we propose the following lightweight alternative. On this purpose, let us define the matrix  $\mathbf{D} \in \mathbb{R}^{M \times \hat{N}}$ , whose elements  $d_{m,n}$  are the distances between sensor  $m$  and node  $n$ . Let us hypothesize that the position of the network nodes is known, so that the distances between the nodes and the sensors can be estimated. As will be shown in Section 4, the network nodes localization can be performed by the sensor network in a phase preceding the signal association [43,44]. Moreover,

---

**Algorithm 2:** Unmixed signals association.

---

**Input** : Unmixed signal  $\tilde{\mathbf{Y}}$ ,  $\hat{N}$ ,  $\mathbf{D}$

**Output:** Aligned unmixed signal  $\mathbf{Y}$

```

1 for  $n$  from 1 to  $\hat{N}$  do
2    $i \leftarrow \arg \min_m \{d_{m,n}\}$ 
3   for  $j$  from 1 to length of  $\tilde{\mathbf{y}}_{:,1}$  do
4      $peaks_j \leftarrow \max \{\text{corr}(\tilde{\mathbf{y}}_{j,:}; \mathbf{x}_{i,:})\}^\dagger$ 
5   end
6    $k \leftarrow \arg \max_j \{\mathbf{peaks}\}$ 
7    $\mathbf{Y}_{n,:} \leftarrow \tilde{\mathbf{y}}_{k,:}$ 
8    $\tilde{\mathbf{Y}} \leftarrow \tilde{\mathbf{Y}}/\tilde{\mathbf{y}}_{k,:}^\ddagger$ 
9 end

```

---

<sup>†</sup>  $\text{corr}(\mathbf{a}; \mathbf{b})$  operator returns the cross-correlation between vectors  $\mathbf{a}$  and  $\mathbf{b}$ .

<sup>‡</sup>  $\mathbf{A}/\mathbf{a}_{k,:}$  operator removes the  $k$ th row from the matrix  $\mathbf{A}$ .

as will be clarified in Chapter 7.2, only a coarse estimate of nodes position is required in this phase.

The proposed algorithm is the following.

- First, we select a node  $n$  from a list of all the nodes of the network and find its nearest RF sensor  $m$ .
- Then, we correlate the sequence measured at sensor  $m$  with all the unmixed sequences (rows of  $\tilde{\mathbf{Y}}$ ) separately.
- The row  $\tilde{\mathbf{y}}_{j,:}$  that shows the highest positive correlation is associated with  $n$  and, thus, is copied into the  $n$ th row of  $\mathbf{Y}$ .
- Then, we remove node  $n$  from the list, delete the  $j$ th sequence from  $\tilde{\mathbf{Y}}$  and iterate the algorithm.

The method is detailed in Algorithm 2. Its complexity is  $\mathcal{O}(\hat{N} \log \hat{N})$ , which is also acceptable when dealing with large networks.

### 3.5 Spatial Filtering

As an alternative to BSS, we propose another approach [45] where a path-loss law is used to weight the sensors measurements and reconstruct the source

signals via linear filtering. The estimation of the number of sources,  $\hat{N}$ , is obtained via the MDL criteria (3.20) or the threshold method 3.20. Then, we build the matrix  $\boldsymbol{\alpha} \in \mathbb{R}^{M \times \hat{N}}$  of weights inversely proportional to the path-loss between sensor  $m$  and node  $n$  as

$$\alpha_{m,n} = (d_{m,n})^{-\eta}$$

where  $\eta$  is the filtering parameter. The matrix  $\mathbf{Y} \in \mathbb{R}^{\hat{N} \times K}$  containing the temporal power profiles reconstructed for all the  $\hat{N}$  nodes is given by

$$\mathbf{Y} = \boldsymbol{\alpha}^T \mathbf{X}.$$

As a result, the traffic profiles generated by distant nodes are filtered out, to some extent, by the weights  $\alpha_{m,n}$ . This approach is simpler than the BSS algorithm based on ICA, but it requires the choice of a filtering parameter  $\eta$  that has to be tuned depending on the specific propagation scenario and sensor and nodes deployment. Note that the parameter  $\eta$  regulates the intensity of the spatial filtering and might not reflect the true path-loss exponent. Furthermore, the performance of this algorithm is strongly influenced by the presence of shadowing, as shown in Chapter 7.

## 3.6 Excision Filter

The algorithms for the extraction of the wireless network analytics are based on the temporal characterization of the packet flows exchanged by the nodes. To extract temporal features, it is necessary to process the time series in  $\mathbf{Y}$  to obtain sequences of 0s and 1s; this is carried out by an excision filter that forces to zero power samples due to crosstalk. The output of the filter is the matrix  $\mathbf{Z}$  with elements

$$z_{n,k} = \begin{cases} 1 & \text{if } y_{n,k} \geq \zeta_n \\ 0 & \text{otherwise} \end{cases}$$

where the threshold  $\zeta_n$  is set as a fraction  $q \in [0, 1]$  of the maximum of  $\mathbf{y}_{n,:}$ , i.e.,

$$\zeta_n = q \cdot \max_k \{y_{n,k}\}, \quad n = 1, \dots, \hat{N}. \quad (3.28)$$



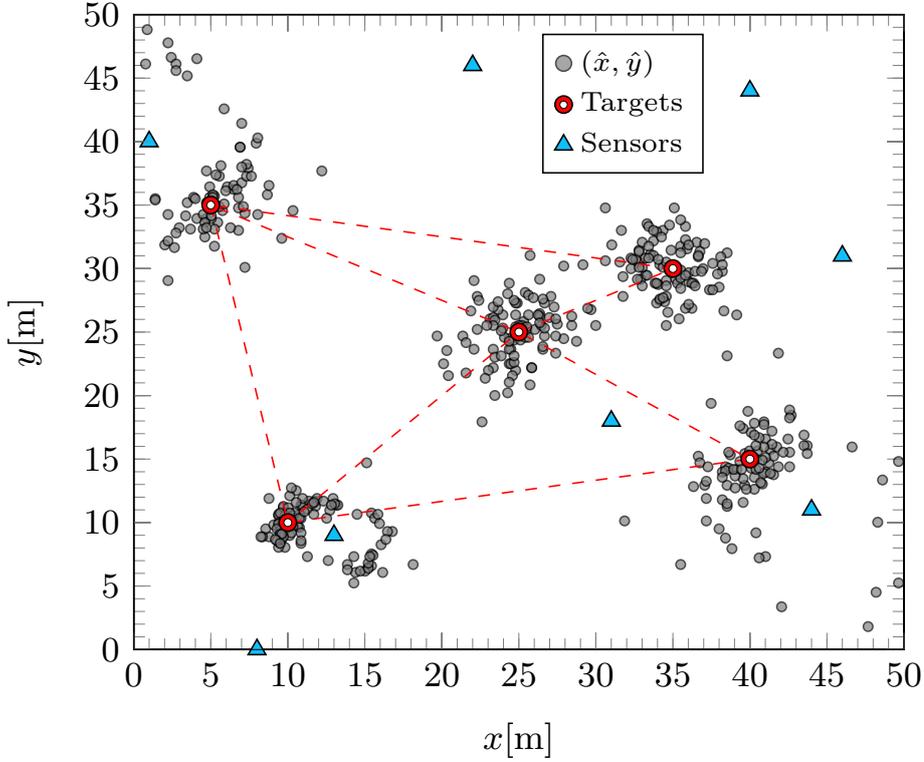
# Chapter 4

## Multiple Transmitter Localization

### 4.1 Problem Statement

We propose a novel methodology for locating packet-based non-collaborative wireless network nodes using over-the-air power profiles captured by RF sensors. In particular, as already mentioned, the sensors do not have access to the target network and ignore its main features (i.e., the number of nodes, modulation type, and MAC). Hence, the RF sensors can only measure the aggregate power received by the nodes. Then, a fusion center processes the power measurements to estimate the position of the nodes. This problem is challenging because nodes may transmit simultaneously in the same frequency band, sensors can be placed in an unfavorable geometrical configuration, and sensing is hindered by path-loss, shadowing, and measurement noise.

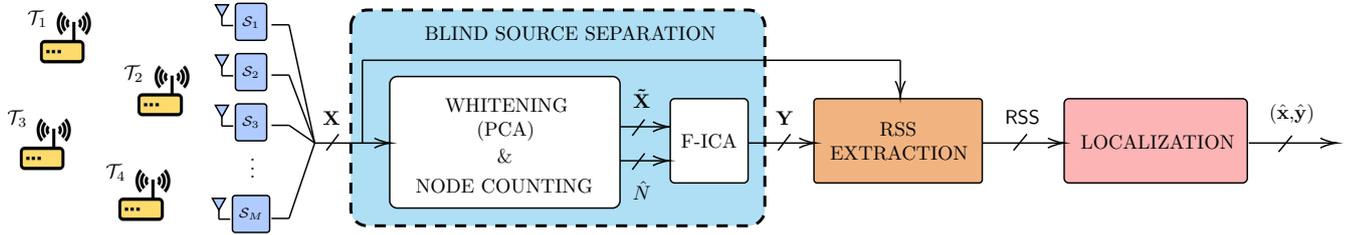
Another localization problem is tackled in Chapter 8. In particular, we propose a navigation system which allows a fleet of UAVs to navigate across a scenario and find the best spatial configuration to localize a wireless transmitter. The patrol can use this system in synergy with the network analytics tool to localize a malicious user (i.e., jammer) and secure a wireless network.



**Figure 4.1:** An example of scenario with  $N = 5$ ,  $M = 8$ , and  $\sigma = 3$  dB. The red circles are the network nodes (or targets), the blue triangles are the RF sensors and the grey circles represent the estimate of the position of the nodes through the BSS-MLE methodology.

### 4.1.1 Existing Works

Over the past decades, several fundamental contributions on localization have been exploiting different technologies, such as WiFi and UWB [46–51]. Furthermore, the problem of multi-target localization using groups of cooperating sensors has also been widely investigated [52, 53], including an original framework that ensures scalability and distributed implementation [48]. However, all the above works assume that the targets can exchange information with the anchors, thus contributing to localization. In [54], a device-free multiple target localization technique based on sensor radars is proposed. In [55], the authors propose an algorithm for the simultaneous localization of multiple non-collaborative users through RSS measurements carried out by a UAV. This methodology is based on a score map constructed using the



**Figure 4.2:** Block scheme of the complete methodology: target nodes ( $\mathcal{T}_1, \mathcal{T}_2, \dots$ ), RF sensors ( $\mathcal{S}_1, \mathcal{S}_2, \dots$ ), node counting (principal component analysis (PCA)), received power profiles separation (fast independent component analysis (F-ICA)), transmitted power profile reconstruction (RSS extraction), and localization.

power measured by a UAV moving across the scenario. Then, a set of ML algorithms are applied to the score map to estimate the number and the positions of the transmitters. In [56], the authors propose a particle simulation algorithm for the localization of wireless transmitters leveraging many RF sensors distributed on a grid.

In this work, we propose a novel solution for the localization of multiple unknown transmitters via BSS [57]. The processing chain depicted in Fig. 4.2 shows the stages of the algorithm, while an example of a typical scenario is shown in Fig. 4.1. Firstly, a BSS is performed to count the number of targets and reconstruct the power profiles transmitted by them. Then, through a non-linear filtering procedure, the RSS associated with each sensor-target couple is extracted. After this filtering step, the sensors can localize the target nodes separately using any conventional positioning technique based on RSS. For instance, in this work, localization is performed through least squares (LS) and MLE techniques. In the numerical results, the methodology's performance is derived in different shadowing regimes and varying the sensors' spatial density, and it is compared with the solution proposed in [56].

## 4.2 Received Signal Strength Extraction

Although the F-ICA technique applied in the power-domain suits this scenario, it presents a relevant issue. The reconstructed signals are scaled and

do not preserve the original power. For this reason, another processing stage is necessary to perform the RSS-based localization.

Let us normalize the reconstructed power profiles  $\mathbf{Y}$  such that  $\tilde{\mathbf{Y}}$  has elements  $\tilde{y}_{n,k} = 1$  if node  $n$  is transmitting in the  $k$ -th bin, and  $\tilde{y}_{n,k} = 0$  otherwise. Such normalization can be performed using the excision filter, i.e.,

$$\tilde{y}_{n,k} = \mathbb{1}_{\{y_{n,k} \geq \zeta_n\}}. \quad (4.1)$$

where  $\zeta_n = q \cdot \max_k \{y_{n,k}\}$  is a normalization threshold chosen as a fraction  $q \in [0, 1]$  of the maximum power of the transmit profile of target node  $n$ . Thus, each row  $\tilde{\mathbf{y}}_n$  can be seen as a mask that, multiplied element-wise by the  $m$ -th row of  $\mathbf{X}$ ,  $\mathbf{x}_m$ , forces to zero all the power samples received by sensor  $m$  that have not been transmitted by node  $n$ . Fig. 4.3 depicts the procedure with an example of two partially overlapped transmissions, where the normalized reconstructed power profile  $\tilde{\mathbf{y}}_1$ , when multiplied element-wise by, e.g.,  $\mathbf{x}_1$ , forces to zero all the power samples received by sensor  $\mathcal{S}_1$  that have not been transmitted by node  $\mathcal{T}_1$ . Then, the received signal strength between the target node  $n$  and the sensor  $m$ ,  $\text{RSS}_{n,m}$ , is obtained by averaging over the non-zero entries of the result of the element-wise product. Such process can be expressed in a compact form as

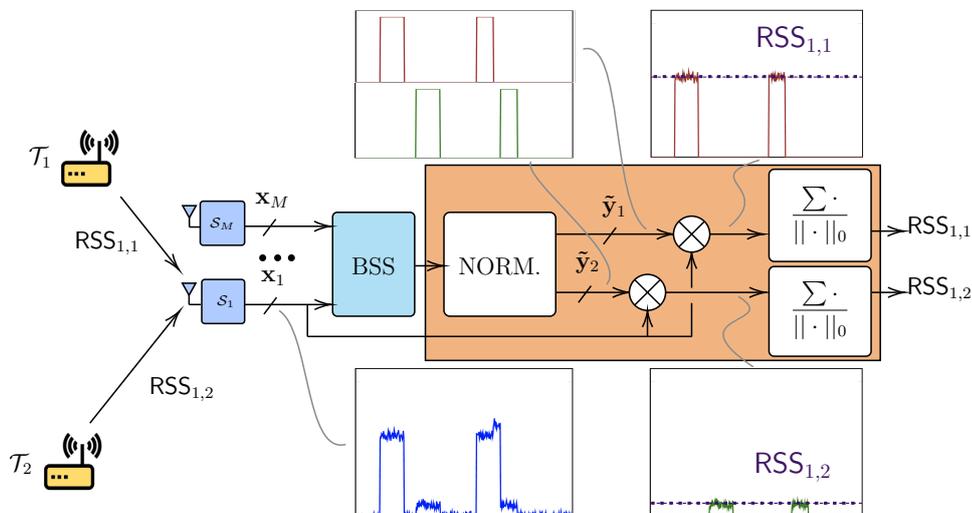
$$\text{RSS}_{m,n} = \frac{\sum_{k=1}^K x_{m,k} \tilde{y}_{n,k}}{\|\mathbf{x}_m \odot \tilde{\mathbf{y}}_n\|_0} \quad m = 1, \dots, M \quad n = 1, \dots, N, \quad (4.2)$$

where  $\odot$  stands for the element-wise product. The averaging ensures that the RSS is estimated within a time frame of duration  $T_{\text{ob}}$ .

### 4.3 Localization

The position estimation is obtained through a two-dimensional RSS-based localization algorithm.<sup>1</sup> Let us assume the targets are located at unknown coordinates  $(x_n, y_n)$  with  $n = 1, \dots, N$  and the RF sensors are at known

<sup>1</sup>For simplicity, height differences between nodes and sensors are considered negligible with respect to the distance between them.



**Figure 4.3:** The RSS extraction process for two non-collaborative transmitters.

positions  $(\tilde{x}_m, \tilde{y}_m)$  with  $m = 1, \dots, M$ . In this work, two well-known solutions for the localization of single targets (the generic node  $n$  in the following) are considered.

### 4.3.1 Least Squares

Let us build a matrix  $\mathbf{B}$ , that contains RSS measurements obtained through (4.1)-(4.2) and the sensors position, with rows

$$\mathbf{b}_s = (2\tilde{x}_m, 2\tilde{y}_m, \text{RSS}_{m,n}^{-1/\nu}, -1) \quad (4.3)$$

for  $m = 1, \dots, M$ , and a vector

$$\mathbf{q} = (\tilde{x}_1^2 + \tilde{y}_1^2, \dots, \tilde{x}_M^2 + \tilde{y}_M^2)^\top. \quad (4.4)$$

Let us also define the two unknowns,  $D^2 = x_n^2 + y_n^2$  and  $P = (P_{\text{tx}} h_0)^\frac{1}{\nu}$ , where  $P_{\text{tx}}$  is the transmit power of the nodes. The localization problem can be formulated as a system of linear equations

$$\mathbf{B} \mathbf{p} = \mathbf{q}, \quad (4.5)$$

where  $\mathbf{p} = (x_n, y_n, P, D^2)^\top$  is the desired solution which can be obtained by the ordinary LS method [58]

$$\hat{\mathbf{p}} = \arg \min_{\mathbf{p} \in \mathbb{R}^4} \{ \|\mathbf{B}\mathbf{p} - \mathbf{q}\|_2^2 \} = (\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{q}. \quad (4.6)$$

### 4.3.2 Maximum Likelihood

Without any prior statistical knowledge about the transmit power and location, the ML estimation of the  $n$ -th target location is given by [58, 59]

$$(\hat{x}_n, \hat{y}_n) = \arg \min_{(x_n, y_n) \in \mathbb{R}^2} \left\{ \sum_{i=1}^M \left( \ln(\text{RSS}_{i,n} d_{i,n}^{2\nu}) - \frac{1}{M} \sum_{j=1}^M \ln(\text{RSS}_{j,n} d_{j,n}^{2\nu}) \right)^2 \right\} \quad (4.7)$$

where  $d_{i,n} = \sqrt{(x_n - \tilde{x}_i)^2 + (y_n - \tilde{y}_i)^2}$ . The objective function (4.7) is differentiable with respect to  $(x_n, y_n)$ , hence it is possible to find the minimum in closed form or via the gradient descent method. However, if the target nodes are arranged in an unfavorable configuration, there can be several local minima. For this reason, the monitored area has been discretized into a two-dimensional grid, and the grid point that gives the minimum value of (4.7) is chosen. The finer the grid, the more accurate the estimation is at the cost of an increased computational burden.

### 4.3.3 Particle Simulation

The solution proposed in this work has been compared with the localization algorithm presented in [56]. In fact, in [56] the authors propose an RSS-based particle simulation algorithm for the localization of a set of non-collaborative wireless transmitters using a network of RF sensors deployed on a grid. The method also accounts for shadowing, so it perfectly fits our scenario.

The main idea comes from a physical interpretation. All the RF sensors are considered as fixed particles. In addition, there are  $N$  estimated transmitters located at some positions which are considered as particles that are

allowed to move during the iterations of a particle simulation. The average received power, which the sensors would receive from the transmitters at the positions estimated by the algorithm, can be calculated using a path-loss law. The difference between the measured receive power and the power estimated through the path-loss law (in log domain) is called potential. Instead of directly minimizing the localization error, which is not available, the authors try to minimize the total potential. A positive (negative) potential indicates that a sensor measures a higher (lower) receive power than the one which would be expected from the transmitter at the estimated position. Based on this interpretation, a force from each sensor on each estimated transmitter is introduced. The magnitude of this force is determined by the absolute value of the potential and decays quadratically with the distance (e.g., gravitational force). Hence, the closer a sensor is to a transmitter and/or the larger the potential, the stronger the induced force. Finally, the potential sign determines whether a sensor pulls the estimated transmitters towards itself or pushes them away, depending on whether they measure a too weak or a too strong receive power, respectively.

It can be noted that without any random effects (i.e., shadowing) the forces would be zero when the estimated positions of the transmitters coincide with the real ones, such that equilibrium would be established. However, equilibria can also be established in points where non-zero forces cancel each other out, freezing the algorithm in local minima. A transmit power estimation is incorporated to enable also the localization of unknown transmitters. The performance of such localization method is used as a benchmark in Chapter 7.

## 4.4 Cramèr-Rao Lower Bound

To assess the asymptotical improvement of the localization performance of the proposed algorithms, the CRLB has been derived. In particular, considering the RSS-based localization of a single transmitter  $n$  through the observations of  $M$  RF sensors, the received power at the  $m$ -th sensor is

$$P_{m,n} = P_{\text{tx}} h_{m,n} = P_{\text{tx}} h'_{m,n} e^{\sigma_S s_{m,n}}, \quad (4.8)$$

where  $P_{\text{tx}}$  is the transmit power, and the path-loss model is of power-law type with channel gain  $h'_{m,n} = h_0 (\frac{d_0}{d_{m,n}})^\nu$ , path-loss exponent  $\nu$  and reference distance  $d_0$ . Thus, the received signal strength is

$$\begin{aligned} \text{RSS}_{m,n} &= 10 \log_{10} P_{m,n} \\ &= 10 \log_{10} (P_{\text{tx}} h_0 d_0^\nu) - 10\nu \log_{10} d_{m,n} + 10 \log_{10} e^{\sigma_S s_{m,n}} \\ &= \text{RSS}(d_0) - 10\nu \log_{10} d_{m,n} + \frac{10}{\ln 10} \sigma_S s_{m,n} \\ &= \bar{P}_{m,n} + s_{m,n} \sigma_S^{\text{dB}}, \end{aligned} \quad (4.9)$$

where  $\bar{P}_{m,n}$  is the true  $\text{RSS}_{m,n}$  expressed in deciBelWatt,  $s_{m,n} \sim \mathcal{N}(0, 1)$  and  $\sigma_S^{\text{dB}}$  is the shadowing parameter expressed in deciBel. For the sake of clarity, in the following, we will refer to  $\sigma_S^{\text{dB}}$  as  $\sigma_S$ .

To derive the CRLB we need to define:

- the unknown parameters vector  $l_n = (x_n, y_n)^\top$ ;
- the vector of observables (or measurements)

$$\text{RSS}_n = (\text{RSS}_{1,n}, \text{RSS}_{2,n}, \dots, \text{RSS}_{M,n}). \quad (4.10)$$

Knowing that  $\text{RSS}_{m,n} \sim \mathcal{N}(\bar{P}_{m,n}, \sigma_S^2)$ , if the measurements at different sensors are independent, the distribution of the observables given the unknown parameters is

$$\begin{aligned} f(\text{RSS}_n, l_n) &= \prod_{m=1}^M f(\text{RSS}_{m,n}, l_n) \\ &= \prod_{m=1}^M \frac{1}{\sqrt{2\pi\sigma_S^2}} e^{-\frac{1}{2\sigma_S^2} (\text{RSS}_{m,n} - \bar{P}_{m,n})^2}. \end{aligned} \quad (4.11)$$

Then, the log-likelihood of the observables given the unknown parameters

is

$$\ln f(\text{RSS}_n, l_n) = \sum_{m=1}^M -\frac{1}{2} \ln 2\pi\sigma_S^2 - \frac{1}{2\sigma_S^2} (\text{RSS}_{m,n} - \text{RSS}(d_0) + 10\nu \log_{10} d_{m,n}). \quad (4.12)$$

The elements of Fisher's information matrix (FIM) can now be calculated as

$$[\mathbf{I}(l_n)]_{1,1} = -\mathbb{E} \left[ \frac{\partial^2}{\partial x_n^2} \ln f(\text{RSS}_n, l_n) \right], \quad (4.13)$$

$$[\mathbf{I}(l_n)]_{1,2} = -\mathbb{E} \left[ \frac{\partial^2}{\partial x_n \partial y_n} \ln f(\text{RSS}_n, l_n) \right], \quad (4.14)$$

$$[\mathbf{I}(l_n)]_{2,1} = -\mathbb{E} \left[ \frac{\partial^2}{\partial y_n \partial x_n} \ln f(\text{RSS}_n, l_n) \right], \quad (4.15)$$

$$[\mathbf{I}(l_n)]_{2,2} = -\mathbb{E} \left[ \frac{\partial^2}{\partial y_n^2} \ln f(\text{RSS}_n, l_n) \right]. \quad (4.16)$$

The complete  $2 \times 2$  FIM is then

$$\mathbf{I}(l_n) = \frac{100\nu^2}{\sigma_S^2 (\ln 10)^2} \begin{bmatrix} \sum_{m=1}^M \frac{(x_m - x_n)^2}{d_{m,n}^4} & \sum_{m=1}^M \frac{(x_m - x_n)(y_m - y_n)}{d_{m,n}^4} \\ \sum_{m=1}^M \frac{(x_m - x_n)(y_m - y_n)}{d_{m,n}^4} & \sum_{m=1}^M \frac{(y_m - y_n)^2}{d_{m,n}^4} \end{bmatrix}, \quad (4.17)$$

and the lower bound of the root mean squared error (RMSE) is

$$\text{RMSE}_n \geq \sqrt{[\mathbf{I}^{-1}(l_n)]_{1,1} + [\mathbf{I}^{-1}(l_n)]_{2,2}}. \quad (4.18)$$



# Chapter 5

## Topology Inference

### 5.1 Problem Statement

As a reminder, we assume that the technical specifications of the network (i.e., physical layer signals, MAC and routing protocols) are unknown. The topology of the network is represented by a directed graph and its associated adjacency matrix  $\mathbf{A} \in \{0, 1\}^{N \times N}$  where

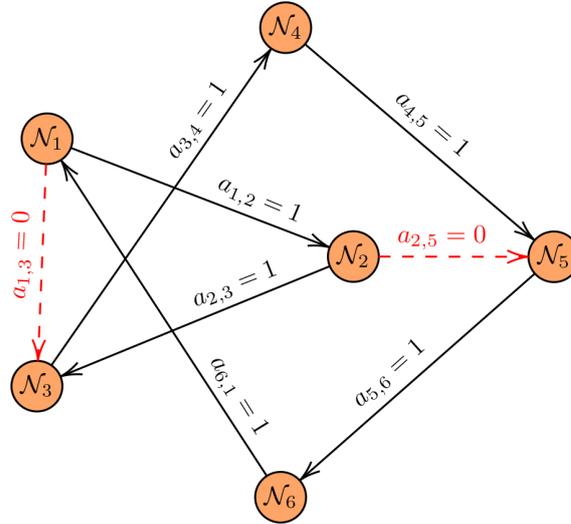
$$a_{i,j} = \begin{cases} 1 & \text{if information flows from node } i \text{ to } j \\ 0 & \text{otherwise.} \end{cases}$$

For example, with reference to Fig. 5.1,  $a_{1,2} = 1$  and  $a_{1,3} = 0$  mean that information flows from node 1 to node 2 but not from node 1 to node 3.

The goal is to find an estimate of the adjacency matrix,  $\widehat{\mathbf{A}}$ , of the wireless network from raw RF measurements carried out by sensors within an observation window of duration  $T_{\text{ob}}$ . Since no interaction is expected between the observed network and the sensors, all the subsequent tasks, summarized in Fig. 5.2, are performed without demodulating the received signals, so that a simple ED receiver suffices [28, 30, 31].

#### 5.1.1 Existing works

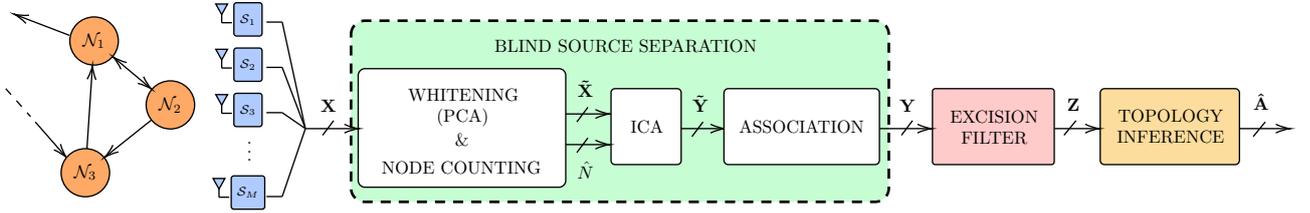
Different approaches and methodologies for network topology inference have been proposed in the literature. Some of them rely on access to the packet's



**Figure 5.1:** An example of directed graph representing a wireless network’s logical topology. The dashed links in red represent the absence of flow of information between the nodes.

content, which is not always feasible and may increment the network overhead [60, 61]. In [62], a path inference approach that exploits routing information within packets is proposed. Others fall into the network tomography category, which requires access to information at endpoints [63, 64]. For example, in [65], a low-complexity inference algorithm based on the Kullback–Leibler (KL) divergence that requires a link rate estimation is developed. Without accessing the packet content, the solution proposed in [66] exploits spectral coherence based on the Lomb-Scargle periodogram to measure causality between two signals. Such an approach relies upon the notion of correlation, which, in principle, does not necessarily imply causation. In [67], a Bayesian nonparametric model to learn the topology of an unknown ad-hoc network is proposed; the solution is based on a hidden semi-Markov model (HSMM) for segmenting nodes transmission activity.

A different research field that contributes to topology inference is represented by graph signal processing (GSP) applied to networks. Graph learning as an edge subset selection problem or a neighborhood-based sparse linear regression is proposed in [68]. In [69], non-linear structural equation models for detecting the topology of a graph from the observations of a process



**Figure 5.2:** Block scheme of the wireless network topology inference: sensor network, data pre-processing and inference.

propagating through it are investigated. In [70], a novel method based on an elastic net solver [71], that performs well even in scenarios where the data are highly correlated, is presented. However, topology inference in GSP assumes that the network is subject to a diffusion process, i.e., there is a piece of information propagating among all the nodes.

### Causality

The task of network topology inference can be seen as learning temporal causal structures among multiple time series. This has roots in the causal inference problem described by Pearl [72] and Granger [73–75]. In particular, the Granger test based on an auto-regressive (AR) model introduced in [73] has become the basis for further causal analysis and topology inference methods. An approach for causal inference on networks involving a specific formulation of Granger causality (GC), named asymmetric Granger causality (AGC), is exploited in [76], where the parametric tests are carried out over groups of time series. Hawkes point processes are a statistical tool to model causal relationships, and recently their connection with GC has been investigated [77]. Multivariate Hawkes processes for topology inference through causal analysis between time series are exploited in [78,79]. Another well-known tool for causal inference is based on the information-theoretic measure, named transfer entropy (TE), proposed in [80]. In [81], the authors propose a TE-based topology inference approach and evaluate its robustness with respect to GC. To overcome some limits of TE, [82] proposes conditional entropy (CE), an approach for causal inference on networks that is optimum under certain *Markovian* assumptions. A non-parametric learning method

related to GC and TE that measure the impact of one node activity over another is developed in [83]. An interesting feature of this approach is that prior knowledge of the underlying network protocol is not needed.

## 5.2 Causality Inference

Identifying the relationship between a cause and its effect is known as causal reasoning. In the last centuries, many philosophers and scientists raised interest in causal inference, the process of studying the response of an effect variable when there is a change in its cause. The objective of causal inference is to provide evidence of the relationship hypothesized by causal reasoning. The study of causal inference has been embraced by many scientists across many scientific and humanistic areas. From econometric [73], to computational neuroscience and neuroimaging [84–86], to psychology [87], up to the telecommunications [76, 81].

The techniques for causality inference can be based on the analysis of observational data or time series, as in our case. The operating principle is more or less the same for all the causal inference methodologies: the acquired data are analyzed and used to build a proper decision test that is able to detect the causality between a possible cause and the possible effect. The most famous state-of-the-art causality inference approaches to detect causality from time series are briefly explained in the following sections.

### 5.2.1 Granger Causality

GC test methods are based on linear  $L$ -order AR models. In particular, considering a pair of time series  $\mathbf{z}_i$  and  $\mathbf{z}_j$  (i.e., two rows of  $\mathbf{Z}$  that correspond to the transmitted power profiles of nodes  $i$  and  $j$ , respectively) two models

(hypotheses) can be formulated

$$\mathcal{H}_1 : z_{j,k} = \sum_{l=1}^L \beta_l z_{j,k-l} + \sum_{l=1}^L \gamma_l z_{i,k-l} + \varepsilon_k \quad (5.1)$$

$$\mathcal{H}_0 : z_{j,k} = \sum_{l=1}^L \delta_l z_{j,k-l} + \omega_k \quad (5.2)$$

where  $\{\beta_l\}_{l=1}^L$ ,  $\{\gamma_l\}_{l=1}^L$ , and  $\{\delta_l\}_{l=1}^L$  are the regression coefficients, and  $\varepsilon_k$ ,  $\omega_k$ , are samples of independent AWGN. The model (5.1) corresponds to hypothesis  $\mathcal{H}_1$  and considers the possibility of a causal relationship between the two time series. At the same time, (5.2) is the null hypothesis  $\mathcal{H}_0$  and excludes the contribution of the past values of  $\mathbf{z}_i$  in the prediction of  $\mathbf{z}_j$ . Note that (5.2) can be seen as a particular case of (5.1) where  $\gamma_l = 0$ ,  $l = 1, \dots, L$ . This means that if  $\mathbf{z}_i$  *Granger causes*  $\mathbf{z}_j$  the prediction error of model (5.1) is less than the one of (5.2). On the other hand, if  $\mathbf{z}_i$  has no causal influence on  $\mathbf{z}_j$  the errors are approximately equal. In [74,75] a GC test based on squared sum of residuals is proposed

$$\text{GC}_{i \rightarrow j} = \frac{\sum_{t=1}^T |\omega_t|^2 - \sum_{t=1}^T |\varepsilon_t|^2}{\sum_{t=1}^T |\varepsilon_t|^2} \cdot \frac{T - 2K - 1}{K} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \theta \quad (5.3)$$

where  $T = K - L$  and  $K$  is the time series length. Since both  $\varepsilon_k$  and  $\omega_k$  are Gaussian distributed, the sum of squared residuals follows a central chi-squared distribution and the test (5.3) is then the ratio of chi-squared r.v.'s which results in a  $\mathcal{F}$ -distribution [74]

$$\text{GC}_{i \rightarrow j} \sim \mathcal{F}(L, T - 2L - 1).$$

The test threshold  $\theta$  can be set fixing the false alarm probability. Alternatively, in [88] a useful tool for quantifying the degree of connectivity between two nodes  $i$  and  $j$ , named causal magnitude, is defined as

$$F_{i \rightarrow j} = \log \left( \frac{V(\boldsymbol{\omega})}{V(\boldsymbol{\epsilon})} \right) \quad (5.4)$$

where  $\boldsymbol{\omega} = (\omega_1, \dots, \omega_T)$ ,  $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_T)$  and  $\mathbf{V}(\cdot)$  is the unbiased estimator of the variance.

### 5.2.2 Transfer Entropy

In [80], a model-independent method is proposed to measure the information flow between two random processes by a specific type of conditional mutual information named TE. Considering two time series  $\mathbf{z}_i$  and  $\mathbf{z}_j$ , modeled as random processes, the TE from node  $i$  to node  $j$  can be expressed as

$$\begin{aligned} \text{TE}_{i \rightarrow j}(R, Q) &= \mathcal{I}(z_{j,k}; \mathbf{z}_{i,k-1:k-R}, \mathbf{z}_{j,k-1:k-Q}) \\ &= \mathbb{E} \left\{ \log_2 \frac{p(z_{j,k} | \mathbf{z}_{i,k-1:k-R}, \mathbf{z}_{j,k-1:k-Q})}{p(z_{j,k} | \mathbf{z}_{j,k-1:k-Q})} \right\} \end{aligned} \quad (5.5)$$

where  $\mathbf{z}_i^-$  and  $\mathbf{z}_j^-$  denote the past samples of  $\mathbf{z}_i$  and  $\mathbf{z}_j$  up to time instant  $k$ , respectively. In general, evaluating conditional probability densities requires the knowledge of infinite past samples of  $\mathbf{z}_i$  and  $\mathbf{z}_j$ . However, in this particular application, TE is calculated considering only  $R$  and  $Q$  past samples of  $\mathbf{z}_i$  and  $\mathbf{z}_j$ , respectively. The decision threshold  $\theta$  is obtained by the null distribution of the TE, estimated from an appropriate manipulation of the time series [81], setting a predefined false-alarm probability. Then, the test becomes

$$\text{TE}_{i \rightarrow j} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \theta. \quad (5.6)$$

The flow of information from node  $i$  to node  $j$  might take some time to generate a response, i.e., sending an acknowledgment (ACK). To predict the causal interaction, the lags  $R$  and  $Q$  should be very large and, thus, the algorithm's complexity gets overwhelming. Hence, an additional interaction delay parameter,  $n_0$ , to select the past values of the time series, is proposed in [89]. The definition of TE is then modified as

$$\text{TE}_{i \rightarrow j}(R, Q, n_0) = \mathcal{I}(z_{j,k}; \mathbf{z}_{i,k-n_0-1:k-n_0-R}, \mathbf{z}_{j,k-1:k-Q}). \quad (5.7)$$

Note that the interaction delay has been considered only on the time series

$\mathbf{z}_i$ . This because the useful information on  $z_{j,k}$  has been extracted from its past  $\mathbf{z}_{j,n-1:n-Q}$ , and only the influence of  $\mathbf{z}_i$  need to be investigated.

### Conditional transfer entropy

If for a given node  $i$ , we evaluate the  $\text{TE}_{j \rightarrow i}(R, Q, n_0)$ ,  $\forall j = 1, \dots, N$  with  $j \neq i$  and then apply the binary hypothesis test for each of the pairs  $\{i, j\}$ , we identify the set of possible neighbours of  $i$ . However, TE is a simplified version of CE and, as shown in [45], it tends to overestimate the number of links. For this reason, the set of possible neighbours is tested again with a variant of TE called conditional transfer entropy (CTE), where the effects of all the possible neighbours on the causal inference are taken into account.

If for a given node  $i$ , we evaluate  $\text{TE}_{i \rightarrow j}(R, Q, n_0)$ ,  $j = 1, \dots, N$  with  $j \neq i$ , and then apply the binary hypothesis test for each pair  $\{i, j\}$ , we identify the set of possible neighbours of node  $i$ . Then, to avoid the presence of spurious links, the set of possible neighbors should be tested again with a variant of TE called CTE, where the effects of all the possible neighbors on the causal inference are taken into account. CTE from node  $i$  to  $j$  is defined as

$$\begin{aligned} \text{CTE}_{i \rightarrow j}(R, Q, n_0, g) \\ = \mathcal{I}(z_{j,k}; \mathbf{z}_{i,k-n_0-1:k-n_0-R} | \mathbf{z}_{j,k-1:k-Q}, \mathbf{z}_{g,k-1:k-Q}) \end{aligned} \quad (5.8)$$

where  $g = 1, \dots, N$  with  $g \neq i, j$ . For a complete description of the CTE algorithm and all its details, please refer to [81].

## 5.3 Causality Detection with Machine Learning

The previously described methods compute a test based on the entire time series. We now propose a different approach that uses time-based features to infer the presence of causality via binary classification [90].

### 5.3.1 Feature Extraction

Indeed, in case of traffic flow from node  $i$  to node  $j$ , a packet-ACK time relation is expected to be found, as depicted in Fig. 5.3. The method might also apply to protocols that do not support ACKs. In that case, temporal features are extracted from the inter-transmission time between the end of a packet sent by node  $i$  and the following transmission from node  $j$ . We indicate with  $\tau^{i \rightarrow j}$  the time elapsed between the end of a packet sent by node  $i$  and the beginning of a packet from node  $j$ . If  $a_{i,j} = 1$  the packet transmitted by node  $j$  is likely to be an ACK, but we are not certain. We only suppose that, if the packet transmitted by node  $j$  is an ACK, the statistical features estimated from  $\tau^{i \rightarrow j}$ , i.e., the sample mean, will significantly change with respect to the case in which the packet is not an ACK.

Denoting with  $N_{\text{ta}}$  the number of time-to-acks detected within the observation window,  $T_{\text{ob}}$ , three main relevant features characterize the statistic of the time-to-ack  $\tau^{i \rightarrow j}$ :

- *Sample mean*

$$\mathbf{M}_{\tau^{i \rightarrow j}} = \frac{1}{N_{\text{ta}}} \sum_{p=1}^{N_{\text{ta}}} \tau_p^{i \rightarrow j} \quad (5.9)$$

- *Sample variance*

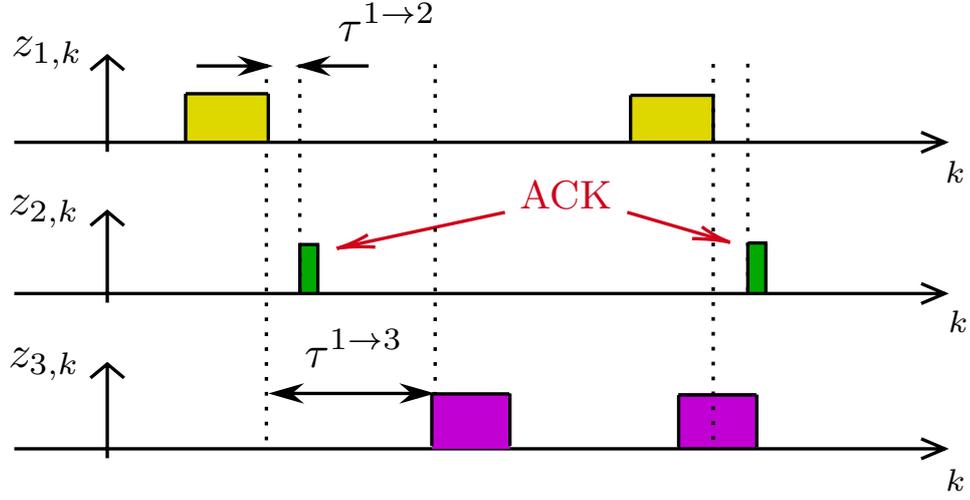
$$\mathbf{V}_{\tau^{i \rightarrow j}} = \frac{1}{N_{\text{ta}} - 1} \sum_{p=1}^{N_{\text{ta}}} (\tau_p^{i \rightarrow j} - \mathbf{M}_{\tau^{i \rightarrow j}})^2 \quad (5.10)$$

- *Kurtosis*

$$\mathbf{K}_{\tau^{i \rightarrow j}} = \frac{\mathbf{m}_4^{i \rightarrow j}}{(\mathbf{m}_2^{i \rightarrow j})^2} \quad (5.11)$$

where  $\mathbf{m}_4^{i \rightarrow j}$  and  $\mathbf{m}_2^{i \rightarrow j}$  are respectively the 4th and the 2nd order moments, estimated from samples as

$$\mathbf{m}_q^{i \rightarrow j} = \frac{1}{N_{\text{ta}}} \sum_{p=1}^{N_{\text{ta}}} (\tau_p^{i \rightarrow j} - \mathbf{M}_{\tau^{i \rightarrow j}})^q. \quad (5.12)$$

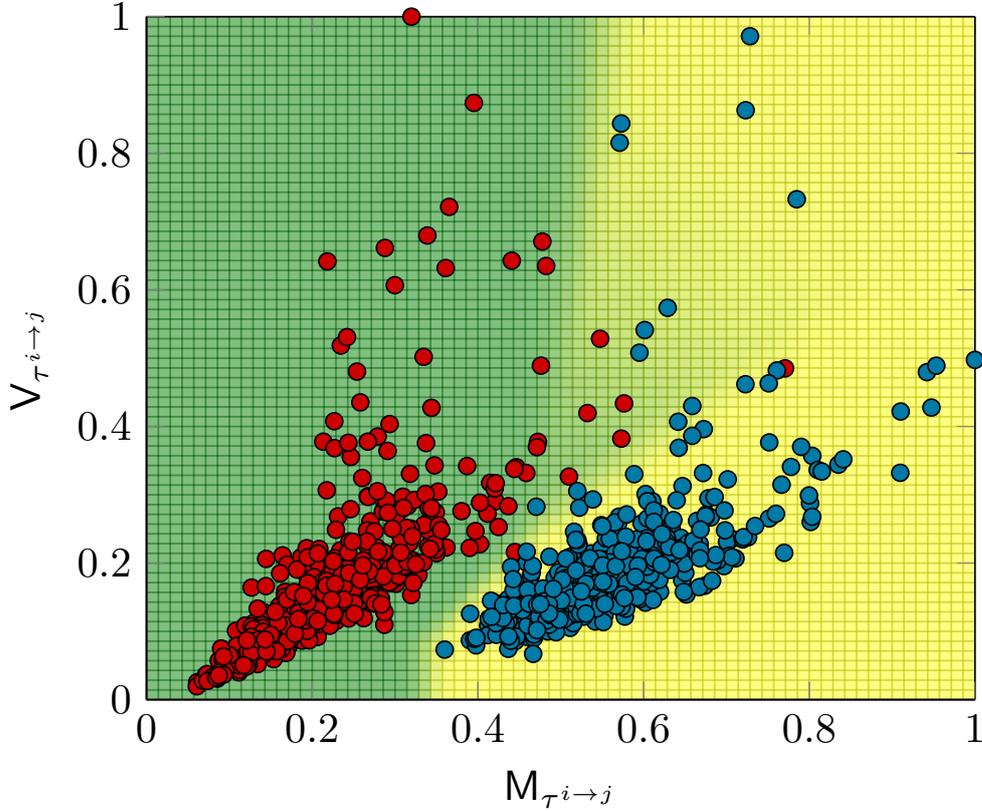


**Figure 5.3:** An example of successful transmission of data packets between nodes  $\mathcal{N}_1$  and  $\mathcal{N}_2$  is shown. In this case node  $\mathcal{N}_1$  is connected to node  $\mathcal{N}_2$  but not to node  $\mathcal{N}_3$ . Notice how the time-to-ack  $\tau^{1 \rightarrow 2}$  and  $\tau^{1 \rightarrow 3}$  differ each other.

The causal magnitude  $F_{i \rightarrow j}$  in (5.4) can be considered as an additional feature to incorporate the benefits of GC for the classification. In the previous example, if the packet transmitted by node  $j$  is an ACK, the statistical features estimated from  $\tau^{i \rightarrow j}$ , e.g., the sample mean, will significantly change with respect to the case in which the packet is not an ACK.

### 5.3.2 Causality Detection

If we consider, e.g., only two features, they can be represented on a plane such as the one reported in Fig. 5.4. Each red and blue point in the figure represents a couple  $\{M_{\tau^{i \rightarrow j}}, V_{\tau^{i \rightarrow j}}\}$  extracted from time series measured at nodes  $i$  and  $j$ . In particular, a red point corresponds to the presence of a directed link from  $i$  to  $j$ , i.e.,  $a_{i,j} = 1$ , while a blue point represents the absence of the link, that is  $a_{i,j} = 0$ . The color gradient shows the decision boundary identified by the classification algorithm. After a proper training phase, a classification algorithm (i.e., NN) can identify a boundary that separates the two classes. Accordingly, this approach needs a preliminary step where the features calculated from time series obtained by real or simulated networks

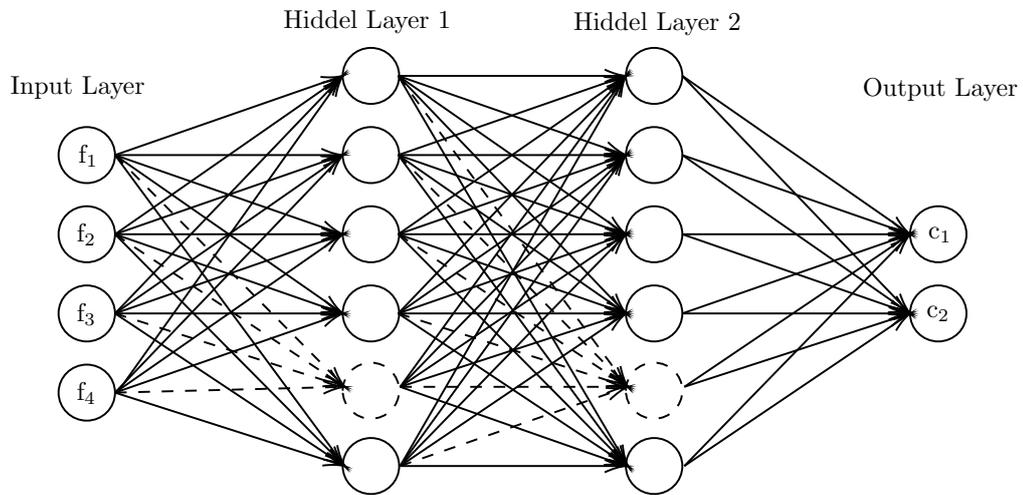


**Figure 5.4:** Example of a two-dimensional normalized features space. In red: points corresponding to the presence of a link ( $a_{i,j} = 1$ ). In blue: points representing couples of nodes that are not connected ( $a_{i,j} = 0$ ). The color gradient shows the decision boundary identified by the classification algorithm.

are collected and used for training. As it is shown in Fig. 5.4 the two groups of points are not linearly separable, therefore a NN has been selected as a proper classification algorithm in this work [91,92].

More features can be used as well, such as higher-order moments (with Kurtosis, we stopped at order four) or cumulants of the time-to-ACK distribution. Also, the causal magnitude  $F_{i \rightarrow j}$  can be considered as an additional feature to incorporate the benefits of GC for the classification. In Fig. 5.5 the input layer has size 4, meaning that the features used as input for the classification are 4:  $M_{\tau^{i \rightarrow j}}$ ,  $V_{\tau^{i \rightarrow j}}$ ,  $K_{\tau^{i \rightarrow j}}$ ,  $F_{i \rightarrow j}$ .

Once the boundary has been found, it is possible to classify new points



**Figure 5.5:** The structure of the 2-hidden-layer NN used to infer the presence/absence of the links.

on-the-fly according to their position on the features space. In this way, we first classify every possible link and then merge all the outcomes to obtain the network topology. Such an approach applies to statistical time-division multiplexing (STDM)-based networks, which encompasses a variety of multiple access algorithms (e.g., CSMA/CA), and appear to be lightweight than known methods under certain conditions.



# Chapter 6

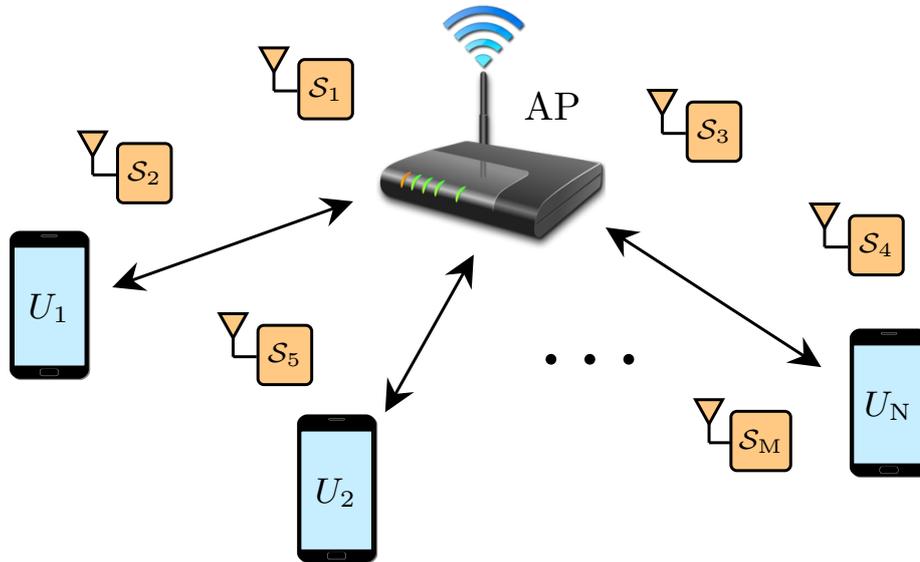
## Traffic Classification

### 6.1 Problem Statement

Here we propose a ML-based approach for traffic classification in wireless networks using low-cost RF sensors, as depicted in Fig. 6.1, where such sensors do not need to be part of the network to perform classification. In particular, the signal unmixed through the BSS is processed to classify the traffic patterns of the users of a wireless network. The problem can be modeled as a multi class (or multinomial) classification, which consists in classifying instances into three or more classes. In this work, three different traffic profiles have been simulated: *video streaming*, *chat*, and *web navigation*. Fig. 6.2 shows an example of the packet profiles generated by the wireless network users for all three classes. It is evident how a time relation between the nodes' transmission can be found and used to discriminate between the three application types. We compare the performance of classifiers, such as SVMs and NNs, and statistical tools, such as PCA and KPCA, to assess their ability to classify these traffic patterns.

#### 6.1.1 Existing Works

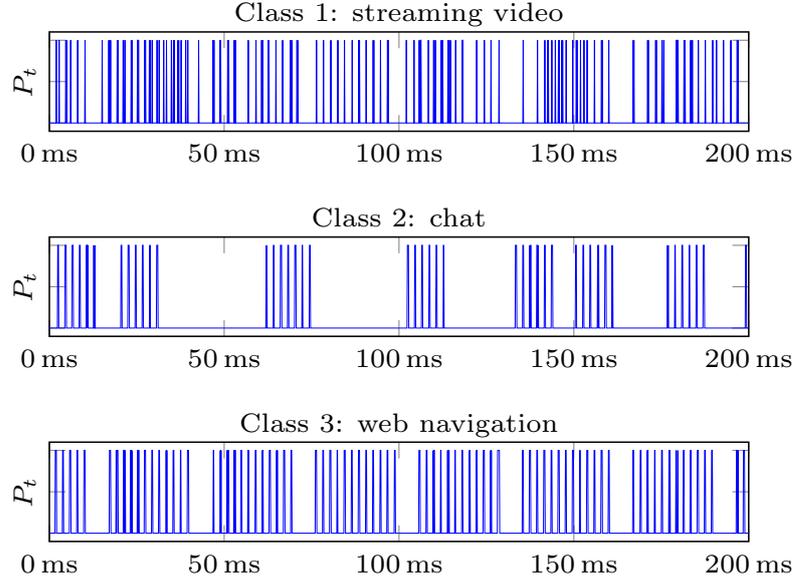
Many approaches and methodologies for traffic classification are proposed in the literature [93, 94]. Such methodologies can be grouped into three main categories [95]. *Port-based classification* is used when the protocols



**Figure 6.1:** System overview: the network of sensors monitoring the traffic patterns among the wireless network nodes.

are assigned to a well-known transport-layer port (i.e., TCP, HTTP). The main issue with this method is that many applications use dynamic port-negotiation mechanisms to guarantee user privacy. *Payload-based classifiers* inspect the content of packets beyond the transport layer headers, looking for features in packet payloads that can distinguish an application protocol from the others. These classifiers are usually used when traffic is not encrypted or enclosed into other application-level protocols. *Statistical classification* analyses statistical attributes, also called *features*, of the received traffic to perform classification through *learning* algorithms [93]. This methodology can be applied to encrypted traffic because the content of packets is never exploited, and it is lightweight in terms of sensing, but it can be less accurate than payload-based classifiers.

While traffic classification in wired networks has been extensively investigated, very few works address the problem in wireless systems, although the emergence of CR technology makes this aspect rather important [95].



**Figure 6.2:** An example of user packet profiles for the three activities considered.

## 6.2 Features Extraction

The traffic profiles depicted in Fig. 6.2 show particular characteristics. Video streaming traffic can be considered a dense stream of packets containing a relatively large volume of data. In contrast, chat traffic can be seen as sparse groups of packets representing the messages sent and received by the user. Web navigation, instead, produces a more variable traffic profile with respect to the other activities. Let us denote with  $N_{\text{ta}}$  the number of inter-arrival times detected within the observation window. Similarly to the topology inference methodology, there are four relevant features which characterize the statistic of packets' inter-arrival time that can be distilled:

- *Sample mean*

$$M_{\tau} = \frac{1}{N_{\text{ta}}} \sum_{k=1}^{N_{\text{ta}}} \tau_k. \quad (6.1)$$

- *Sample variance*

$$V_{\tau} = \frac{1}{N_{\text{ta}} - 1} \sum_{k=1}^{N_{\text{ta}}} (\tau_k - M_{\tau})^2. \quad (6.2)$$

- *Kurtosis*

$$K_\tau = \frac{m_4}{m_2^2} \quad (6.3)$$

where  $m_4$  and  $m_2$  are respectively the 4th and the 2nd order moments, estimated from samples as

$$m_q = \frac{1}{N_{\text{ta}}} \sum_{k=1}^{N_{\text{ta}}} (\tau_k - M_\tau)^q. \quad (6.4)$$

- *Rate of packets*,  $R_p$ , i.e., number of packet arrivals per second.

### 6.3 Survey of ML Classifiers

Let us define the *feature matrix*  $\Phi \in \mathbb{R}^{F \times D}$  where  $D$  is the number of points while  $F$  is the number of features extrapolated for each point, i.e.,  $F = 4$  according to Section 6.2.

The matrix  $\Phi$  is related to the *association matrix*  $\mathbf{t} \in \mathbb{R}^{D \times C}$ , where  $C$  is the number of classes (or categories);  $C = 3$  in the current setting. The element  $t_{dc}$  of  $\mathbf{t}$  is 1 when the  $d$ th observation belongs to the  $c$ th class, otherwise its value is set to  $-1$ .

We now briefly review the algorithms adopted for over-the-air traffic classification: PCA, KPCA, SVM, and NN.

#### PCA

PCA is a widely known algorithm in exploratory data analysis. Considering the  $c$ th class, given the centered training set  $\Phi_c$ , the algorithm remaps the training data from the feature space  $\mathbb{R}^F$  in a subspace  $\mathbb{R}^P$  (where  $P < F$  is the number of principal components selected) that minimizes the information loss between the projected data and the original ones. The best subspace over which to project the data depends on the training set distribution and the number of components selected  $P$ . Iterating this process for all the  $C$  classes, we obtain a set of subspaces, one for each class, where to project the data. For the multi-class classification purpose, we seek to find which

subspaces give a better representation of the test data. For this reason, firstly, we project the test data set over all the subspaces already found. Then, data are remapped to their original space, and the Euclidean distance between the original data and the remapped ones is calculated for each subspace of the set. The class corresponding to the subspace that gives the minimum Euclidean distance is the classifier's output.

## KPCA

This approach takes inspiration from the standard PCA and overcomes the limitation of the linear mapping that corresponds to finding linear boundaries in the original feature space. This constraint represents a severe limitation in many applications and can sharply decrease the classification accuracy. KPCA first maps the data with a non-linear function, then applies the standard PCA to find a linear boundary in the new feature space. Such boundary becomes non-linear, going back to the original feature space. A crucial point in KPCA is selecting a non-linear function that leads to linearly separable data in the new feature space. In the literature, when the data distribution is unknown, the radial basis function (RBF) kernel is often proposed as the right candidate to accomplish this task [96]. Suppose we have a generic point  $\phi_d$  that corresponds to a vector of length  $F$ , we can apply the RBF as follows

$$K_{\phi_{f,d}} = e^{-\gamma \|\phi_d - \phi_{f,d}\|_2^2} \quad \text{with } f = 1, 2, \dots, F \quad (6.5)$$

where  $\gamma$  is a kernel parameter (inversely proportional to the width of the Gaussian function) that must be appropriately set, and  $K_{\phi_{f,d}}$  is the  $f$ th component of the point  $\phi_d$  in the kernel space. Overall the starting vector  $\phi_d$  is mapped in a vector  $\mathbf{K}_{\phi_d}$  of length  $F$ . Applying now the PCA to the new data set obtained remapping all the training points, it is possible to find non-linear boundaries in the starting feature space for a better classification. It is good practice to center the points mapped with the RBF because the mapping in the new feature space could be non-zero mean.

### Support vector machine

The SVM constructs a set of hyperplanes in high-dimensional space that can be used for tasks like classification or regression [90, 92]. Hence, it is a parametric learning algorithm whose error function includes a regularization term as follows:

$$g(\mathbf{w}) = \sum_{d=1}^D \ln \left( 1 + e^{-y_d(\phi_d^T \mathbf{w})} \right) + \omega \|\mathbf{w}\|_2^2 \quad (6.6)$$

where  $\mathbf{w}$  is the vector of the weights of the SVM model, and  $\omega$  is the regularization parameter.

### NN

Considering the case study of this work, the groups of points of the three classes are not linearly separable; therefore, a shallow NN has been chosen as a fourth classification algorithm [91, 97]. In particular, we adopt a 2-hidden-layer feed-forward NN. The well-known *k-fold cross-validation* method has been chosen to avoid overfitting. During the training phase, the network tracks the function described by the features matrix and finds the classification region's boundary. Once the boundaries have been found it is possible to classify new points according to their position on the hyperplane.

# Chapter 7

## Framework Validation

This chapter presents a set of tests that validate the proposed framework for spectrum patrolling and the extraction of wireless network analytics. In particular, Section 7.1 details the simulation parameters that have been used throughout all the proposed tests. Instead, the specific parameters, which are different for each test, are listed in the respective subsections.

### 7.1 Simulation Setup

As a case study, we recreated an IEEE 802.11s ad-hoc network, operating at  $f_0 = 2.412$  GHz, using a simulator developed through the *ns3* platform. The wireless network landscape is a square area of side 10 m. The propagation scenario is characterized by omnidirectional antennas at the nodes and the sensors, path-loss, log-normal shadowing, and thermal noise. The path-loss model is of power-law type with channel gain  $h'_{m,n} = h_0(\frac{d_0}{d_{m,n}})^\nu$  where the path-loss exponent is  $\nu = 3$ , the reference distance is  $d_0 = 1$  m, and  $h_0 = -60.1$  dB [98]. The transmit power of the nodes is  $P_T = 10$  dBm, while the thermal noise power for both nodes and sensors is  $\sigma_N^2 = -93$  dBm. The RF sensors have a  $W = 20$  MHz bandwidth and continuously sense the spectrum, with an ED integration time of  $T_b = 10 \mu\text{s}$ .<sup>1</sup> The shadowing parameter  $\sigma$  is expressed in deciBel as  $\sigma(\text{dB}) = 10 \sigma / \ln 10$ . Two types of packets are

---

<sup>1</sup>Note how the number of degrees of freedom,  $N_{\text{dof}} = 2WT_b = 400$ , is considerably high in this configuration.

**Table 7.1:** Set of parameters used in the tests described in Section 7.2

Parameter Set	$A_0$	$A_1$	$A_2$	$B_0$	$B_1$	$B_2$	$C_0$	$C_1$	$C_2$
$\rho_S$ (nodes/ $m^2$ )	0.3	0.3	0.3	0.2	0.2	0.2	0.1	0.1	0.1
$\sigma_S$ (dB)	0	3	6	0	3	6	0	3	6

present in the target network, data packets with size 1024 Byte, and ACK packets of 112 Byte. Each node has an offered traffic of 1 Mb/s. Regarding the signal processing chain, the loss parameter of the spatial filter, used as a benchmark, is set to  $\eta = 4$ .

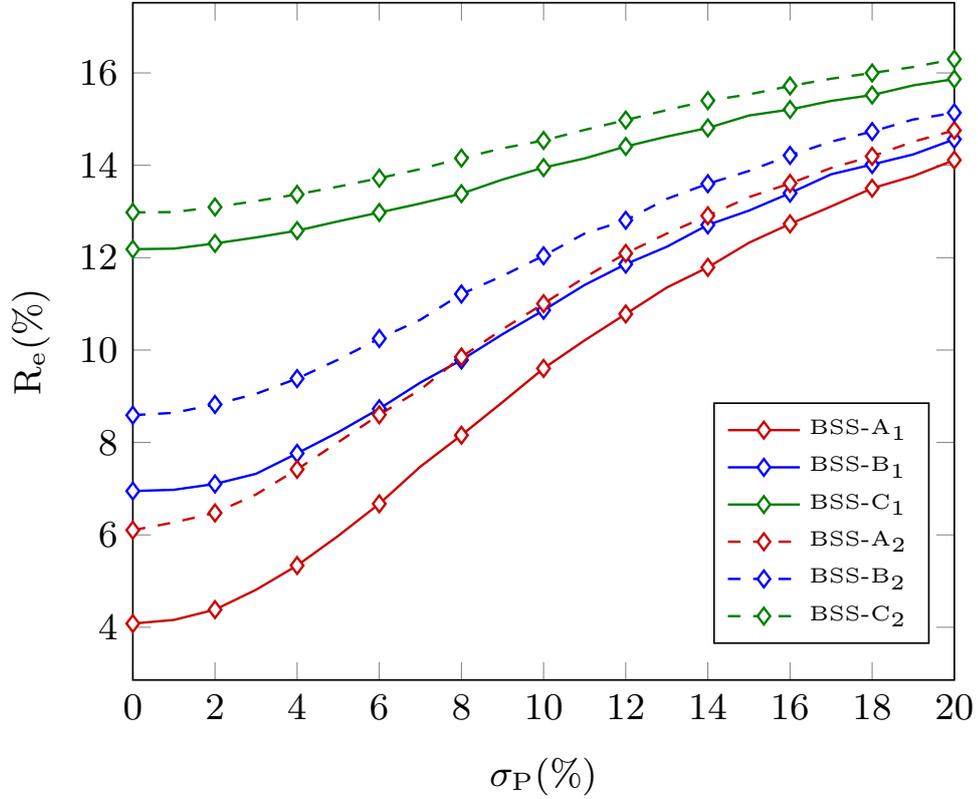
## 7.2 Blind Source Separation

### 7.2.1 Parameter Setting and Figures of Merit

To evaluate the performance of BSS we define the reconstruction error as

$$R_e = \frac{\# \text{ of wrong samples}}{\# \text{ of total samples}} = \frac{\|\mathbf{Z} - \bar{\mathbf{P}}\|_1}{N \cdot K}$$

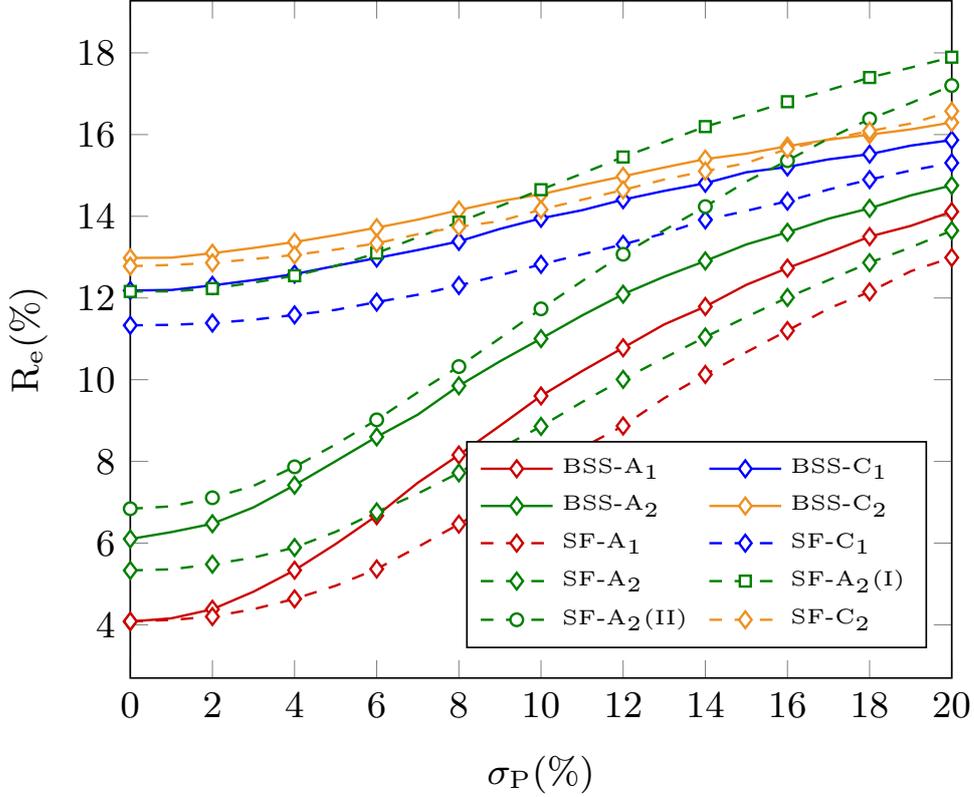
where the matrix  $\bar{\mathbf{P}}$  has elements  $\bar{p}_{n,k} = 1$  if node  $n$  is transmitting in the  $k$ th bin, i.e.,  $p_{n,k} > 0$ , and 0 otherwise. Note that the extraction of the network analytics exploits temporal statistics of the transmitted signals, so the quality of the reconstructed traffic profiles needs not to account for the recovered transmit power error. Hence,  $\bar{\mathbf{P}}$  can be interpreted as normalization of  $\mathbf{P}$  to discard irrelevant amplitude-related information. The accuracy of the time series reconstructed is degraded by noise and shadowing, while the node-source association could be affected by uncertainties on node positions. We model position uncertainty as a Gaussian distributed r.v. with standard deviation  $\sigma_P$  added to each node's coordinates. To characterize the impact of the number of sensors on the performance of the source separation methods, we define the density of sensors  $\rho_S$ , as the number of sensors per square meter. The nine parameters configurations used in this test are summarized in Table 7.1 [99].



**Figure 7.1:** BSS algorithm performance varying the standard deviation  $\sigma_P$  of the location uncertainty, the shadowing parameter  $\sigma_S$  and the density  $\rho_S$ .

## 7.2.2 Reconstruction Error vs. Position Uncertainty

In Fig. 7.1, the BSS algorithm has been tested varying the standard deviation  $\sigma_P$  (%), defined as percentage of the side of the landscape, the shadowing parameter  $\sigma_S$ , and the density  $\rho_S$ . The figure depicts how  $R_e$  increases when  $\sigma_P$  gets higher, even at relatively high density, i.e.,  $\rho_S = 0.3$ . Moreover, the curves translate upward when the shadowing intensity increases, reaching an error  $R_e = 16\%$  with  $\sigma_S = 6$  dB and  $\sigma_P = 20\%$ . In Fig. 7.2 the performance of BSS is compared to the SF benchmark method described in Section 3.5. The figure shows how the filtering parameter  $\eta$  strongly influences SF performance. It outperforms the BSS algorithm in many cases, but requires an experimental tuning that might not always be possible. Moreover, in the presence of strong shadowing, the performance of this method rapidly



**Figure 7.2:** Performance of BSS compared to the SF parametric method varying  $\rho_S$  and  $\sigma_S$  according to Table 7.1.

degrades.

## 7.3 Multiple Transmitter Localization

### 7.3.1 Parameter Setting and Figures of Merit

Regarding the node counting process, the eigenvalue selection parameter is set to  $w = 10^{-4}$  via scree plot [42], which ensures the best accuracy in the specific scenario, while in the RSS extraction phase, the threshold parameter is set to  $q = 0.7$ . The observation time is  $T_{\text{ob}} = 1$  s, corresponding to  $K = 100 \cdot 10^3$  power samples. We can estimate and update the target nodes position every  $T_{\text{ob}} = 1$  s. For grid-based search in the MLE algorithm, the area is split into equal square cells of side 0.01 m. The parameters of the

**Table 7.2:** Probability of correct estimation of the number of transmitters,  $\widehat{N}$ , as a function of  $\rho$  and  $\sigma$ .

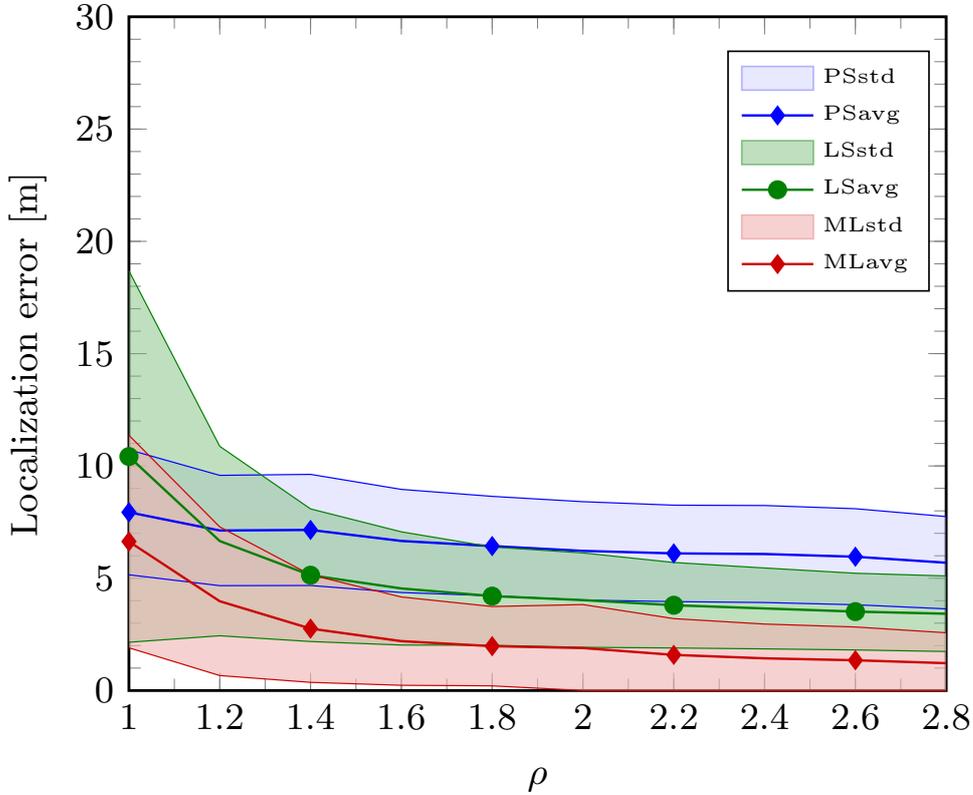
$\frac{\rho \rightarrow}{\sigma(\text{dB}) \downarrow}$	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8
1	0.83	0.98	0.98	0.99	1	1	1	1	1	1
3	0.78	0.97	0.98	0.99	0.99	1	1	1	1	1
5	0.69	0.91	0.94	0.95	0.99	1	1	1	1	1
7	0.57	0.73	0.87	0.91	0.94	0.97	0.98	0.99	1	1
9	0.37	0.48	0.72	0.75	0.83	0.83	0.86	0.88	0.94	0.95

particle simulation algorithm where set to  $h = 0.1$  and  $N_{\text{iterations}} = 500$  according to [56].

All the results reported in this section are extracted by the simulations of  $N_{\text{net}} = 2000$  different wireless networks where the position of the nodes and the sensors is random within the area with the only constraint that the nodes and sensors are spaced apart by at least 5 m. Fig. 4.1 shows an example of a simulation scenario with a network of  $N = 5$  nodes and  $M = 8$  sensors. The clouds of grey circles are position estimates of the nodes at different Monte Carlo (MC) instances using the proposed methodology with MLE location estimation.

For each MC run, the localization error, defined as the Euclidean distance between the actual target position and the estimated one, and its RMSE have been recorded.<sup>2</sup> Since both the sensors' and the nodes' spatial configuration significantly influence the position estimate, the localization error may deviate considerably from its average. Therefore, besides the average, 80-th and 20-th percentiles, standard deviation, and RMSE of the location error are also considered. The number of MC iterations for each network realization is 1000.

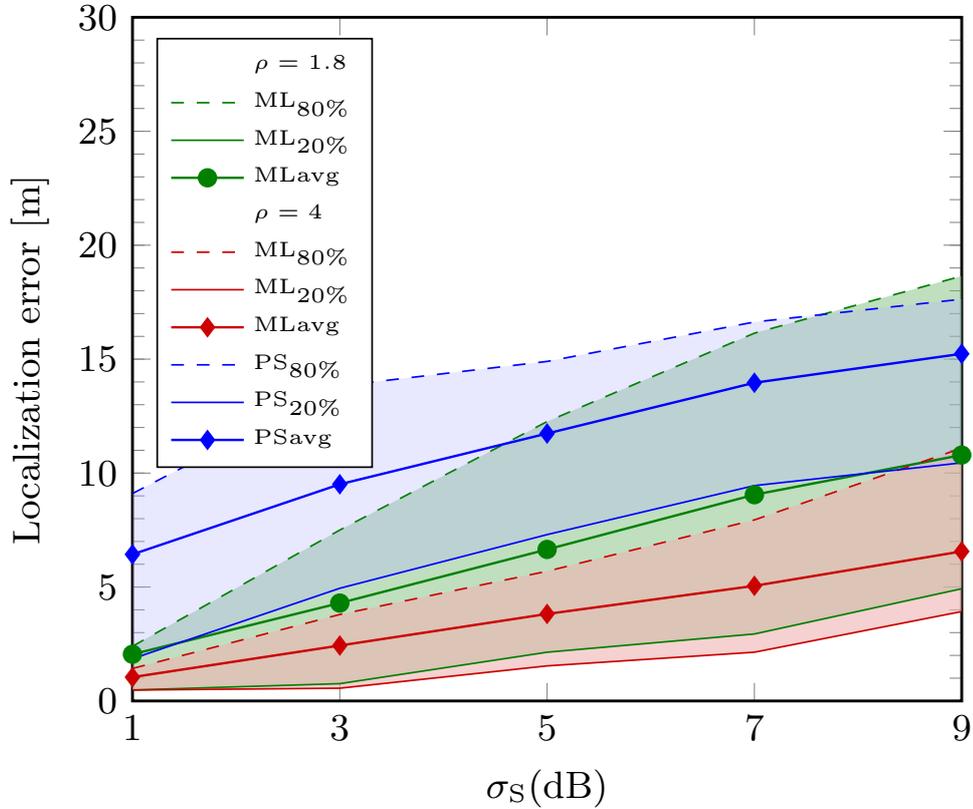
<sup>2</sup>The RMSE for each MC simulation has been calculated over 200 independent realizations of shadowing.



**Figure 7.3:** Comparison between the mean and the standard deviation of the localization error of the three algorithms (BSS-LS, BSS-MLE and particle simulation (denoted as PS)) varying the ratio between the number of sensors and nodes,  $\rho$ , in a mild shadowing regime with  $\sigma_S = 1$  dB.

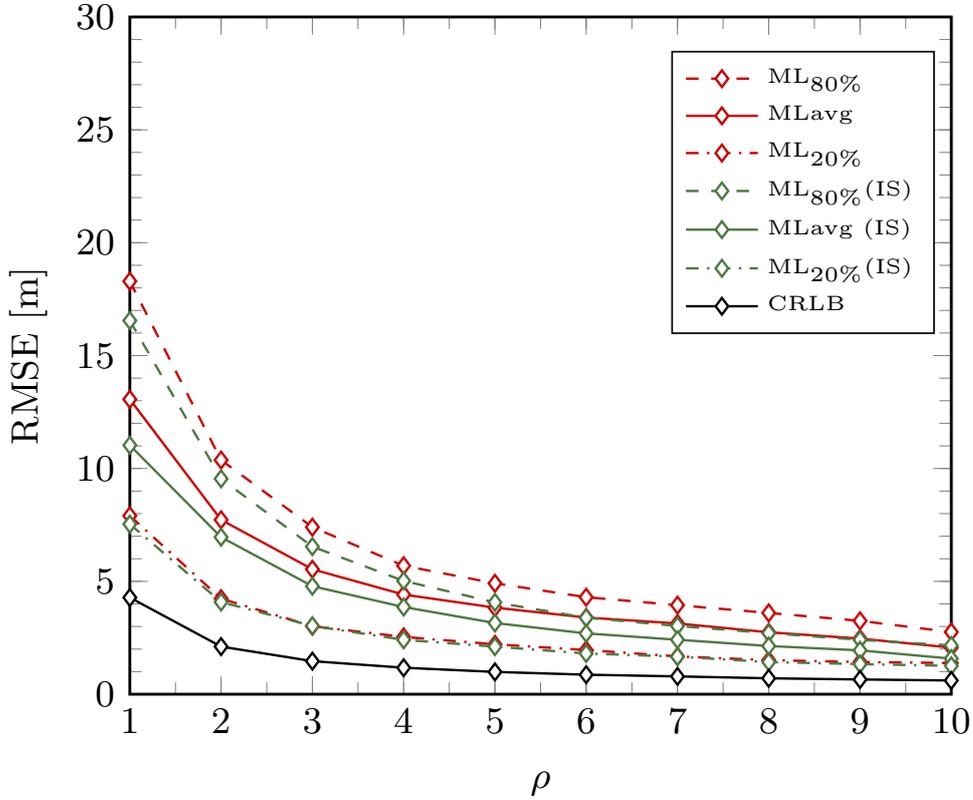
### 7.3.2 Number of Sensors and Shadowing

Our purpose is to study the effect of the number of sensors and the shadowing parameter on the localization performance. To make the results more understandable, we define the ratio  $\rho = M/N$ . In particular, in the simulations  $N \in \{3, \dots, 10\}$  and  $M$  is selected accordingly. The performance of the node counting is shown in Table 7.2. In particular, the table reports the probability of correct counting, calculated as the ratio between the number of MC instances where the number of nodes is estimated correctly and the total number of MC instances, varying  $\rho$  and  $\sigma_S$ . As expected, the accuracy of the estimation degrades when the shadowing intensity  $\sigma_S$  increases, but such degradation can be counteracted by increasing the number of sensors.



**Figure 7.4:** The 20-th and 80-th percentiles and the mean localization error, for both BSS-MLE and particle simulation, varying the shadowing parameter  $\sigma_S$ (dB) for  $\rho = 1.8$  and  $\rho = 4$ .

The following results on localization performance are obtained considering 1000 MC instances with the correct node count. Fig. 7.3 shows the average value and the standard deviation of the localization error for the two proposed localization approaches and the solution based on particle simulation presented in [56], varying the ratio  $\rho$ . As it can be evinced, when  $\rho$  increases, the error decreases, showing that a more significant number of sensors positively influences the localization performance. Notably, the particle simulation algorithm requires a larger  $\rho$  to reach the same performance of the BSS-MLE approach. Moreover, Fig 7.4 shows how increasing the shadowing parameter  $\sigma_S$ , the quality of the location estimation degrades significantly. Is it also shown how the ML approach can compensate for the error due to



**Figure 7.5:** The 20-th and 80-th percentiles and the RMSE of BSS-MLE localization with the proposed BSS compared with the ideal BSS (denoted as IS) varying  $\rho$  with  $\sigma = 5$  dB. The CRLB is used as a benchmark to assess the asymptotical improvement of the localization performance.

the presence of strong shadowing with a further increase in  $\rho$ . Considering a mild shadowing regime with  $\sigma_S = 1$  dB, and  $\rho = 4$ , the average localization error drops to 1 m when the proposed methodology is combined with MLE. Instead, considering a strong shadowing scenario with  $\sigma_S = 7$  dB, the error reaches 9 m with  $\rho = 1.8$  and 5 m with  $\rho = 4$ . In general, the BSS-MLE approach performs better in all the scenarios, proving to be less sensitive to shadowing with respect to the BSS-LS and the particle simulation approach, and presenting an acceptable error for a RSS-based localization methodology.

It is also important to note that increasing  $\rho$  and decreasing  $\sigma_S$  the performance of the BSS improves with benefits on the localization step. As proof of this behavior, Fig. 7.5 compares the performance with the case of

ideal separation (IS), i.e., considering a hypothetical BSS that perfectly reconstructs the transmitted power profiles. As can be noticed, increasing  $\rho$ , the error introduced by the BSS decreases, and the performance obtained coincides with the ideal one. Moreover, in Fig. 7.5 the performance of the BSS-MLE approach is compared to the CRLB (4.18). As expected, the proposed solution tends to approach the CRLB as  $\rho$  increases. For example, for  $N = 1$  and  $\rho = 50$  the RMSE deviates from the CRLB by 0.07 m.

### 7.3.3 Computational Complexity Analysis

In [56] the authors state that the computational complexity of the particle simulation algorithm, considering that each iteration involves a set of simple operations in the order of  $\mathcal{O}(NM)$ , is given mainly by the number of iterations and can be expressed as  $\mathcal{O}(NMN_{\text{iterations}})$ .

Similarly, our novel methodology considers the MLE, which involves a set of simple operations in the order of  $\mathcal{O}(NM)$ . These operations are repeated for each point of the grid, so the complexity of the algorithm results  $\mathcal{O}(NMN_{\text{grid}})$ , where  $N_{\text{grid}}$  is the number of grid points.

Thus, the complexities of the two algorithms are comparable despite presenting different localization performances.

## 7.4 Topology Inference

In this section, we present several tests to evaluate the performance of the whole processing chain, the impact of channel impairments on BSS, and compare the state-of-the-art solutions in topology inference with the NN-based method.

### 7.4.1 Parameter Settings and Figures of Merit

Regarding the signal processing chain, the excision filter threshold,  $\zeta_n$ , is set as in (3.28) with  $q = 0.7$ , and the termination parameter of the F-ICA is set to  $\epsilon_t = 10^{-5}$ . The results presented in this section are obtained from the data

extracted by the simulations of  $M_{\text{top}} = 100$  different mesh topologies, such as the one depicted in Fig. 7.6. Then, 100 Monte Carlo trials are performed for each topology to change the nodes and sensors' position randomly.

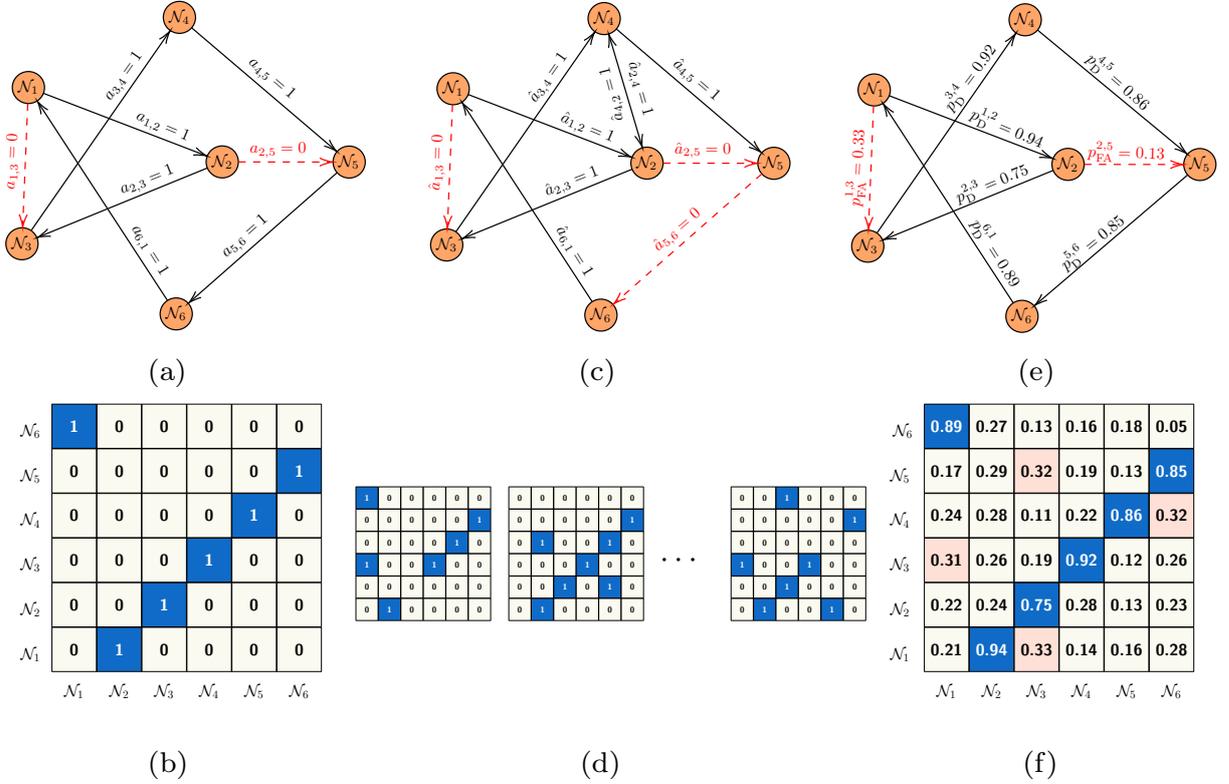
In a real wireless network, the adjacency matrix  $\mathbf{A}$  is sparse, hence the number of links to be detected is much lower than the number of possible connections. Thus, the standard non-weighted metrics (i.e., accuracy) are not suitable for evaluating topology inference performance. Therefore, we adopt the detection probability (or recall),  $p_D^{i,j}$ , and the false alarm probability (or false positive rate),  $p_{\text{FA}}^{i,j}$ , of the directed link from node  $i$  to node  $j$ , defined as

$$p_D^{i,j} = \mathbb{P}\{\hat{a}_{i,j} = 1 | a_{i,j} = 1\}$$

$$p_{\text{FA}}^{i,j} = \mathbb{P}\{\hat{a}_{i,j} = 1 | a_{i,j} = 0\}.$$

In Fig. 7.6, we show the detection and false alarm probabilities for some of the links of the network, using our NN-based approach, estimated from the results of Monte Carlo trials in which the position of the nodes vary inside the landscape, but the network maintain the same logical topology. Moreover, the true adjacency matrix  $\mathbf{A}$ , some of the adjacency matrices  $\hat{\mathbf{A}}$  estimated during the Monte Carlo trials, and the matrix summarizing  $p_D^{i,j}$  and  $p_{\text{FA}}^{i,j}$  for each possible link of the network, are shown. In the following,  $p_D$  and  $p_{\text{FA}}$  are, respectively, the detection and false alarm probabilities averaged over all the network links to summarize the topology inference performance.

The algorithm used for classification is a 2-hidden-layer feed-forward NN with 40 neurons in the first hidden layer and 10 in the second one. All the layers are fully connected, and the activation functions are ReLU for the hidden layers and softmax for the output layer. The considered features are the mean, variance, kurtosis, and GCs causal magnitude. Thus, the input layer has a size of 4. The network is trained with the features extracted by the links of 70 different simulated topologies for 5000 epochs (iterations of the stochastic gradient descent algorithm) with an initial learning rate of 0.1. The learning rate decreases by a factor of 10 after 3000 epochs [91]. A  $k$ -fold cross-validation is performed to avoid overfitting with a validation set



**Figure 7.6:** (a) The directed graph representing the real topology of the network, whose adjacency matrix  $\mathbf{A}$  is shown in (b); (c) and (d) show examples of the inferred topology graph and adjacency matrices  $\hat{\mathbf{A}}$  estimated over the Monte Carlo trials of the same topology depicted in (a) while varying the position of the nodes and the sensors in the landscape; (e) visualization of the performance where  $p_D$  and  $p_{FA}$  indicates the probability of detection and false alarm of the directed link from node  $i$  to node  $j$ , calculated over 100 Monte Carlo trials; the corresponding matrix of  $p_D$  and  $p_{FA}$  is shown in (f).

composed by the features extracted by the links of 30 different topologies. The simulated configurations used in training and testing differ in the number of connections, the number of nodes, their position (i.e., network topology), and the position of the sensors. For the  $\mathcal{F}$ -test of GC a time lag  $L = 4$  is set according to the Akaike information criterion (AIC) [41, 100], while for the CTE the parameters  $R = 2$ ,  $Q = 1$ , and  $n_0 = 3$ , are chosen. For the decision threshold, the false alarm probability is set to  $10^{-2}$  for both algorithms.

The inferred topologies have to be considered instantaneous, i.e., the topology of the network in a time horizon confined by the observation time,

$T_{\text{ob}} = 1$  s in this case, which corresponds to  $K = 100 \cdot 10^3$  samples.<sup>3</sup> This way, the proposed framework can capture the network’s dynamical behavior, including nodes that join and leave the system.

### 7.4.2 Topology Inference and Number of Nodes

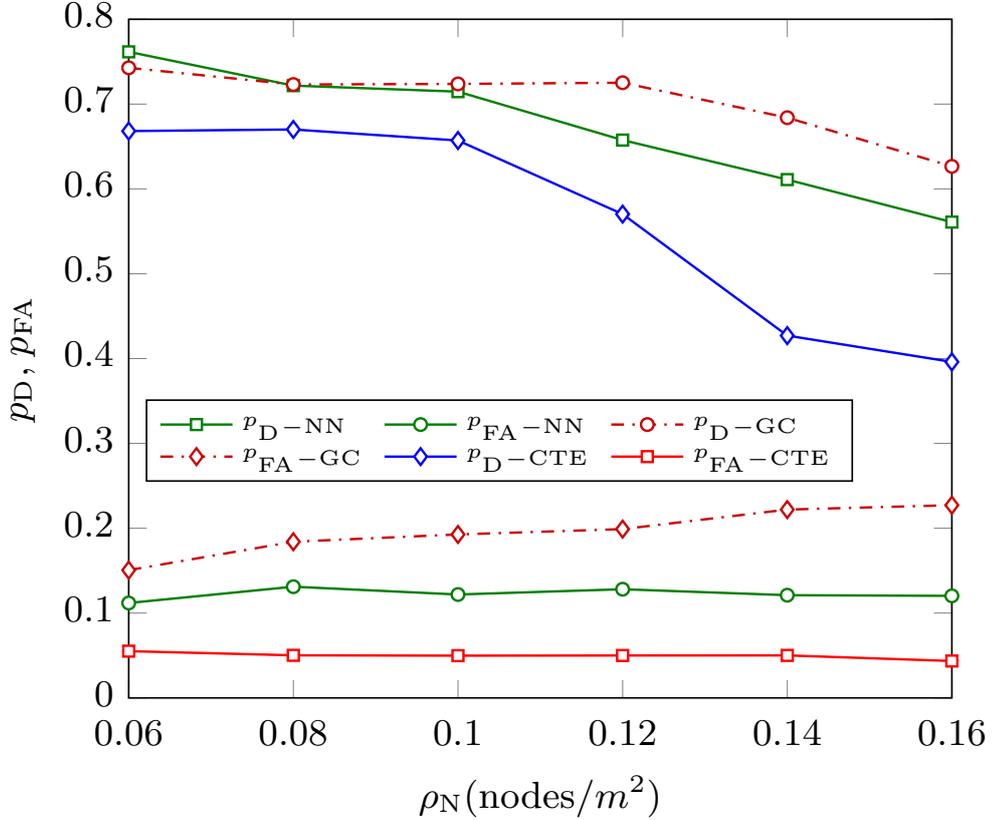
After BSS, the time series are processed to extract the topology information. In this section, the state-of-the-art methods for topology inference described in Chapter 5 are compared varying the density of wireless nodes per square meter,  $\rho_N$ . In this test, the BSS was performed with  $\rho_S = 0.3$  sensors/m<sup>2</sup>,  $\sigma_S = 3$  dB and  $\sigma_P = 0$ . Note that increasing the number of nodes in the landscape leads to an increase in collision probability, which results in network congestion. The NN has been trained only once on the data captured with  $\rho_N = 0.06$  nodes/m<sup>2</sup>. A variation on the density of nodes  $\rho_N$  affects the topology inference when it significantly deviates from the density considered for the training. As depicted in Fig. 7.7,  $p_D$  for NN is comparable with that of GC when the density of nodes is close to the one used for training. However, when the nodes’ density doubles compared to that considered for the training, GC outperforms the NN. On the contrary, when considering  $p_{\text{FA}}$  the NN is better than GC regardless of the density of nodes. As far as CTE is concerned, it presents the lower  $p_{\text{FA}}$ , but the  $p_D$  is lower than the other methods. Therefore, the error in the reconstruction impacts more CTE than the other approaches.

### 7.4.3 Impact of shadowing

On this point, it is important to study the accuracy of the algorithms in scenarios with different propagation characteristics. Fig. 7.8 shows how increasing  $\sigma_S$  degrades the accuracy of the algorithms, as expected. In this case, we set the density of sensors  $\rho_S = 0.3$  sensors/m<sup>2</sup>, the density of nodes  $\rho_N = 0.06$  nodes/m<sup>2</sup>, and  $\sigma_P = 0$ . Even in this scenario, CTE presents a false

---

<sup>3</sup>The topology of the network can be obtained by collecting several instantaneous topologies and mixing all the estimations to have a topology representation on a broader time horizon, as suggested in [74]

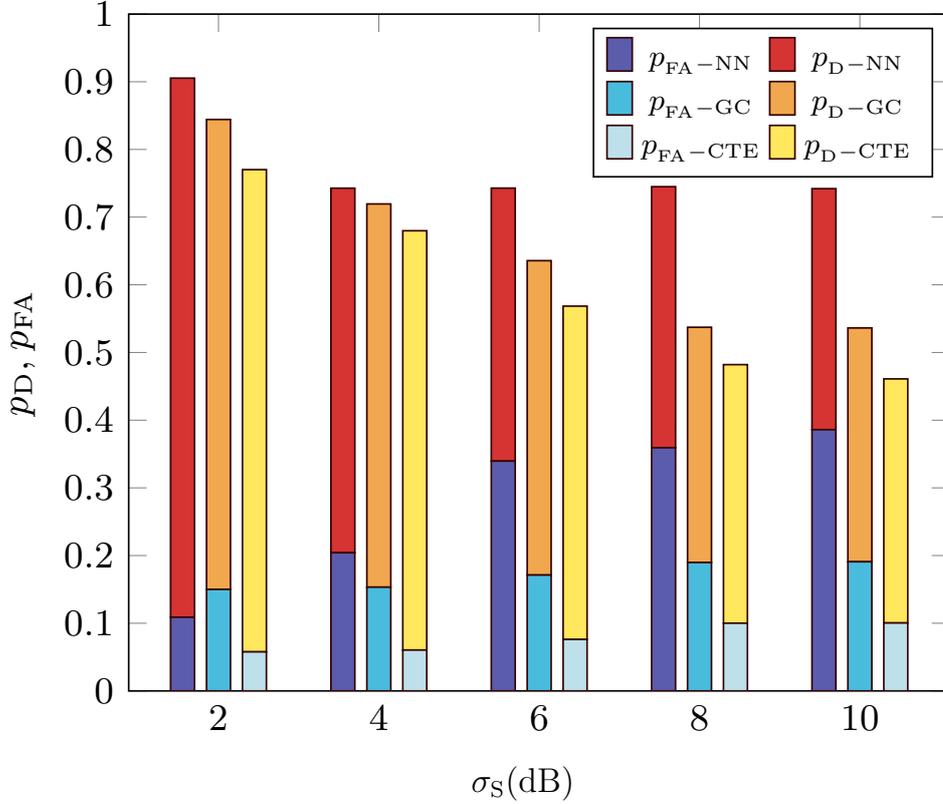


**Figure 7.7:**  $p_D$  and  $p_{FA}$  of the topology inference algorithms as a function of the density of nodes  $\rho_N$  for  $\rho_S = 0.3$  sensors/m<sup>2</sup> and  $\sigma_S = 3$  dB.

alarm rate lower than the other methods, but  $p_D$  is still the lowest. Furthermore, even if the NN outperforms the other methods for low  $\sigma_S$ , increasing the shadowing intensity results in a substantial increment of  $p_{FA}$ .

#### 7.4.4 Impact of nodes mobility

In this test, the effect of the nodes' mobility on the performance of the topology inference is investigated. The mobility model chosen is the Random Walk [101]; within an observation window  $T_{ob}$ , each node moves along a random direction with speed  $v$ . In Fig. 7.9, the performance of the topology inference varying the speed of the nodes is shown. In particular, we set  $v = 2, 10, 20$  m/s to simulate human walking, a slow vehicle (i.e., low-altitude

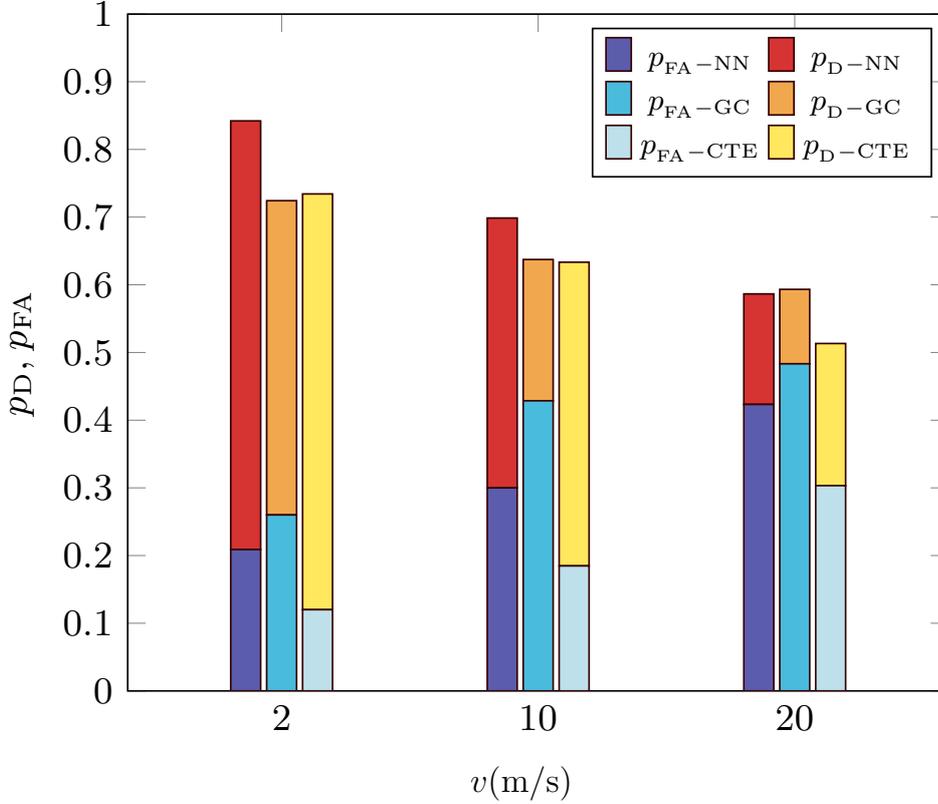


**Figure 7.8:**  $p_D$  and  $p_{FA}$  of the topology inference algorithms as a function of the shadowing parameter  $\sigma_S$  (dB) for  $\rho_S = 0.3$  sensors/m<sup>2</sup>.

UAV) and a fast vehicle, respectively. As the figure highlights, the topology inference is strongly affected by the network nodes' mobility. More specifically, in case of  $v = 2$  m/s the performance is preserved. With  $v = 10$  m/s the inference is degraded, with a detection probability reduced to 70%, while in case of  $v = 20$  m/s, topology inference is compromised. This is due to the inability of BSS to reconstruct the power profile transmitted by the nodes correctly.

#### 7.4.5 Computational Complexity Analysis

This section discusses the computational complexity of the topology inference algorithms as a function of the number of network nodes.



**Figure 7.9:**  $p_D$  and  $p_{FA}$  of the topology inference algorithms in the parameter setting  $A_1$  varying the velocity  $v$  of the wireless network nodes.

- **Granger causality.** A linear regression like (5.1) with  $K$  data points and  $2L$  parameters has complexity  $\mathcal{O}(K4L^2 + 8L^3)$ . Similarly, including the linear regression in (5.2), and considering that  $L^3 \ll K$  and that there are  $N^2 - N$  couples of nodes in the network, the overall complexity is  $\mathcal{O}(N^2K4L^2)$ .
- **NN-based method.** Since the training phase can be executed offline, we account for only the forward propagation in the complexity of the NN. The number of operations strictly depends on the number of neurons and layers and can be treated as a constant  $B$ . Thus, the complexity of the NN is  $\mathcal{O}(N^2B)$ . The forward propagation is preceded by the feature extraction, whose complexity is dominated by the most computational expensive feature to extract, the causal magnitude — for

this reason, considering that  $L^2 \ll B$ , the complete ML-based method has overall complexity  $\mathcal{O}(N^2KB)$ .

- **Conditional transfer entropy.** The complexity is  $\mathcal{O}(N^2KC_1)$  and  $\mathcal{O}(N^2KC_2)$  for the two steps of the algorithm, respectively [81, 102].  $C_1$  and  $C_2$  are two constants that take into account the operations for the choice of the interaction delay and the number of bootstraps iterations in both steps. Combining the two steps, the overall complexity is  $\mathcal{O}(N^2K(C_1 + C_2))$ .

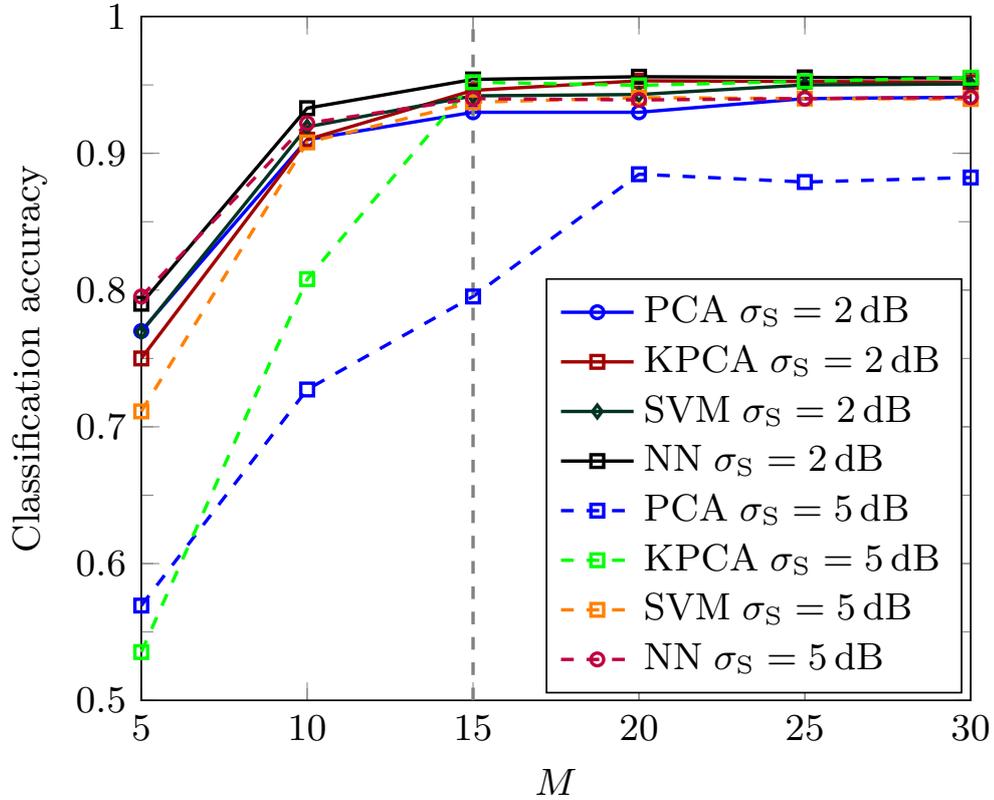
Although the three algorithms have the same complexity trends,  $\mathcal{O}(N^2K)$ , considering a typical range of values for  $N$  in a practical scenario, the time complexity of the NN results considerably lower than the CTE. In fact, since in general  $B \ll C_1 + C_2$ , although they are constant factors, their values can differ by several orders of magnitude, so they are relevant for comparing the algorithms. This means that, in cases similar to those analyzed in this section, the impact of such constant factors is not negligible. To provide a qualitative example, for  $N = 6$ , inferring the complete topology requires an average execution time  $t_{GC} = 8.09$  s for GC, and  $t_{NN} = 8.13$  s for a NN that includes the GCs causal magnitude as a feature. In the same setting, the execution of CTE requires  $t_{CTE} = 117.26$  s.

## 7.5 Traffic Classification

This section presents several tests performed to compare the classification algorithms and reveal when a RF-based traffic classification is possible with satisfactory performance.

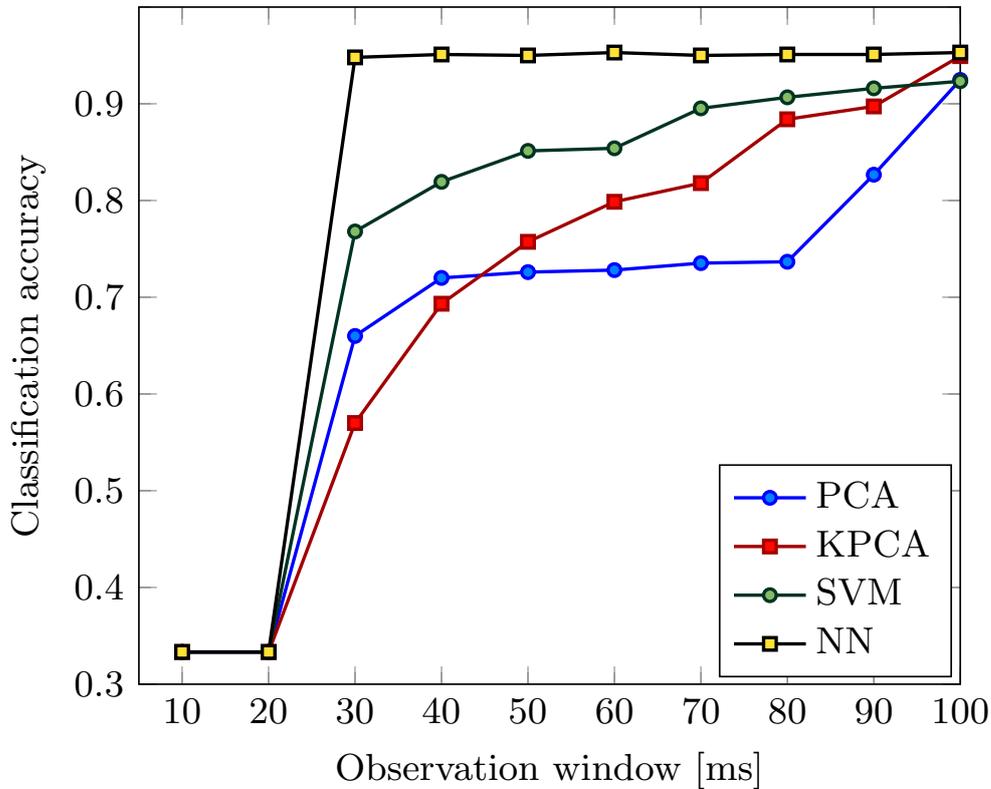
### 7.5.1 Parameter Settings and Figures of Merit

As a figure of merit, we define the accuracy as the number of traffic streams correctly classified over all the tested ones. Since the number of test points for each class is the same, this figure of merit perfectly suits this case study. The wireless network under test is based on the IEEE 802.11n standard and



**Figure 7.10:** Accuracy varying the number of sensors  $M$  with an observation window of 1 s.

is composed of an access point (AP) and  $N = 3$  devices, all randomly placed in the area. The number of classes considered is  $C = 3$ , corresponding to three different patterns of activities generated by the users: *web navigation*, *video stream* and *chat*. The results presented in this section are obtained from NS-3 simulations of 100 different scenarios. More precisely, in each scenario, the position of the nodes, the position of the sensors, and the shadowing are chosen randomly, while each user generates its traffic profile according to one of the three patterns (an example is reported in Fig. 6.2). The excision filter threshold  $\zeta$  is set with  $q = 0.9$ . For the PCA algorithm the number of components is set to  $P = 1$ , while for KPCA is  $P = 3$  and  $\gamma = 30$ . The SVM parameter  $\omega$  is set to 0.1. The NN has 40 nodes in the first hidden layer and 20 in the second one. All the layers are fully connected, and the activation



**Figure 7.11:** Accuracy as a function of the observation window duration for  $M = 15$  sensors.

functions are ReLU for the hidden layers and softmax for the output layer. The network is trained with 1500 points for 2500 epochs (iterations of the stochastic gradient descent algorithm) with an initial learning rate of  $10^{-4}$ . The learning rate decreases by a factor 10 after 2000 epochs.

### 7.5.2 Accuracy vs. number of RF sensors

In this test, we studied the performance of the classifiers as a function of the number of RF sensors distributed in the landscape. The shadowing parameter is set to  $\sigma_S = 2$  dB and  $\sigma_S = 5$  dB, respectively. As expected, the reconstruction error of the BSS increases when a low number of sensors is used. In Fig. 7.10, it is shown how the quality of the classification of the algorithms falls as  $M$  drops below 15 sensors. Note that, the performance

of the PCA results less effective than the other algorithms. In particular, Fig. 7.10 shows that the NN is the most suitable classifier in this scenario. According to the next section, the observation window is set to 1 s.

### 7.5.3 Accuracy vs. observation window

This test aims to find a proper acquisition window duration to guarantee that the algorithm reaches the maximum achievable accuracy. The number of sensors is  $M = 15$ , and the shadowing parameter is  $\sigma_s = 2$  dB. With this aim, Fig. 7.11 shows how the accuracy of the classification algorithms depends on the window width. As expected, if the capture window is too short (i.e., 20 ms), the accuracy degrades significantly. This behavior is related to the time scale for which the features selected are meaningful. Moreover, the figure shows that the NN outperforms the other algorithms even with a short observation window (i.e., 30 ms). This is probably due to the different training procedures.



## Chapter 8

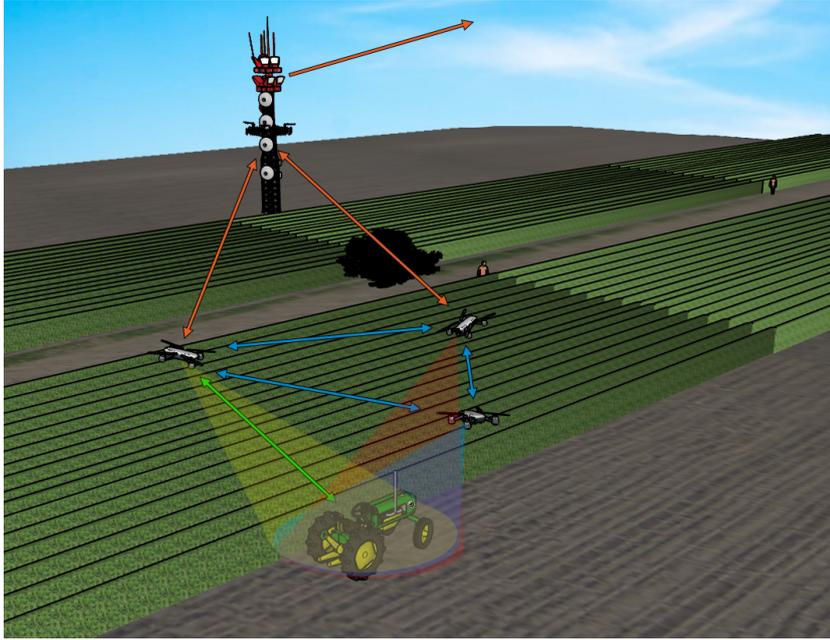
# Reinforcement Learning-Based UAVs Patrolling

This chapter aims to propose an automatic navigation system for a spectrum patrol composed of a fleet of UAVs. The proposed solution allows the fleet to navigate a scenario and find the best spatial configuration to localize a wireless transmitter [103,104]. An illustration of this scenario is proposed in Fig. 8.1.

The UAVs have to localize and track the transmitter, navigating through the scenario in complete autonomy. The drones can benefit from the extensive use of ML algorithms to accomplish these tasks, especially reinforcement learning (RL) [9]. The patrol can use this system in synergy with the network analytics tool to localize a malicious user (i.e., jammer) and secure a wireless network.

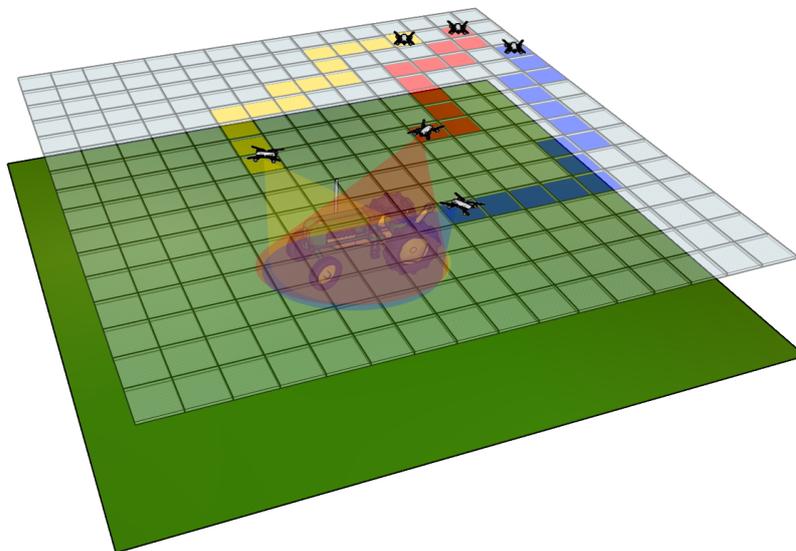
### 8.1 Problem Statement

Let us consider a swarm of  $N$  UAVs and a target wireless transmitter (i.e., a malicious user) moving inside a squared field of side  $L$ , as depicted in Fig. 8.2. The area is represented as a grid, in which cells are  $1 \times 1$  m<sup>2</sup> width. Each UAV, or agent, occupies a grid cell and can move one cell north, south, east, west, or remain in its current position for a total of  $N_a = 5$  possible actions.



**Figure 8.1:** Example of wireless network setup with  $N = 3$  UAVs that localize a malicious transmitter (mounted on a rover) and forward the information to a base station (BS) (i.e., the regulator).

The cell occupied by an agent at the  $t$ -th time instant defines its state  $s_t$ . An action brings the agents from the state  $s_t$  to  $s_{t+1}$ . The number of possible states  $N_s$  is the number of grid cells and varies according to its dimension. An action is not allowed if it would cause a collision between two agents or between an agent and an obstacle in the environment. Let us suppose that the UAVs can estimate the transmitter's position through a simple RSS-based method. We aim to develop a RL-based tool that allows the agents to find the optimal placement to perform localization of the transmitter. The well-known multi-agent Q-learning algorithm is used to let the agents explore the space and find the best spatial configuration to perform localization [105]. Each time an agent makes a move, it is rewarded with a prize depending on the transmitter's estimated distance and the geometric dilution of precision (GDOP). In Section 8.2, the development of the RL algorithm is described in detail.



**Figure 8.2:** An example of grid representation of the environment where the agents can move, searching for the optimal spatial configuration to localize the transmitter (rover).

## 8.2 Multi-Agent Q-Learning

Q-learning is an off-policy temporal difference look-up table-based RL algorithm. Tabular methods are the simplest but most effective forms of RL algorithms, where states and actions are collected in a large matrix or table [105]. Although the computational burden due to the look-up table is remarkable, table-based methods reach very accurate solutions in most cases.

### 8.2.1 Update Rule

Q-learning learns an action-value function, approximated by a Q-table, of size  $N_s \times N_a$ , whose elements are Q-values. The Q-learning step update rule is the following [105]

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)], \quad (8.1)$$

where  $Q(s_t, a_t)$  is the Q-value related to the state  $s_t$  and the action  $a_t$ ,  $r_{t+1}$  is the reward calculated at the  $t + 1$  time instant,  $\alpha$  is the learning rate and

$\gamma$  is the discount factor. At each step, the agent will decide whether to act randomly or follow the update rule. This choice depends on a parameter  $\epsilon$ . The higher  $\epsilon$  is, the less likely the agent will take a random action. This parameter regulates the trade-off between exploration (random action) and exploitation (greedy action). There are various approaches to setting the value of  $\epsilon$ . The two main ones foresee respectively to set  $\epsilon$  to a fixed low value ( $\epsilon$ -greedy method) or to start with a high  $\epsilon$  value and then decrease it as the episodes go by ( $\epsilon$ -decay method). The  $\epsilon$  parameter is used when selecting specific actions based on the calculated Q values; setting  $\epsilon = 0$  means choosing the highest Q value among all the stored Q values for a specific state. This causes an issue in exploration, as the algorithm can easily get stuck at local optima.

Therefore, in this work, we introduce randomness using the  $\epsilon$ -decay method.

In this particular scenario, we adopted a variant of the classic Q-learning where multiple agents are considered. In fact, each UAV will update its Q-table, also based on the other agents' information.

## 8.2.2 Rewards

Rewards are the most crucial part of the algorithm. They define the purpose of the problem and the behavior of the agents. Considering that it is a localization problem, the rewards should depend on the quality of the estimation of the target's position. A well-known figure of merit for a successful localization is the GDOP. In wireless sensor networks, the anchor node position's geometry has a significant influence on the positioning accuracy. For this reason, GDOP has been used to relate the agents' spatial disposition to the accuracy of the target position estimation. Let us define the matrix  $\mathbf{H}$  as follows

$$\mathbf{H} = \begin{pmatrix} \frac{\hat{x}-x_1}{d_1} & \frac{\hat{y}-y_1}{d_1} & 1 \\ \frac{\hat{x}-x_2}{d_2} & \frac{\hat{y}-y_2}{d_2} & 1 \\ \vdots & \vdots & \vdots \\ \frac{\hat{x}-x_N}{d_N} & \frac{\hat{y}-y_N}{d_N} & 1 \end{pmatrix}, \quad (8.2)$$

where  $(\hat{x}, \hat{y})$  is the estimated position of the target. Then,  $\widehat{\text{GDOP}}$  is given by

$$\widehat{\text{GDOP}} = \sqrt{\text{trace}(\mathbf{H}^T \mathbf{H})^{-1}}. \quad (8.3)$$

The closest  $\widehat{\text{GDOP}}$  is to 1, the better the target localization is. Another important index of good localization performance is the distance between the agent and the transmitter. The closer the UAVs are to the target, the less is the error in estimating its position. The rewards are normalized according to the size of the environment and ensure the convergence of the algorithm. They are always negative except when the best configuration is reached. This occurs when the average distance between the agents and the target is less than a threshold distance  $d_0$  and  $1 \leq \widehat{\text{GDOP}} \leq 1.3$ . Only in this case, the reward becomes  $r = r_0$ .

Each step of the algorithm,  $\widehat{\text{GDOP}}$  and the distances between the agents and the target are estimated. Then, the corresponding reward is calculated by

$$r = \begin{cases} \frac{r_0}{A}, & \text{if } d_n \leq d_0 \text{ and } 1 \leq \widehat{\text{GDOP}} \leq 1.3 \\ \frac{-k-d_n}{A}, & \text{otherwise} \end{cases}, \quad (8.4)$$

where  $k$  is a scalar linearly dependent on  $\widehat{\text{GDOP}}$ ,  $d_n$  is the average distance between the three UAVs and the target, and  $A$  is the area of the environment. In particular,  $k$  is calculated by

$$k = m \widehat{\text{GDOP}} + k_0, \quad (8.5)$$

where  $m$  and  $k_0$  are tunable parameters.

When all the agents receive the reward  $\frac{r_0}{A}$ , the episode is interrupted.

### 8.3 Algorithm Validation

This section presents several figures of merit of the RL-based algorithm to evaluate its performance in different conditions.

The number of episodes is kept constant in the simulations, while the number of steps changes according to the grid size. Moreover, the agents'

initial position, as well as the target position (in the center of the grid), are fixed.

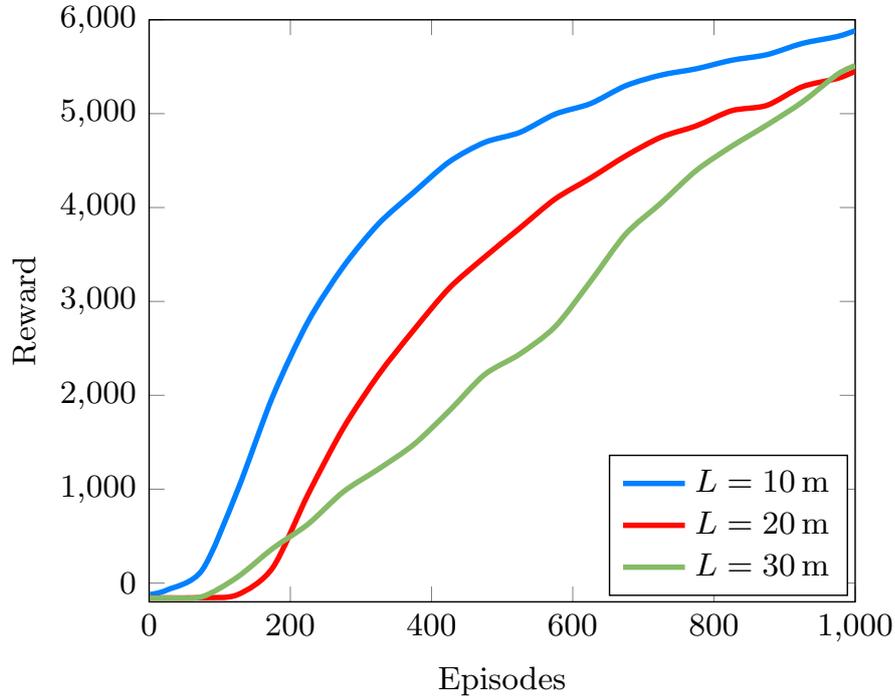
Three square grids of size  $L$  equal to 10 m, 20 m, 30 m, corresponding to a number of available states  $N_s$  equal to 100, 400, and 900, respectively, are considered. The number of episodes is  $N_{ep} = 1000$ , while the number of steps per episode is set to 100, 200, 1000 in the three grids respectively. The number of steps varies according to the size of the environment. The smaller it is, the fewer iterations will be required to complete the algorithm. The experiments are repeated 500 times, and the corresponding performance is averaged. The terminal state is reached when all the episode steps are completed, or the agents get the maximum reward. The parameters are set to  $r_0 = 10000$ ,  $m = 150$ , and  $k_0 = 160$ , while the multi-agent Q-learning parameters are  $\alpha = 0.2$ , and  $\gamma = 0.99$ .

An increase in the size of the environment drops the performance of the Q-learning algorithm. In general, considering large grids, the algorithm's success is not guaranteed. In that case, more complicated methods that approximate the high number of states (i.e., deep Q-learning) are suggested.

### 8.3.1 Rewards

In this test, we study the average rewards received by the agents during the execution of the Q-learning algorithm.

As depicted in Fig. 8.3, the average reward received by the agents in the last episodes is almost the same for the three scenarios. Although the higher reward is  $r_0$ , its value reached after 1000 episodes is lower; this is due to the randomness of the algorithm and to the discretization of the environment that, in some cases, might prevent the agents from reaching the optimal configuration. However, in the first half of the episodes, the averaged reward rises fast in the smaller scenarios while it shows a linear dependence with the number of episodes in case of  $L = 30$  m. This means that, in this case, the algorithm requires more episodes to learn the optimal policy. To compare the three scenarios, the rewards are denormalized.



**Figure 8.3:** Average reward for the agents as a function of the number of episodes in three different environments.

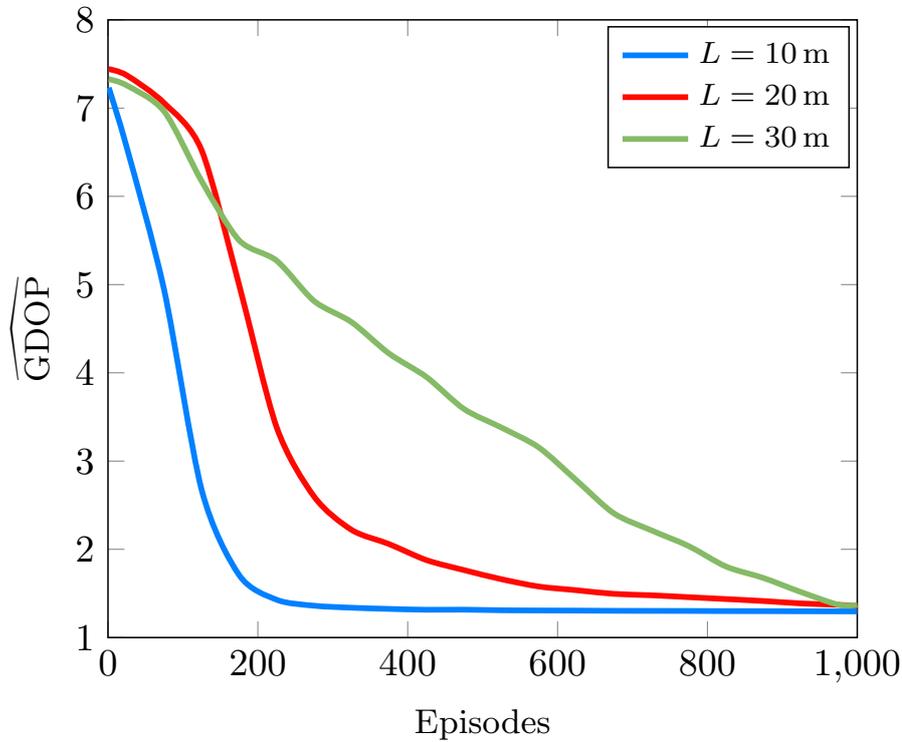
### 8.3.2 GDOP

This test analyzes the average  $\widehat{\text{GDOP}}$  per episode, as shown in Fig. 8.4. Since the  $\widehat{\text{GDOP}}$  tends to infinity when the agents reach some particularly bad configuration, and since the agents start the first episode from an unfavorable position, the  $\widehat{\text{GDOP}}$  is very high at the beginning.

However, as shown in Fig. 8.4, it decreases to a floor around 1. As for the rewards, the  $\widehat{\text{GDOP}}$  saturates earlier in the smaller environments, while it shows a linear dependence with the number of episodes in case of  $L = 30$  m.

### 8.3.3 Localization Error

In the final test, we studied how the localization accuracy varies during the episodes of the RL algorithm. As a figure of merit for the localization algorithm, we used the RMSE. As depicted in Fig. 8.5, as the number of episodes

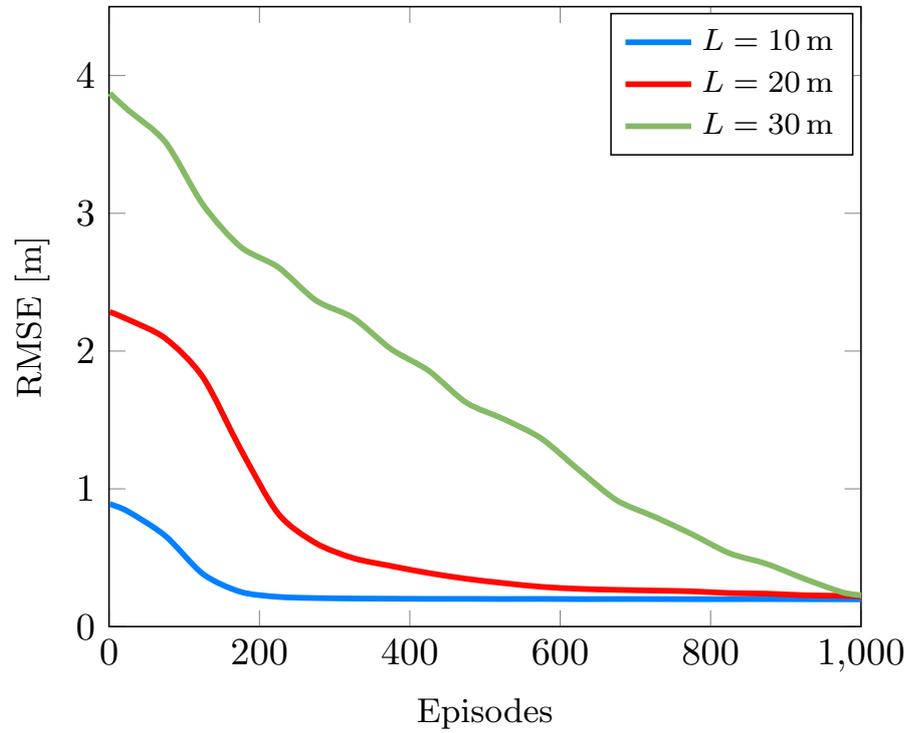


**Figure 8.4:** Average  $\widehat{\text{GDOP}}$  for the agents as a function of the number of episodes in three different environments.

increases, the RMSE decreases. Again, the RMSE reaches its minimum earlier in small environments, while it shows an almost linear dependence with the number of episodes when  $L = 30$  m.

### 8.3.4 Conclusions

The purpose of this chapter is to propose a viable RL-based navigation algorithm to enable a swarm of UAVs to find the best spatial configuration to localize a non-cooperative transmitter in an unknown environment. In particular, to govern the UAVs we adopt the well-known multi-agent Q-learning algorithm, designing an ad-hoc reward function based on the distance between the UAVs and the target, and on the estimated GDOP. Numerical results show how the proposed algorithm performs well, reaching a  $\text{RMSE} < 1$  m at the end of the last episode in the three scenarios. The main limit of



**Figure 8.5:** RMSE of location estimation as a function of the number of episodes in three different environments.

the proposed solution is the size of its Q-table; in fact, it starts performing worse when the dimension of the environment, and so the number of possible states, increases. A patrol can adopt the proposed system to perform spectrum monitoring and localization of a malicious user, i.e., a jammer, aided by the wireless network analytics tools discussed in the previous chapters.



# Chapter 9

## Conclusions

The objective of this thesis was the development of a novel framework for the extraction of a set of analytics of a non-collaborative wireless network whose key features are unknown. The analysis is performed exploiting only the over-the-air power profiles captured by RF sensors.

The framework combines BSS, measurement association, excision filtering, and analytics extraction algorithms. The analytics extracted are the number and position of the target network nodes, their logical topology and the application-level traffic generated by the nodes.

To answer *Q1*, the framework validation showed that extracting the analytics of a wireless network from external is possible with satisfactory performance. In particular, the research outcomes for novel methodologies for the extraction of network analytics are further described.

### Blind Source Separation

The proposed BSS methodology, based on F-ICA, has been proven effective in the validation scenario. To answer *Q2*, the node counting tests proved that estimating the number of transmitters from the mixture of signals collected by the RF sensors is possible. Moreover, the reconstruction error of the BSS algorithm reaches  $R_e = 4\%$  in a mild shadowing regime. Thus, the mixtures can be unmixed with relatively high performance, even when the position of the target nodes, necessary for the permutation algorithm, is

roughly estimated. This provides the answer to *Q3 (Is it possible to separate over-the-air signals to get the transmitted traffic profiles of each node?)*.

### Multiple Transmitter Localization

This work proved that it is possible to localize nodes of a non collaborative packet-based wireless network using only over-the-air power profiles captured by RF sensors. This provides the answer to *Q4 (Is the RSS-based localization of multiple unknown transmitters possible?)*.

To answer *Q5 (Is the proposed approach better than the current state-of-the-art?)*, the results confirmed the satisfactory performance of the proposed solution, showing how BSS combined with MLE outperformed a state-of-the-art algorithm in realistic channels with noise and shadowing.

Furthermore, to answer *Q6, Q7, and Q8* we found that in a mild shadowing regime, even with relatively few sensors, i.e.,  $\rho \approx 1$ , the localization error can be small when MLE position estimation is adopted. We also showed that the performance degradation due to BSS is tolerable considering that it can locate multiple transmitters.

Finally, Chapter 8 presented a viable RL-based navigation algorithm that enables a swarm of UAVs to find the best spatial configuration to localize a non-cooperative transmitter in an unknown environment. A spectrum patrol equipped with the proposed system can monitor a wide area and identify and localize a malicious user with the help of network analytics.

### Topology Inference

To answer *Q9*, the logical topology has been successfully extracted.

This step is performed by adopting state-of-the-art causal inference methods such as Granger causality (GC) and conditional transfer entropy (CTE), which exploit the times series of traffic profiles, and a novel solution based on a properly designed and trained NN that makes use of distilled time-based features. This provides the answer to *Q10 (Can the causality inference approach be improved?)*.

The numerical results accounting for packet collisions, nodes mobility, and

realistic channel impairments, such as noise, and shadowing, revealed that topology inference of a wireless network is possible, even with a relatively low number of sensors. Moreover, we found that in mild shadowing regimes and low mobility, the performance in terms of probability of detection and probability of false alarm is remarkably good, especially for the proposed NN-based solution. These considerations, provided in Sections 7.4.2, 7.4.3, and 7.4.4 answer to *Q11* and *Q12*.

### Traffic Classification

A user traffic classification framework for wireless networks based on RF measurements has been proposed. To answer *Q13*, the classification methodology validation showed that the classification of the application-level traffic generated by the nodes of a wireless network from external is possible.

We showed that, after the BSS, the NN outperforms the other classifiers achieving remarkable performance also in case of propagation impairments (e.g., shadowing), and with a short observation window (30 ms). This answers to *Q14* and *Q15*.

To answer *Q16* (*How many sensors are needed to reach a prescribed performance?*), the analysis of the proposed solution revealed that the number of RF sensors strongly impacts the performance of the algorithms. For example, in the considered scenario  $M = 15$  sensors were necessary to classify the traffic of  $N = 3$  nodes with satisfactory performance. This is because traffic classification is affected by imperfect power profile reconstruction of the transmitted signals at the nodes.



# List of Tables

7.1	Set of parameters used in the tests described in Section 7.2 . . .	66
7.2	Probability of correct estimation of the number of transmitters, $\widehat{N}$ , as a function of $\rho$ and $\sigma$ . . . . .	69



# List of Figures

1.1	An illustration of a next generation wireless network interfered by a smart jammer. A spectrum patrol, e.g., an UAV, senses the RF medium and sends the data to the authority or regulator. The regulator extracts a set of analytics from the data to acquire as much information as possible about the scenario and eventually detect the jammer. . . . .	5
1.2	An example of urban scenario where a cloud of RF sensors aboard UAVs monitors a wireless network to extract analytics. Alternatively, sensors could be deployed in fixed positions within the network landscape. . . . .	6
1.3	An example of a spectrum patrolling scenario composed of the nodes of a wireless network, many patrol units equipped with RF sensors, and a fusion center. The patrol units collect the over-the-air power profiles transmitted by the nodes and send them to the fusion center. The fusion center, which can also be an authority (or regulator) extracts the wireless network analytics and gets a detailed description of the analyzed network. Such information can be used to secure the network from malicious users (i.e., jammers) and optimize spectrum usage. . . . .	7

1.4	A logical block schematic representation of the proposed framework. The spectrum patrol collects the over-the-air traffic profiles generated by the wireless network nodes. Then, BSS is performed to unmix the received signals. The last stage is the extraction of network analytics. . . . .	10
2.1	A cloud of randomly distributed RF sensors (in blue) across the wireless network landscape (in orange). . . . .	18
2.2	An illustration of the ED: bandpass zonal filter followed by a square-law device and finite time integrator. . . . .	19
3.1	A block scheme of the blind source separation process. . . . .	22
3.2	Examples of positively and negatively skewed distributions. . . . .	24
3.3	Shape of a distribution with respect to its kurtosis. . . . .	26
3.4	An illustration of the eigenvalue selection procedure. The green line shows the amplitude of the eigenvalues, the red dashed line represents the threshold, and the blue dotted line highlights the number of eigenvalues selected. . . . .	28
4.1	An example of scenario with $N = 5$ , $M = 8$ , and $\sigma = 3$ dB. The red circles are the network nodes (or targets), the blue triangles are the RF sensors and the grey circles represent the estimate of the position of the nodes through the BSS-MLE methodology. . . . .	38
4.2	Block scheme of the complete methodology: target nodes ( $\mathcal{T}_1, \mathcal{T}_2, \dots$ ), RF sensors ( $\mathcal{S}_1, \mathcal{S}_2, \dots$ ), node counting (principal component analysis (PCA)), received power profiles separation (fast independent component analysis (F-ICA)), transmitted power profile reconstruction (RSS extraction), and localization. . . . .	39
4.3	The RSS extraction process for two non-collaborative transmitters. . . . .	41
5.1	An example of directed graph representing a wireless network's logical topology. The dashed links in red represent the absence of flow of information between the nodes. . . . .	48

5.2	Block scheme of the wireless network topology inference: sensor network, data pre-processing and inference. . . . .	49
5.3	An example of successful transmission of data packets between nodes $\mathcal{N}_1$ and $\mathcal{N}_2$ is shown. In this case node $\mathcal{N}_1$ is connected to node $\mathcal{N}_2$ but not to node $\mathcal{N}_3$ . Notice how the time-to-ack $\tau^{1 \rightarrow 2}$ and $\tau^{1 \rightarrow 3}$ differ each other. . . . .	55
5.4	Example of a two-dimensional normalized features space. In red: points corresponding to the presence of a link ( $a_{i,j} = 1$ ). In blue: points representing couples of nodes that are not connected ( $a_{i,j} = 0$ ). The color gradient shows the decision boundary identified by the classification algorithm. . . . .	56
5.5	The structure of the 2-hidden-layer NN used to infer the presence/absence of the links. . . . .	57
6.1	System overview: the network of sensors monitoring the traffic patterns among the wireless network nodes. . . . .	60
6.2	An example of user packet profiles for the three activities considered. . . . .	61
7.1	BSS algorithm performance varying the standard deviation $\sigma_P$ of the location uncertainty, the shadowing parameter $\sigma_S$ and the density $\rho_S$ . . . . .	67
7.2	Performance of BSS compared to the SF parametric method varying $\rho_S$ and $\sigma_S$ according to Table 7.1. . . . .	68
7.3	Comparison between the mean and the standard deviation of the localization error of the three algorithms (BSS-LS, BSS-MLE and particle simulation (denoted as PS)) varying the ratio between the number of sensors and nodes, $\rho$ , in a mild shadowing regime with $\sigma_S = 1$ dB. . . . .	70
7.4	The 20-th and 80-th percentiles and the mean localization error, for both BSS-MLE and particle simulation, varying the shadowing parameter $\sigma_S$ (dB) for $\rho = 1.8$ and $\rho = 4$ . . . . .	71

7.5	The 20-th and 80-th percentiles and the RMSE of BSS-MLE localization with the proposed BSS compared with the ideal BSS (denoted as IS) varying $\rho$ with $\sigma = 5$ dB. The CRLB is used as a benchmark to assess the asymptotical improvement of the localization performance. . . . .	72
7.6	(a) The directed graph representing the real topology of the network, whose adjacency matrix $\mathbf{A}$ is shown in (b); (c) and (d) show examples of the inferred topology graph and adjacency matrices $\hat{\mathbf{A}}$ estimated over the Monte Carlo trials of the same topology depicted in (a) while varying the position of the nodes and the sensors in the landscape; (e) visualization of the performance where $p_D^{i,j}$ and $p_{FA}^{i,j}$ indicates the probability of detection and false alarm of the directed link from node $i$ to node $j$ , calculated over 100 Monte Carlo trials; the corresponding matrix of $p_D^{i,j}$ and $p_{FA}^{i,j}$ is shown in (f). . . . .	75
7.7	$p_D$ and $p_{FA}$ of the topology inference algorithms as a function of the density of nodes $\rho_N$ for $\rho_S = 0.3$ sensors/m <sup>2</sup> and $\sigma_S = 3$ dB. . . . .	77
7.8	$p_D$ and $p_{FA}$ of the topology inference algorithms as a function of the shadowing parameter $\sigma_S$ (dB) for $\rho_S = 0.3$ sensors/m <sup>2</sup> . . . . .	78
7.9	$p_D$ and $p_{FA}$ of the topology inference algorithms in the parameter setting $A_1$ varying the velocity $v$ of the wireless network nodes. . . . .	79
7.10	Accuracy varying the number of sensors $M$ with an observation window of 1 s. . . . .	81
7.11	Accuracy as a function of the observation window duration for $M = 15$ sensors. . . . .	82
8.1	Example of wireless network setup with $N = 3$ UAVs that localize a malicious transmitter (mounted on a rover) and forward the information to a BS (i.e., the regulator). . . . .	86
8.2	An example of grid representation of the environment where the agents can move, searching for the optimal spatial configuration to localize the transmitter (rover). . . . .	87

- 8.3 Average reward for the agents as a function of the number of episodes in three different environments. . . . . 91
- 8.4 Average  $\widehat{\text{GDOP}}$  for the agents as a function of the number of episodes in three different environments. . . . . 92
- 8.5 RMSE of location estimation as a function of the number of episodes in three different environments. . . . . 93



# Acronyms

**ACK** acknowledgment

**AGC** asymmetric Granger causality

**AI** artificial intelligence

**AIC** Akaike information criterion

**AP** access point

**AR** auto-regressive

**AWGN** additive white Gaussian noise

**BS** base station

**BSS** blind source separation

**CE** conditional entropy

**CR** cognitive radio

**CRLB** Cramèr-Rao lower bound

**CTE** conditional transfer entropy

**ED** energy detector

**FIM** Fisher's information matrix

**F-ICA** fast independent component analysis

**GC** Granger causality

**GDOP** geometric dilution of precision

**GPS** Global Positioning System

**GPS-RTK** Global Positioning System real-time kinematic

**GSP** graph signal processing

**HSMM** hidden semi-Markov model

**KPCA** kernel principal component analysis

**ICA** independent component analysis

**IoT** internet of things

**IoIT** internet of intelligent things

**i.i.d.** independent, identically distributed

**KL** Kullback–Leibler

**LOS** line-of-sight

**LS** least squares

**MAC** medium access protocol

**MC** Monte Carlo

**MDL** minimum description length

**ML** machine learning

**MLE** maximum likelihood estimation

**NN** neural network

**NLOS** non-line-of-sight

**PCA** principal component analysis

**RBF** radial basis function

**RF** radio-frequency

**RL** reinforcement learning

**RMSE** root mean squared error

**RSS** received signal strength

**r.v.** random variable

**SF** spatial filtering

**STDM** statistical time-division multiplexing

**SU** secondary user

**SVM** support vector machine

**TE** transfer entropy

**UAV** unmanned aerial vehicle

**UWB** ultra wideband



# Bibliography

- [1] F. Tariq, M. R. A. Khandaker, K.-K. Wong, M. A. Imran, M. Bennis, and M. Debbah, “A speculative study on 6g,” *IEEE Wireless Communications*, vol. 27, no. 4, pp. 118–125, Aug. 2020.
- [2] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, “6g: Opening new horizons for integration of comfort, security, and intelligence,” *IEEE Wireless Communications*, vol. 27, no. 5, pp. 126–132, Mar. 2020.
- [3] L. Columbus, “Roundup of internet of things forecasts and market estimates,” *Forbes*, 2016.
- [4] I. I. for Business Value, “Internet of threats: Securing the internet of things for industrial and utility companies,” 2020. [Online]. Available: <https://www.ibm.com/downloads/cas/ZJRRVRKW>
- [5] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, “Machine learning paradigms for next-generation wireless networks,” *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, Dec. 2017.
- [6] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, “Deep learning for physical-layer 5g wireless techniques: Opportunities, challenges and solutions,” *IEEE Wireless Communications*, vol. 27, no. 1, pp. 214–222, Aug. 2020.
- [7] M. Abusubaih, “Intelligent wireless networks: Challenges and future research topics,” *Journal of Network and Systems Management*, vol. 30, Jan. 2022.

- 
- [8] D. Huang, A. Al-Hourani, K. Sithampanathan, W. S. T. Rowe, L. Bulot, and A. Thompson, "Deep learning methods for device authentication using RF fingerprinting," in *IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*, Sydney, Australia, Dec. 2021, pp. 1–7.
- [9] E. Testi, E. Favarelli, and A. Giorgetti, "Reinforcement learning for connected autonomous vehicle localization via uavs," in *IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, Online, Nov. 2020, pp. 13–17.
- [10] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks*. Springer International Publishing, 2016, pp. 213–226.
- [11] J. Mitola and G. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.
- [12] A. Schwartz, S. Kandeepan, P. Conder, and K. Wang, "Sensing-based SU access performance study with a four-state markov model in cognitive radios," in *IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*, Sydney, Australia, Dec. 2021, pp. 1–7.
- [13] T. Nawaz, D. Campo, M. O. Mughal, L. Marcenaro, and C. S. Regazzoni, "Jammer detection algorithm for wide-band radios using spectral correlation and neural networks," in *International Wireless Communications and Mobile Computing Conference (IWCMC)*, Jun. 2017, pp. 246–251.
- [14] M. Camilo, D. Moura, J. Galdino, and R. M. Salles, "Anti-jamming defense mechanism in cognitive radios networks," in *IEEE Military Communications Conference (MILCOM)*, Jan. 2012, pp. 1–6.

- 
- [15] J. Xu, S. Ying, and H. Li, “Gps interference signal recognition based on machine learning,” *Mobile Networks and Applications*, vol. 25, pp. 2336–2350, Dec. 2020.
- [16] L. Qian, X. Li, and S. Wei, “Cross-layer detection of stealthy jammers in multihop cognitive radio networks,” in *International Conference on Computing, Networking and Communications (ICNC)*, Jan. 2013, pp. 1026–1030.
- [17] Y. Arjoune, F. Salahdine, M. Islam, E. Ghribi, and N. Kaabouch, “A novel jamming attacks detection approach based on machine learning for wireless communication,” in *International Conference on Information Networking (ICOIN)*, Barcelona, Spain, Jan. 2020.
- [18] A. Krayani, M. Farrukh, M. Baydoun, L. Marcenaro, Y. Gao, and C. Regazzoni, “Jammer detection in m-qam-ofdm by learning a dynamic bayesian model for the cognitive radio,” in *European Signal Processing Conference (EUSIPCO)*, Coruna, Spain, Sep. 2019, pp. 1–5.
- [19] T. Nawaz, L. Marcenaro, and C. Regazzoni, “Defense against stealthy jamming attacks in wide-band radios: A physical layer approach,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Montreal, Canada, Nov. 2017, pp. 497–501.
- [20] A. Krayani, M. Baydoun, L. Marcenaro, A. S. Alam, and C. Regazzoni, “Self-learning bayesian generative models for jammer detection in cognitive-uav-radios,” in *IEEE Global Communications Conference (GLOBECOM)*, Taipei, Taiwan, Dec. 2020, pp. 1–7.
- [21] A. Chakraborty, A. Bhattacharya, S. Kamal, S. R. Das, H. Gupta, and P. M. Djuric, “Spectrum patrolling with crowdsourced spectrum sensors,” in *IEEE Conference on Computer Communications (INFOCOM)*, Honolulu, USA, Apr. 2018, pp. 1682–1690.
- [22] D. T. Kanapram, M. Marchese, E. L. Bodanese, D. M. Gomez, L. Marcenaro, and C. Regazzoni, “Dynamic bayesian collective aware-

- ness models for a network of ego-things,” *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3224–3241, Dec. 2020.
- [23] A. Krayani, A. S. Alam, M. Calipari, L. Marcenaro, A. Nallanathan, and C. Regazzoni, “Automatic modulation classification in cognitive-iot radios using generalized dynamic bayesian networks,” in *IEEE World Forum on Internet of Things (WF-IoT)*, New Orleans, LA, USA, Jul. 2021, pp. 235–240.
- [24] N. Radhakrishnan, S. Kandeepan, X. Yu, and G. Baldini, “Soft fusion based cooperative spectrum prediction using LSTM,” in *IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*, Sydney, Australia, Dec. 2021, pp. 1–7.
- [25] E. Testi, E. Favarelli, and A. Giorgetti, “Machine learning for user traffic classification in wireless systems,” in *Europ. Sig. Proc. Conf. (EUSIPCO)*, Rome, Italy, Sep. 2018, pp. 2040–2044.
- [26] E. Favarelli, E. Testi, L. Pucci, and A. Giorgetti, “Anomaly detection using WiFi signal of opportunity,” in *IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*, Surfers Paradise, Gold Coast, Australia, Dec. 2019.
- [27] IEEE 802.22, “Standard for Wireless Regional Area Networks—Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Policies and procedures for operation in the TV Bands,” Jul. 2011.
- [28] A. Mariani, A. Giorgetti, and M. Chiani, “Effects of noise power estimation on energy detection for cognitive radio applications,” *IEEE Trans. Commun.*, vol. 59, no. 12, pp. 3410–3420, Dec. 2011.
- [29] A. Mariani, S. Kandeepan, and A. Giorgetti, “Periodic spectrum sensing with non-continuous primary user transmissions,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1636–1649, Mar. 2015.

- [30] H. Urkowitz, “Energy detection of unknown deterministic signals,” *Proc. IEEE*, vol. 55, no. 4, pp. 523–531, April 1967.
- [31] A. Mariani, A. Giorgetti, and M. Chiani, “Wideband spectrum sensing by model order selection,” *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6710–6721, Dec. 2015.
- [32] G. L. Stuber, *Principles of Mobile Communication*, 2nd ed. Norwell, MA, USA: Kluwer Academic Publishers, 2001.
- [33] X. Yu, D. Hu, and J. Xu, *Blind Source Separation: Theory and Applications*, 1st ed. Wiley Publishing, 2014.
- [34] M. Joho, H. Mathis, and R. H. Lambert, “Overdetermined blind source separation: using more sensors than source signals in a noisy mixture,” in *Int. Conf. on Indep. Comp. Analysis and Blind Signal Sep. (ICA)*, Helsinki, Finland, Jun 2000, pp. 81–86.
- [35] A. Cichocki, R. Zdunek, and S. Amari, “New algorithms for non-negative matrix factorization in applications to blind source separation,” in *IEEE Int. Conf. on Acoustics Speech and Sig. Processing Proceedings*, vol. 5, Toulouse, France, June 2006.
- [36] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis,” *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.
- [37] G. Ivkovic, P. Spasojevic, and I. Seskar, “Localization of packet based radio transmitters in space, time, and frequency,” in *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2008, pp. 288–292.
- [38] A. Hyvarinen, “Fast and robust fixed-point algorithms for independent component analysis,” *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 626–634, May 1999.

- [39] A. Hyvärinen and E. Oja, “A fast fixed-point algorithm for independent component analysis,” *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, Jul. 1997.
- [40] M. Wax and T. Kailath, “Detection of signals by information theoretic criteria,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 387–392, Apr. 1985.
- [41] A. Mariani, A. Giorgetti, and M. Chiani, “Model order selection based on information theoretic criteria: Design of the penalty,” *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2779–2789, Jun. 2015.
- [42] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 2002.
- [43] H. Nurminen, M. Dashti, and R. Piché, “A survey on wireless transmitter localization using signal strength measurements,” *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–12, Feb. 2017.
- [44] K. Pahlavan, X. Li, M. Ylianttila, R. Chana, and M. Latva-aho, “An overview of wireless indoor geolocation techniques and systems,” in *International Conference on Mobile and Wireless Communication Networks (IFIP)*, Paris, France, May 2000, pp. 1–13.
- [45] E. Testi, E. Favarelli, L. Pucci, and A. Giorgetti, “Machine learning for wireless network topology inference,” in *IEEE Int. Conf. on Signal Proc. and Comm. Syst. (ICSPCS)*, Surfers Paradise, Gold Coast, Australia, Dec. 2019.
- [46] M. Z. Win, A. Conti, S. Mazuelas, Y. Shen, W. M. Gifford, D. Dardari, and M. Chiani, “Network localization and navigation via cooperation,” *IEEE Communications Magazine*, vol. 49, no. 5, pp. 56–62, 2011.
- [47] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, “Soft information for Localization-of-Things,” *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Nov. 2019.

- [48] M. Z. Win, F. Meyer, Z. Liu, W. Dai, S. Bartoletti, and A. Conti, “Efficient multisensor localization for the Internet of Things: Exploring a new class of scalable localization algorithms,” *IEEE Signal Process. Mag.*, vol. 35, no. 5, pp. 153–167, Sep. 2018.
- [49] S. Bartoletti, A. Conti, and M. Z. Win, “Device-free counting via wide-band signals,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1163–1174, May 2017.
- [50] S. Bartoletti, W. Dai, A. Conti, and M. Z. Win, “A mathematical model for wideband ranging,” *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 2, pp. 216–228, Mar. 2015.
- [51] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, “Ranging with ultrawide bandwidth signals in multipath environments,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 404–426, 2009.
- [52] C. An, Y. K. An, S.-M. Yoo, and B. E. Wells, “Efficient data association to targets for tracking in passive wireless sensor networks,” *Ad Hoc Networks*, vol. 75-76, pp. 19 – 32, Jun. 2018.
- [53] K. Magowe, A. Giorgetti, S. Kandeepan, and X. Yu, “Accurate analysis of weighted centroid localization,” *IEEE Trans. on Cogn. Commun. Netw.*, vol. 5, no. 1, pp. 153–164, Mar. 2019.
- [54] S. Bartoletti, Z. Liu, M. Z. Win, and A. Conti, “Device-free localization of multiple targets in cluttered environments,” *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2021.
- [55] Z. Li, A. Giorgetti, and S. Kandeepan, “Multiple radio transmitter localization via UAV-based mapping,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8811–8822, 2021.
- [56] P. Schulz, N. Franchi, and G. Fettweis, “RSS-based localization of multiple unknown transmitters through particle simulation,” in *Proc. IEEE Int. Symp. on Joint Comm. & Sensing*, Dresden, Germany, Feb. 2021, pp. 1–6.

- 
- [57] E. Testi and A. Giorgetti, “Rss-based localization of multiple radio transmitters via blind source separation,” *IEEE Communications Letters*, pp. 1–1, 2021.
- [58] A. Giorgetti and K. Sithampanathan, *Cognitive Radios Techniques: Spectrum Sensing, Interference Mitigation and Localization*. Artech House, Norwood, USA, 2012.
- [59] M. Zafer, B. J. Ko, and I. W. Ho, “Transmit power estimation using spatially diverse measurements under wireless fading,” *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1171–1180, Jan. 2010.
- [60] J. Wenli, G. Teng, and J. Meiyin, “Researching topology inference based on end-to-end data in wireless sensor networks,” in *Int. Conf. on Int. Comp. Tech. and Autom. (ICICTA)*, vol. 2, Shenzhen, China, Apr. 2011, pp. 683–686.
- [61] T. Kontos, G. S. Alyfantis, Y. Angelopoulos, and S. Hadjiefthymiades, “A topology inference algorithm for wireless sensor networks,” in *IEEE Symp. on Comp. and Comm. (ISCC)*, Cappadocia, Turkey, Jul. 2012, pp. 479–484.
- [62] Y. Gao, W. Dong, C. Chen, J. Bu, W. Wu, and X. Liu, “ipath: Path inference in wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 517–528, Feb. 2016.
- [63] C. Yu, K. Chen, and S. Cheng, “Cognitive radio network tomography,” *IEEE Trans. on Veh. Technol.*, vol. 59, no. 4, pp. 1980–1997, May 2010.
- [64] Y. Vardi, “Network tomography: estimating source-destination traffic intensities from link data,” *J. of the American Stat. Ass.*, vol. 91, no. 433, pp. 365–377, Mar. 1996.
- [65] Y. E. Sagduyu, Y. Shi, A. Fanous, and J. H. Li, “Wireless network inference and optimization: algorithm design and implementation,” *IEEE Trans. Mobile Comput.*, vol. 16, no. 1, pp. 257–267, Jan. 2017.

- 
- [66] C. Partridge, D. Cousins, A. W. Jackson, R. Krishnan, T. Saxena, and W. T. Strayer, “Using signal processing to analyze wireless data traffic,” in *Proc. ACM Work. on Wireless Security (WiSE)*, New York, NY, USA, Sep. 2002, pp. 67–76.
- [67] S. Kokalj-Filipovic, C. Bertoncini Acosta, and M. Pepe, “Learning structural properties of wireless ad-hoc networks non-parametrically from spectral activity samples,” in *IEEE Global Conf. on Signal and Info. Proc. (GlobalSIP)*, Washington, DC, USA, Dec 2016, pp. 1092–1097.
- [68] G. Mateos, S. Segarra, A. Marques, and A. Ribeiro, “Connecting the dots: Identifying network structure via graph signal processing,” *IEEE Signal Process. Mag.*, vol. 36, pp. 16–43, May 2019.
- [69] Y. Shen, B. Baingana, and G. B. Giannakis, “Nonlinear structural equation models for network topology inference,” in *Annual Conference on Information Science and Systems (CISS)*, Princeton, NJ, USA, Mar. 2016, pp. 163–168.
- [70] P. A. Traganitis, Y. Shen, and G. B. Giannakis, “Network topology inference via elastic net structural equation models,” in *European Signal Processing Conf. (EUSIPCO)*, Kos, Greece, Aug. 2017, pp. 146–150.
- [71] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
- [72] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. New York, NY, USA: Cambridge University Press, 2009.
- [73] C. W. J. Granger, “Investigating causal relations by econometric models and cross-spectral methods,” *Econometrica*, vol. 37, no. 3, pp. 424–438, Aug. 1969.
- [74] P. Tilghman and D. Rosenbluth, “Inferring wireless communications links and network topology from externals using Granger causality,”

- in *IEEE Mil. Comm. Conf. (MILCOM)*, San Diego, CA, USA, Nov. 2013, pp. 1284–1289.
- [75] X. Qin and W. Lee, “Statistical causality analysis of Infosec alert data,” in *Recent Advances in Intrusion Detection*, G. Vigna, C. Kruegel, and E. Jonsson, Eds. Springer Berlin Heidelberg, 2003, pp. 73–93.
- [76] M. Laghate and D. Cabric, “Learning wireless networks’ topologies using asymmetric Granger causality,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 233–247, Feb. 2018.
- [77] H. Xu, M. Farajtabar, and H. Zha, “Learning Granger causality for Hawkes processes,” in *International Conference on Machine Learning ICML*, vol. 48, New York, NY, USA, Feb. 2016.
- [78] J. Etesami, N. Kiyavash, K. Zhang, and K. Singhal, “Learning network of multivariate Hawkes processes: A time series approach,” in *Conf. on Uncertainty in Artif. Int. (UAI)*, Jersey City, New Jersey, USA, Mar. 2016, pp. 162–171.
- [79] M. G. Moore and M. A. Davenport, “Analysis of wireless networks using Hawkes processes,” in *IEEE Int. Work. on Signal Proc. Adv. in Wireless Comm. (SPAWC)*, Edinburgh, UK, Jul. 2016, pp. 1–5.
- [80] T. Schreiber, “Measuring information transfer,” *Phys. Rev. Lett.*, vol. 85, pp. 461–464, July 2000.
- [81] P. Sharma, D. J. Bucci, S. K. Brahma, and P. K. Varshney, “Communication network topology inference via transfer entropy,” *IEEE Trans. Netw. Sci. Eng.*, pp. 1–7, Jan. 2019.
- [82] J. Sun, D. Taylor, and E. Bollt, “Causal network inference by optimal causation entropy,” *SIAM Journal on Applied Dynamical Systems*, vol. 14, p. 73–106, Jan. 2014.
- [83] S. Kokalj-Filipovic, P. Spasojevic, and A. Poylisher, “Non-parametric learning to infer wireless relays, routes and traffic patterns from time

- series of spectrum activity,” in *Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Oct. 2017, pp. 920–927.
- [84] K. Friston, R. Moran, and A. Seth, “Analysing connectivity with Granger causality and dynamic causal modelling,” *Current Opinion in Neurobiology*, vol. 23, no. 2, pp. 172–178, Apr. 2013.
- [85] A. K. Seth, A. B. Barrett, and L. Barnett, “Granger causality analysis in neuroscience and neuroimaging,” *J. of Neuroscience*, vol. 35, no. 8, pp. 3293–3297, Feb. 2015.
- [86] M. Ding, Y. Chen, and S. Bressler, *Granger Causality: basic Theory and Application to Neuroscience*. Wiley-VCH Verlage, Sep. 2006.
- [87] M. Foster, “Causal inference and developmental psychology,” *Developmental Psychology*, vol. 46, no. 6, pp. 1454–1480, Nov. 2010.
- [88] A. K. Seth, “A Matlab toolbox for Granger causal connectivity analysis,” *J. of Neuroscience Methods*, vol. 186, no. 2, pp. 262 – 273, Feb. 2010.
- [89] M. Wibral, N. Pampu, V. Priesemann, F. Siebenhühner, H. Seiwert, M. Lindner, J. T. Lizier, and R. Vicente, “Measuring information-transfer delays,” *PLOS ONE*, vol. 8, pp. 1–19, Feb. 2013.
- [90] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Verlag, Aug. 2006.
- [91] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [92] J. Watt, R. Borhani, and A. K. Katsaggelos, *Machine Learning Refined*. Cambridge University Press, 2016.
- [93] S. Valenti, D. Rossi, A. Dainotti, A. Pescape, A. Finamore, and M. Mellia, “Reviewing traffic classification,” *Lect. Notes in Comp. Science*, vol. 7754, pp. 123–147, Feb. 2013.

- 
- [94] T. De Schepper, M. Camelo, J. Famaey, and S. Latré, “Traffic classification at the radio spectrum level using deep learning models trained with synthetic data,” *International Journal of Network Management*, 2020.
- [95] J. Kornycky, O. Abdul-Hameed, A. Kondozi, and B. Barber, “Radio frequency traffic classification over WLAN,” *IEEE/ACM Trans. on Netw.*, vol. 25, no. 1, pp. 56–68, Feb. 2017.
- [96] B. Schölkopf, A. Smola, E. Smola, and K.-R. Müller, “Nonlinear component analysis as a kernel eigenvalue problem,” *Neural Computation*, vol. 10, pp. 1299–1319, 1998.
- [97] E. Favarelli, E. Testi, and A. Giorgetti, “One class classifier neural network for anomaly detection in low dimensional feature spaces,” in *IEEE Int. Conf. on Signal Proc. and Comm. Systems (ICSPCS)*, Gold Coast, Australia, Dec. 2019.
- [98] M. Lacage and T. R. Henderson, “Yet another network simulator,” in *Proc. Work. on Ns-2: the IP netw. sim. (WNS2)*, Pisa, Italy, Oct. 2006, p. 12–22.
- [99] P. Stuedi, O. Chinellato, and G. Alonso, “Connectivity in the presence of shadowing in 802.11 ad hoc networks,” in *Proc. IEEE Wireless Comm. and Netw. Conf.*, vol. 4, New Orleans, LA, USA, Mar. 2005, pp. 2225–2230.
- [100] P. Stoica and Y. Selen, “Model-order selection: a review of information criterion rules,” *IEEE Signal Proc. Mag.*, vol. 21, no. 4, pp. 36–47, July 2004.
- [101] C. Bettstetter, “Mobility modeling in wireless networks: categorization, smooth movement, and border effects,” *ACM Mobile Computing and Communications Review*, vol. 5, no. 3, p. 55–66, Jul. 2001.

- 
- [102] J. T. Lizier, “Jidt: An information-theoretic toolkit for studying the dynamics of complex systems,” *Frontiers in Robotics and AI*, vol. 1, pp. 1–11, Dec. 2014.
- [103] S. Mignardi, C. Buratti, A. Bazzi, and R. Verdone, “Trajectories and resource management of flying base stations for C-V2X,” *Sensors*, vol. 19, no. 4, p. 811, 2019.
- [104] S. Mignardi, C. Buratti, V. Cacchiani, and R. Verdone, “Path optimization for unmanned aerial base stations with limited radio resources,” in *IEEE Int. Symp. on Pers., Indoor and Mobile Radio Comm. (PIMRC)*, 2018, pp. 328–332.
- [105] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

