UNIVERSITY OF CATANIA

DEPARTMENT OF ELECTRICAL, ELECTRONIC AND COMPUTER ENGINEERING

PhD IN SYSTEM, ENERGY, COMPUTER AND TELECOMMUNICATIONS ENGINEERING

XXXV CYCLE

Innovative Smart Cities Telemedicine and Environment Risk Management Techniques through DL-based IoT Sensors

Roberta Avanzato

Coordinator: Prof. Paolo Pietro Arena

Tutor: Prof. Francesco Beritelli

Acknowledgement

With immense pleasure and deep sense of gratitude, I wish to express my sincere thanks to my tutor **prof. Francesco Beritelli**, without his motivation and continuous encouragement, this research would not have been successfully completed.

I wish to extend my profound sense of gratitude to **my parents and Antonio** for all the sacrifices they made during my research and also providing me with moral support and encouragement whenever required.

Place: Catania

Date: 24/11/2022 **Roberta Avanzato**

Contents

List of Terms and Abbreviations

AI	Artificial Intelligence
AP	Access Point
AUC	Area Under the Curve
BS	Base Station
CID	Cell Identifier
CVD	Cardiovascular Disease
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DWT	Discret Wavelet Transformation
DL	Deep Learning
eNB	eNodeB
FPS	Frame Per Second
FDR	Fischer Discriminant Ratio
FFT	Fast Fourier Transform
ECG	Electrocardiogram
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
GT	Game Theory
GPS	Global Positioning System
ICT	Information and Communication Technology

IoT	Internet of Things
KPI	Key Performance Indicators
LAC	Location Area Code
LAN	Local Area Network
LoS	Line of Sight
LTE	Long Term Evolution
MFCC	Mel Frequency Cepstrum Coefficients
MIMO	Multiple-input Multiple-output
MI	Myocardial Infarction
ML	Machine Learning
MLP	Multilayer Perceptron
MLVPN	Multi Link Virtual Private Network
NF	Network Function
NN	Neural Network
PER	Packet Error Rate
PCG	Phonocardiogram
PLR	Packet Loss Rate
QoE	Quality of Experience
QoS	Quality of Service
RF	Radio Frequency
ROC	Receiver Operator Characteristic
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indication

SLP	Single Layer Perceptron
SNR	Signal to Noise Ratio
UE	User Equipment
UAV	Unmanned Aerial Vehicles
UMTS	Universal Mobile Telecommunications System
VMAF	Video Multimethod Assessment Fusion
VPN	Virtual Private Network

List of Figures

List of Tables

ABSTRACT

The rapid urbanization process in the last century has deeply changed the way we live and interact with each other. As most people now live in urban areas, cities are experiencing growing demands for more efficient and sustainable public services that may improve the perceived quality of life, specially with the anticipated impacts of climatic changes. In this already complex scenario with increasingly overcrowded urban areas, different types of emergency situations may happen anywhere and anytime, with unpredictable costs in human lives and economic losses. In order to cope with unexpected and potentially dangerous emergencies, smart cities initiatives have been developed in different cities, addressing multiple aspects of emergencies detection, alerting, and mitigation.

In this thesis, three main issues relating to environmental and health risk management in smart cities were addressed. The first topic focuses on the possibility of using drone-femtocell systems to locate devices under rubble in post-earthquake scenarios by devising location algorithms with very low error and very high drone energy efficiency.

The second topic addresses detection and classification of rainfall levels using different types of signals (audio, video, radio) and deep learning techniques. Classifying rainfall intensity precisely and in real-time would mean providing smart cities with a system that predicts and manages hydrogeological risk conditions (landslides, floods and inundations) in cities. And thus, it provides information in terms of road safety and much more; such as, aspects related to the management of mobile radio connections in order to maintain good radio signal quality during heavy rainfall situations.

The third topic concerns healthcare management within smart cities, in particular, the possibility of defining innovative algorithms for the detection and classification (in very short times and with high precision) of heart disease using ECG and PCG signals in conjunction with deep learning techniques.

Keywords: *Risk Management*, *Health Management*, *IoT devices*, *Localization Algorithm*, *Deep Learning*, *Rainfall Estimation*, *Heart Disease Classification.*

Chapter 1

Introduction

The development of new communication technologies, data processing algorithms, and cyber physical systems has not only transformed the way we gather information and interact with each other, but also how cities have evolved in the last decade [1, 2]. Affordable high-bandwidth communication networks and miniaturized hardware components with increasing processing power have become a reality, with deep but sometimes imperceptible impacts on the way we comprehend and handle different urban environments [3, 4]. For an increasing number of cities, smarter has become common sense [5].

The availability of new technological resources is expected to be a breakthrough for the development of sustainable, resilient, and smarter cities [6]. By exploiting huge amounts of heterogeneous data, it is possible to better understand the multiple complexities of the urban environments, eventually leading to the implementation of "smart services" [7, 8]. Among them, emergencies management is expected to be a fundamental service in modern cities, with direct impact on urban safety and the perceived quality of life.

Generally speaking, emergencies have been an old and recurrent problem in cities, although their impact and influences have changed as cities grew larger [9]. The spatial distribution of the cities and their geography through the centuries, as well as inherent characteristics such as poor sanitation, low mobility efficiency, dominance of wooden buildings, and the absence of rescue and emergency teams, have made cities highly susceptible to catastrophes resulted from emergency situations. With the industrialization process and the further adoption of motor vehicles and telephone networks, emergencies management in modern cities improved, but actions were still dependent on emergency calls and non automated dispatching of response vehicles [10]. Currently, considering the new technologies available, more efficient solutions have been sought in order to minimize the

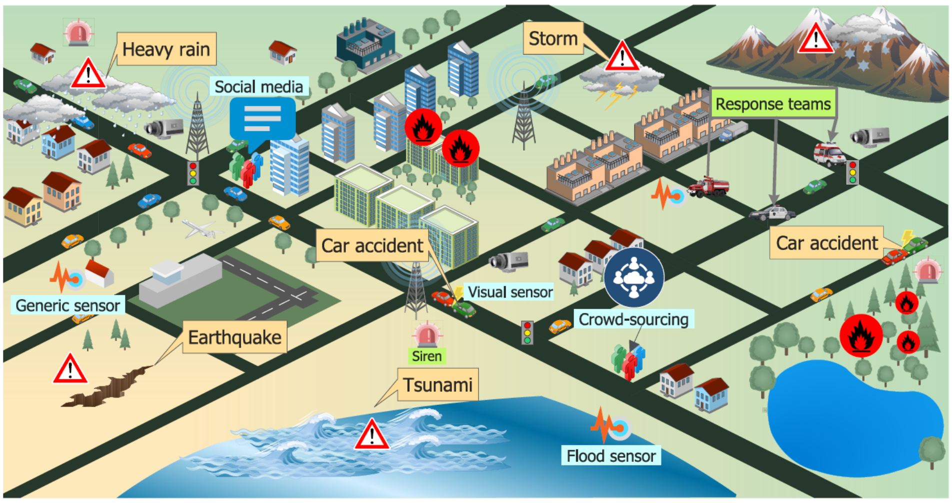


Figure 1.1: Urban areas: a crisis-prone environment.

negative impacts of emergencies.

Emergencies can be better detected and managed with the use of electronic sensors, smartphones, personal gadgets, vehicles, drones, robots, social media networks, web servers, and could based services. Usually, cyber-physical systems will be created to manage emergencies, processing data ows to provide one or more services in a city.

In fact, such emergencies management systems will operate processing multiple types of data in a urban scale, which indeed is one of the fundamentals of the so-called smart cities paradigm [6]. This is the conceptual background from where modern emergencies management systems have ourished. Therefore, in order to make cities safer and more resilient, emergencies management systems have embraced different technologies to provide detection, alerting, and mitigation services in a city, automatizing different steps of the emergencies management cycle. However, since solutions have been developed following different approaches and premises, there is no consensus when addressing this problem, which may impair research and developments efforts in this area. Thus, a better understanding of this subject is desired. Figure 1.1 depicts a general schema of an emergencies management system in a smart city with multiple detected emergencies, highlighting different sources of data [11].

This thesis focuses on two main themes:environment related risks and health risks. In particular, this work addresses 3 important issues:

i. Risk management in post-earthquake scenarios: a drone-femtocell system was proposed in order to locate the devices under the rubble. This system comes together through the study, design and implementation of various localization algorithms and optimization of the drone's energy efficiency. In particular, the idea is to send one or more drones with femtocells on board to the monitoring area hit by a catastrophe. These systems provide radio-mobile coverage in the relevant area allowing us the possibility to apply all the localization algorithms in existance and / or those created ad hoc for the scenario in question. The goal is to find the devices in the shortest possible time and with a very low localization error (1-3 meters) despite the complexity of the scenario, due to the presence of rubble with different density and level attenuation. Several simulation tools, using software such as python and C#, were employed in order to create tools for: data collection, algorithm evaluation (localization error estimation and drone energy efficiency) and simulation of monitoring areas (with presence or less than rubble of different density and level attenuation);

ii. Risk management in scenarios affected by hydrogeological instability: a multimodal rain gauge based on audio, video and radio recording (collection of radio parameters relating to LTE technology) was built for real-time detection of the intensity of rain in a specific area. The data is collected in a database which is subsequently fed as input to a neural network. After a training phase of the network, a phase of testing and real-time inference follows. This system (multimodal rain gauge and neural network) will be able to detect the intensity of the rain and possibly send alerts when it is constantly increasing (or at very high levels). This will make it easier to prevent landslides and floods and ensure road safety. In the case of using the radio parameters of the LTE signal, the possibility of understanding the various levels of intensity of the rain was studied. In addition to estimating the intensity levels and intervening in anticipation of the formation of hydrogeological instability, it led to the birth of a new study. This study proposes algorithms to solve the problem of radio channel fluctuations caused by the presence of rain, since the trend of some radio parameters negatively affects the performance of the network;

iii. Health risk management: among the various issues that fall within the management of health in the context of smart cities, in this thesis, greater attention will be given to the field of cardiology. In particular, heart disease has been a major cause of death among the population in recent years. Providing tools and algorithms, within smart cities, capable of preventing illness problems due to a heart disease is a topic of great interest. In this

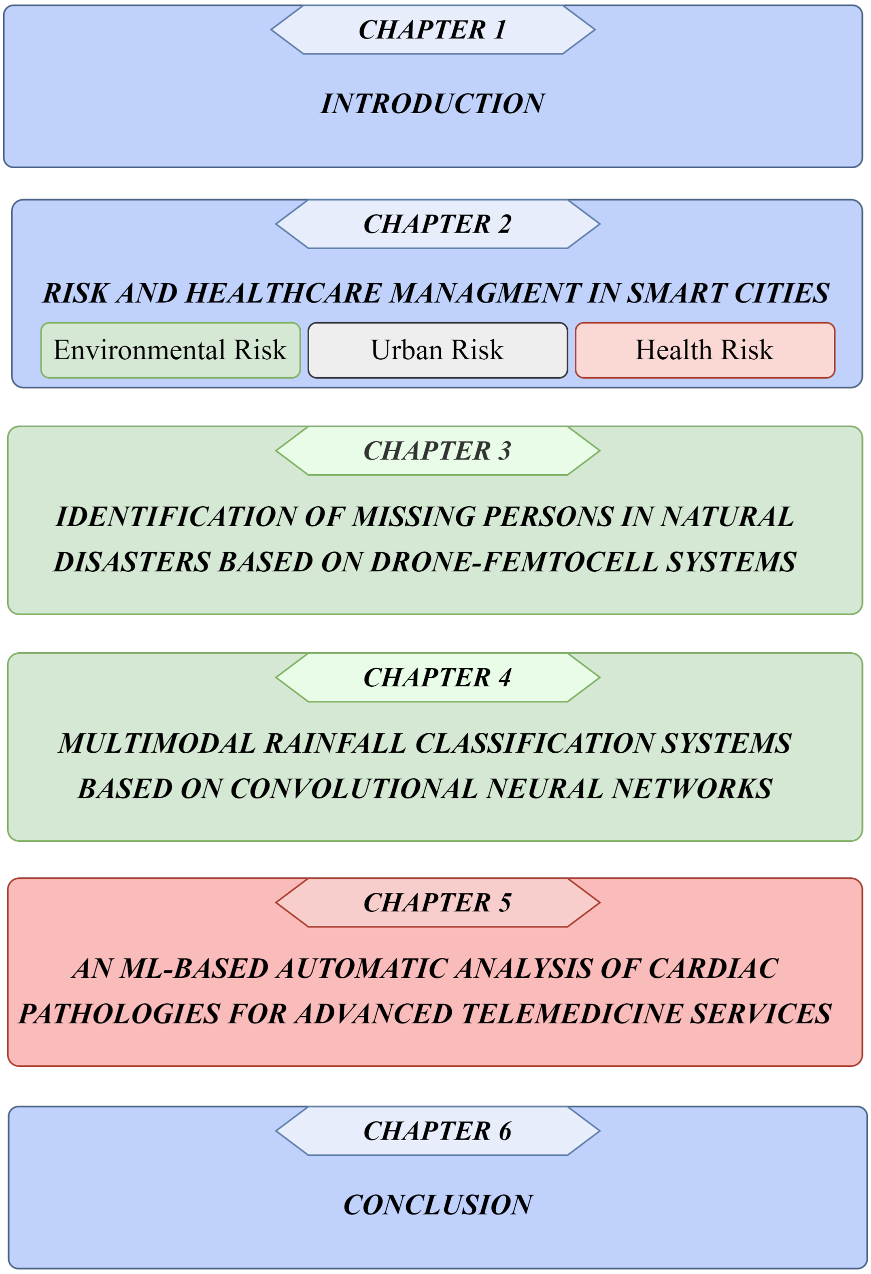

CHAPTER 1
INTRODUCTION
CHAPTER 2
RISK AND HEALTHCARE MANAGMENT IN SMART CITIES
Environmental Risk
Urban Risk
Health Risk
CHAPTER 3
IDENTIFICATION OF MISSING PERSONS IN NATURAL DISASTERS BASED ON DRONE-FEMTOCELL SYSTEMS
CHAPTER 4
MULTIMODAL RAINFALL CLASSIFICATION SYSTEMS BASED ON CONVOLUTIONAL NEURAL NETWORKS
CHAPTER 5
AN ML-BASED AUTOMATIC ANALYSIS OF CARDIAC PATHOLOGIES FOR ADVANCED TELEMEDICINE SERVICES
CHAPTER 6
CONCLUSION

Figure 1.2: Thesis's Structure.

thesis, a series of recognition and classification algorithms of various cardiac pathologies have been proposed through the use of ECG and PCG signals and deep learning techniques. The main objective is to recognize the presence of any cardiovascular diseases in the shortest time (2-60 seconds) and with high accuracy. This allows medical support (through wearable devices, for example smart fitness wristbands, smart-watches, pacemakers, subcutaneous monitors and such) to be provided quickly and preventively at any time and in any place.

This thesis work is structured as shown in Figure 1.2.

Chapter 2 will provide an overview of the new services and technologies that smart cities offer to the population; in view of this, the state of the art on emergency management systems (environmental, urban and health) will be examined for ideas that seem viable as a possible service to be provided in the definition of smart cities.

Chapter 3 will focus on the management of a particular type of risk: post-earthquake scenarios. In particular, the chapter will be dedicated to the description of the various device localization techniques in post-earthquake scenarios developed in the PhD period. In this scenario, a drone-femtocell system and various localisation algorithms were proposed in order to optimize the level of energy efficiency (flight and processing time) of the drones and the localisation error of each device. Chapter 4 will focus on the management of hydrogeological risks in smart cities. In particular, the chapter will be dedicated to the description of various techniques for the detection and classification of rainfall levels using deep learning techniques and different types of signals (audio, video and radio), studied in the PhD period. In this scenario, a multi-modal rain gauge was built and different algorithms based on audio, video and radio parameters of the LTE system were implemented in order to classify rain intensity levels.

Chapter 5 will focus on health-related risk management. In particular, the chapter will be devoted to describing the various techniques for detecting and classifying cardiac pathologies by means of deep learning techniques and ECG/PCG signals (obtained from public databases), studied during the PhD period.

Finally, in Chapter 6 the conclusions of this thesis will be drawn.

Chapter 2

Risk and healthcare management in Smart Cities

The term “smart cities” generally refers to the use of technology based solutions to enhance the quality of life for citizens, improve interaction with government and promote sustainable development. A city can be described as smart where social, environmental and economic development factors are balanced and linked via devolved processes to more efficiently manage key assets, resources and urban flows for real-time processes. Smart cities are designed around an ICT based infrastructure with IoT enabled sensor technology to support social and urban inter-connectivity through greater citizen interaction and government efficiency.

The benefits of developing smart cities will be huge and will benefit every sphere of life. Smart cities will produce smart citizens, smart living, smart administration, automated homes, smart mobility, smart waste disposal system, smart surroundings, smart economy, and much more. Smart cities will add luxury to the lives of people. To make the concept of smart cities anactuality, effective tools and technologies must be implemented. The IoT will play a significant role in the development of smart cities [11, 12].

The IoT refers to a network in which all the physical devices, vehicles, buildings, and various electronics and electrical devices which we use daily are interconnected to each other over the Internet so that they may gather and share data amongst themselves. The collected data from various devices may further be analyzed and processed for better management and monitoring of traffic and transport systems, pure drinking water supply and sanitation, proper disposal of waste, power plants, air and water pollution, better healthcare, effective video surveillance systems, a better hydrogeological and seismic risk management in post-disaster scenarios, etc.

When considering the development of research works within the context of smart cities, some concerns may arise due to the nature of such environments. In recent years, smart cities initiatives have been proposed and implemented, embracing new technologies to enhance the citizens' quality of life, potentially making the urban living easier and bringing better management of its resources. Although such objectives seem to be clear, the number and types of complexities that emerge from smart cities initiatives are considerable, raising concerns that may echo and reach the development of emergencies management systems.

Much research efforts have been devoted to transform cities into smart cities [13, 14]. Since this is inherently a multidisciplinary area, researchers have tried to define best practices and engineering procedures to provide smart services in a city, taking into consideration development issues such as sustainability, social responsibility, and energy consumption [15]. As a result, the literature in this area is diverse and potentially huge, covering different subjects comprising multiple views of the same urban environment.

In a typical smart city, several initiatives are devised to make citizens' life better and easier, improving several services such as public transportation, traffic management,water and energy supply, among others. In this sense, emergencies management systems arise as one of the possible services to be provided when defining smart cities, which may even coexist with other parallel services.

The basic function of emergencies management systems in a smart city is the processing of conceptual emergencies along the time in response to detected critical situations. Such emergencies will typically be associated to the causes that created them, influencing the way how emergencies will be alerted and mitigated. Particularly, it is reasonable to say that emergencies will be associated to one or more causes (hazards), as well as to a group of additional information (metadata) that will give more details to support emergencies processing.

In a urban environment, a hazard is any source of potential damage that may harm people or incur in economic losses when it becomes an emergency. Usually, a hazard is perceived as an emergency when it is a current threatening condition, and it ceases to exist when it represents no more risks.

The proper modelling of the hazards is then of paramount importance for this type of applications. The most usual approach to model and process hazards is to monitor a particular variable along the time. Actually, such approach is easier to implement and may produce very quick responses when employing electronic sensor devices or active data sources. Differently, hazards may be also processed as more complex variables, for example employing cameras [16] or artificial intel-

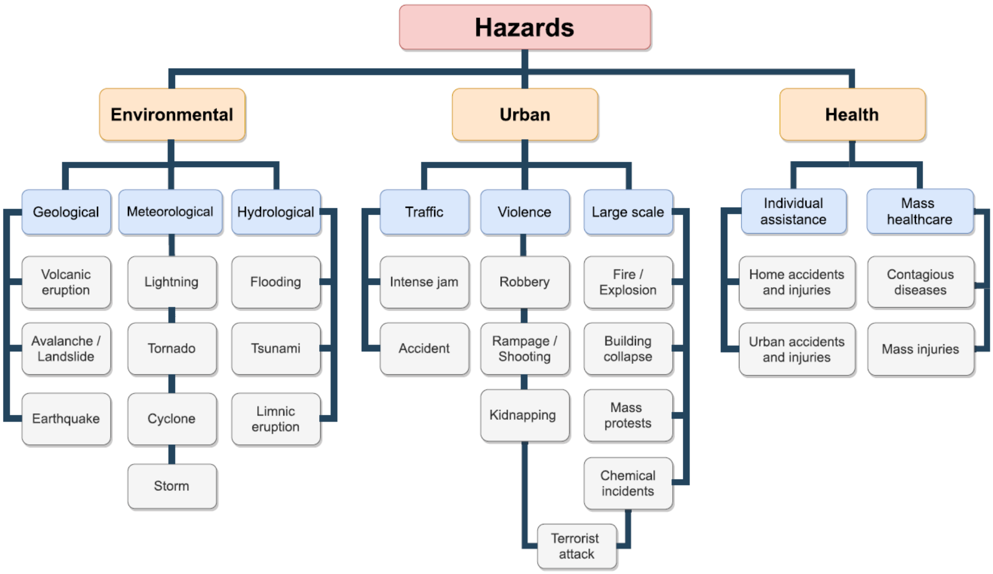


Figure 2.1: Classification and some common hazards in modern cities [11].

ligence algorithms [17] to detect hazards that could not be easily identified using individual sensors, generating different types of emergencies. As an example, a fire event could be identified processing still images or processing public posts on social media, potentially providing different types of metadata to support emergencies mitigation actions. Whatever the case, each system will have a particular configuration for hazards monitoring and emergencies detection, according to the characteristics of the target city.

Since the nature of a hazard will dictate how an emergency will be eventually mitigated in a urban environment [18, 19], city-related hazards may be classified into three different groups: Environmental, Urban and Health. Environmental hazards are those resulted from natural conditions that may affect a city, such as heavy rain, hurricanes, earthquakes, volcano eruptions, among others.

The other two types of hazards, Urban and Health, are both causes of human induced disasters. Can be subdivide them into two different groups due to the expected relevance that outbreaks surveillance and detection systems should assume in the development of smart cities. This way, Urban hazards will be related to the way we live in cities, with increasing overpopulated areas and crowded mobility systems, resulting in hazards related to traffic accidents, house firing, gas explosion, building collapsing, terrorist attacks, violent protests, robbery, etc. Finally, Health hazards will not only be associated to individual health emergencies, such as heart attacks, but collective threats due to the spread of infectious diseases. Figure 2.1 presents a comprehensive organization of the expected hazards in urban areas.

Human-induced disasters may result in infrastructural damages, injuries and deaths. Although there is not a straight line between the frequency of Urban and Health hazards and the urbanization process, the urban sprawl in this century will result in more large and mega cities around the world, with potential for higher number of disasters [20]. In parallel, climate changes are strengthening the destructive power of natural disasters, putting additional pressure on emergencies management systems. As a result, the last decades have seen an increasing in the number of emergencies detection approaches, focusing on different types of hazards.

2.1 Disaster Risk Monitoring and Management System in Smart Cities

In general, many scientific studies have focused on providing new solutions to identify and manage hazards induced by human and nature in the context of smart cities. In recent years, UAVs have helped manage disasters such as floods, fires, earthquakes, etc. Knowledge about the disaster-hit areas through aerial images, or other audio/radio sensors, gained by UAVs helps emergency evacuations and rescue missions by finding a safe route to make help reach where it is needed.

For the earthquake scenarios, most of the disaster-related studies are focused on the post-disaster phase only and use conventional and learning-based algorithms with applications to localize victims and optimize paths. Regarding the UAV communication network (UAVCN), the key challenges are communication issues, resource allocation, UAV deployment, defining UAV trajectory, and content security. UAV path planning's key barriers are path optimization, path completeness, optimality, efficiency, and achieving robustness [21].

The localization approach of victims is not limited only using UAVs system. When some of the infrastructures are down on account of a disaster, the effects it have on localization accuracy has been modeled in [22]. They used the Pedestrian Dead Reckoning and Wi-Fi RSSI fingerprinting models to simulate indoor localization. A hybrid algorithm has been proposed to provide location information of victims to rescue workers using time-of-arrival and received power of GSM network [23]. They simulated and showed the trade-offs between location error and path-loss exponent. A similar work with real deployment has been presented in [24]. In this work, a local GSM base station was deployed and directional antennas were used to locate victim's mobile phone. A camera based victim localization model is proposed in [25], where the localization error is not specified explicitly. In [26], an android app is used to detect several victims trapped under the WLAN network of a building by using Euclidean Distance (ED) algorithm. In recent times, building information modeling (BIM) has gain significant consideration in industry as a central repository of building information. In [27] a BIM-based Indoor Location (BIMIL) protocol is designed for automated data extraction and transformation of BIM emergency-related data for public safety purposes. This approach can help to localize crucial portions of a disastrous construction site with indoor positioning data to support emergency responses. Availability of an up-to-date layout of a building is crucial for faster rescue management. After localization, access to an optimized path to the victim plays a significant role for emergency support. Often due to structural and interior changes inside a building it is difficult to maintain a synchronic layout. One probable approach is to reconstruct the 3D models of the building after any renovation in an automatic approach. In [28], a complete workflow is introduced that generate 3D models from point clouds of a building to support sophisticated path planning for disaster management. However, this study does not anyhow facilitate localizing the victims. The study in [29] proposed a WSN based support for emergency responders or rescuers. It proposed a joint routing and localizing algorithm based on pre-deployed Wi-Fi network. The study in [30] proposed an inertial sensor based technique to localize first-responders in disaster scenario. For inertial sensors such as, accelerometer, gyroscope, and magnetometer to work, a centralized system needs to know the starting position of the first-responder. Moreover, continuous sensing is required. In [31] the authors contributed to develop a prototype for victim localization (VLoc) that is not dependent on any pre-installed infrastructure, and exploits pervasive devices and networks, such as smartphones and Wi-Fi. In particular, during a post disaster situation, a text message (SMS) is sent to victims' phone number from a number registered for emergency purposes. An app installed in the victim's smartphone intercepts this text mes-sage and switches on the Wi-Fi hotspot. Operating victims' phones in this mode is crucial for locating one or more victims' with their approximate locations in the disaster site. Once an approximate location is identified, rescuers leverage trilateration or multilateration techniques on the RSSI from the Wi-Fi signal of victims' phones. VLoc estimates the current location of a victim from three or four reference points in trilateration or multilateration. The reference points are the rescuers and VLoc provides the distance in meters to the victim from all rescuers.

Smartphone devices, therefore, play a fundamental role in emergency scenarios thanks to their pervasiveness and the ability to convey the emergency requests of the people involved to the rescue teams. At the same time, effective use of such devices in critical scenarios with limited mobile Internet access is challenging. Alternatively, several recent research studies have proposed an Emergency Communication System (ECS) based on short-range Device-to-Device (D2D) solutions available on Commercial Off The Shelf (COTS) devices (e.g. Wi-Fi Direct); however, the target of these solutions is small internal areas, as scaling over large environments is often a problem. In [32] authors propose a distributed algorithm aimed to build a connected graph over a phone-network, by properly setting the role of each device as client, relay or Group Owner (GO). In [33], the authors propose SENSE-ME, a phone-based infrastructure less ECS providing multi-hop connectivity (via Wi-Fi Direct), sensing data sharing (via information centric networking), and distributed data processing for emergency detection (via consensus algorithms). Beside the communication technology, another key issue of phone-based network is the strategy used to disseminate data among devices when considering the intermittent connectivity and the likely presence of network partitions. In [34] an integrated platform for smartphone connectivity in disaster recovery is described; the proposed system, called Team-Phone, supports both the creation of energy-efficient spontaneous groups among survivors, and a multi-hop messaging system between survivors and rescue teams, integrating AODV and opportunistic routing mechanisms. In [35], authors proposed a phone-based Emergency Communication System (ECS) enabling long range communication among survivors and rescue teams over critical environments where 3/4G cellular connectivity is not available and the traditional geo-localization technologies (e.g. the GPS) provide only partial coverage of the environment. The proposed system consists of a mobile application connected to a LoRa transceiver via Bluetooth Low Energy (BLE); through the app, users can send emergency requests that are re-broadcasted by other peers until reaching a rescue personnel who is able to handle the emergency.

Seismic scenarios are not the only calamitous events that incur in smart cities. Due to climate change caused by global warming in recent years, hydrogeological disasters (river floods, landslides, etc.) are increasingly on the rise. In this context, risk management process for identifying flood situations and taking the coordinative steps for its treatment plays a great role in planning and decision making in smart city.

Several studies, for an example, concern on urban construction of urban rain flood sponges. Authors in [36] provided a lateral study emphasizing the rain flood management process in the developing countries and summarize the foreign rain flood management system periodically. The process of development and evolution, the governance model of comprehensive, space features of regional convergence, and deep form of urban development system analyzes the reason for combining this study with the practical situation of China's cities. They also presented the future urban rain flood management method, which can lead to the development in the direction of thinking and can use the history of exploration to assist the future multi-scale hydrological conditions in the rain flood damage. Authors in [37] analyze the generally-used at the rain flood management measures, and summarize its advantages and disadvantages in China to put forward the background of "sponge city". In [38],the authors contributed to putting forward the suitability for sponge key design strategy of city constructions, the total content and regulatory rules in the content integration, combining the characteristics of all kinds of the new city district. The study propose a planned and feasible urban rain flood ecosystem is designed to promote the construction of sponge city, improve the water environment, control urban waterlogging, reduce runoff pollution, improve river and lake water quality, recycle rainwater resources, replenish groundwater, etc. The traditional regulation and sponge city construction are reconstructed and integrated into this work to provide a feasible urban rain flood ecosystem in the industrial and smart city scenario.

Other studies, on this topic, regard the rainfall estimation and monitoring using several type of signal (audio, radio, images), in order to detect previously, the flood insurgent. Smart cities around the world are supported by high-capacity wireless communication networks, which are based on millimeter-waves links. The propagating waves are sensitive to hydrometeors, and their signal level is attenuated by rain. Recently, wireless telecommunication links have gained attention as a promising rainfall measurement method because the power of received signals, which is extremely sensitive to rainfall in microwave and millimetric frequency bands, can be measured everywhere for communications. With the roll-out of 5G wireless networks, relevant research has indicated that densely distributed terrestrial links can measure rainfall in urban areas either as a primary or supplementary precipitation monitoring method [39]. Many studies have also shown that dense commercial earth-space links are potential global high resolution rainfall monitoring systems [40, 41]. In [42] authors demonstrate, using actual measurements from the city of Rehovot, Israel, how high-resolution rain maps can be generated from the received signal level measurements collected by these links. They first propose a method for reducing the errors in converting signal attenuation to rainfall estimates in short, in-city links. The proposed method requires calibration of model parameters using side information from either a rain gauge or a long link in the vicinity of the network. The results of the calibrating method was analyzed empirically using either auxiliary measurements and show that the performance is satisfactory for both. Then, was applied a spatial interpolation method on the rainfall resulting estimates, and demonstrated the construction of an high-resolution 2-D map of the accumulated rain in a city. In [43], authors investigated rainfall monitoring results from experimental measurements and deep learning approaches such as artificial neural networks and long short-term memory. The experimental setups were in South Korea over terrestrial and satellite links, and in Ethiopia over terrestrial link for different frequency bands and link distances. The received signal power data were used to derive the rainfall rate distribution and compared to actual rainfall measurements over the same time periods. Results demonstrate that the proposed deep learning-based models generally have a good t with the measured rainfall rates. The rainfall rate generated from terrestrial links was a better to the actual rainfall rate data than that generated from satellite links.

Smart cities, in addition to being equipped with high-capacity wireless communication networks, are equipped with video surveillance systems capable of recording and transmitting real-time audio and video data to the cloud of the network. In this context, it is possible to record and classify the audio obtained from the rain in order to detect its intensity. In [44] authors used smartphones to collect the sound of raindrops hitting umbrellas for rainfall measurement. Study in [45] designed an Arduino-based acoustic sensor composed of a microphone and tipping bucket for rainfall monitoring and warning. Moreover, authors in [46] used a set of acoustic features to build a decision tree to classify non-rain and heavy rain events in environmental audio recordings. Study [47], for an example, using surveillance audio as input and an automatic rainfall level classification system was built. Three 2-D baseline CNNs were proposed as the classifiers. In view of the classifier training and testing, a new dataset named Rainfall Audio_XZ (RA_XZ) was generated based on the surveillance audio data. The experimental results demonstrate that the proposed CNN (7-stack CNN) achieves 81.67% accuracy in rainfall level classification in the RA_XZ dataset. Furthermore, audio aggregation strategies that facilitate the representation and classification of rainfall events are investigated, which has important implications for audio and speech classification systems.

Accordingly, managing disasters such as floods, fires, earthquakes, in the smart city is really important for people's safety. For this reason, two main topic were explored during the PhD period:

- Localization of mobile devices in post-earthquake scenarios, through the introduction of different algorithms, simulation tools and testbed;
- Classification of the rainfall intensity through audio / video / radio signals and techniques of ML and DL in order to monitor and predict the insurgent of floods and / or landslides.

2.2 Healthcare Monitoring and Management System in Smart Cities

The concept of smart city healthcare is one that many traditional cities aim to emulate by setting up conventional devices and equipment for integrating healthcare resources with smart solutions. Smart solutions and ICT play a crucial role in ensuring smart cities' success in providing citizens with quality healthcare services. The smart city's vital goals include making provision for high quality living, conserving healthcare service quality, and promoting more conducive quality conditions for citizens. Sensors, monitoring, and control are required to make healthcare cities smarter. These sensors' feedback values help healthcare providers carry out monitoring and control through a series of automation. The IoT, wireless sensor networks, deep learning, and other technologies can be used successfully to accomplish these goals. Smart cities can quickly attend to many people's healthcare needs at once by having access to real-time information. Healthcare providers can make quick decisions that yield positive results. IoTs, AI, and computing technology have changed the face of healthcare. Sensors can be implanted in the body or worn on the body's surface, such as smartwatches [48].

In general, so, the idea of continuously monitoring a patient's health using wearable and implanted devices has recently been gaining widespread adoption. A detailed account of the same was presented in [49], which reports a survey of the advances in Wireless Body Area Network (WBAN) systems and integrated technologies. This paper discusses several low-power wireless technologies that can be deployed. These sensor nodes can continuously monitor patients' health and collect vital data. The authors have detailed the usage of Bluetooth Low Energy (BLE) IEEE 802.15.4 and ZigBee, classic Bluetooth, ANT, RuBee, Sensium, and Zarlink among other low power consumption technologies. The authors also talk about intra body communication technologies that use wirelessly interconnected implanted devices. The importance of communication protocols is also highlighted. The utility of m-Health applications built on smartphones that help monitor patient parameters like temperature, pulse rate, and breathing rate is also shown. In the paper [50], the authors described the utility of IoT architectures for enhanced living environments and healthcare systems. They discussed at length open-source platforms like Kaa and Thingsboard, protocols like MQTT, smartphones, wearables serving as sensors, healthcare applications like the SPHERE project, and Home Health Hub IoT, as well as issues like Quality of Service, security, availability, compatibility, reliability, and future prospects. In the paper [51], the authors specifically addressed the issue of an enhanced monitoring scheme for patients admitted in critical condition. A framework using low-cost, low-power wearable sensors connected to the internet and using the open communication protocol oneM2M was outlined. Here, the wearables were used as Application Dedicated Nodes (ADN) that communicate with an infrastructure node containing the common services entity. Furthermore, openEHR has been used in the higher layers for providing functionality, namely data semantics, storage, and monitoring. The experiments report a latency of 20–50 ms, and a 30–50 h sensor autonomy. They also discussed the design of efficient M2M-capable sensors that could provide the benefit of low cost, and energy requirements through ESP8266 Wi-Fi modules. A multilayer fusion of Convolutional Neural Networks (CNNs) was proposed to detect Electroencephalogram (EEG)-based pathology detection in [52]. The fusion was done using a multilayer perceptron or an autoencoder. The experiments were performed in a smart healthcare framework. Around 90% accuracy and 97% specificity were obtained by the proposed system. Several papers have talked about an important concern related to the advancement of Wireless Body Area Network (WBAN) technology, which is the energy consumption of wearable and implanted sensor nodes. In [53], the authors proposed a solution to the joint scheduling and admission control problem to optimize the energy consumption by the gateway node and WBAN sensor node. The authors used the constrained Markov decision process in their approach and the Lagrange multiplier to arrive at a solution. Through simulation, the method gave rise to a 100% throughput improvement and reduced the power consumption by nearly 5.5-fold. With an enormous amount of data being generated, it has become challenging to store, access, and analyze the data in a time-bound manner to stimulate prompt actions that can preserve patient health and also reduce the cost of treatment. It is known that IoT devices have limited storage and computational capabilities. For the same, deep learning-based solution [54] may prove to be of immense significance.

In [55], the authors illustrated the possibility of making preliminary diagnoses of diseases from home. This achievement is due to the integration of telemedicine and new artificial intelligence techniques which are more effective and less complex in nature. In particular, one of the most important health monitoring concerns is heart disease, which, to date, is the leading cause of mortality [56].

Another topic studied in the literature of smart Healthcare in smart city is the automatic recognition of heart disease [57–63]. Cardiovascular disease refers to various malfunctions of the heart or blood vessels. The main ways to ascertain the proper functioning of the heart or detect certain cardiac diseases are: cardiac auscultation using the stethoscope and PCG, ECG by applying electrodes to certain points of the body and recording the electrical impulses generated by the heartbeat.

Several researchers have conducted multiple studies employing segmentation and classification algorithms of PCG [64–67] and ECG [68, 69] signals to recognize cardiac pathologies.

Monitoring and management of health risks in smart cities is another goal for successful smart cities. In this PhD period, more attention has been given to the recognition and management of heart disease using ECG/PCG signals and machine learning and deep learning techniques.

Chapter 3

Identification of missing persons in natural disasters based on drone-femtocell systems

In recent years advanced methods and alternatives to traditional ones have grown in public interest, for the identification of missing people in situations of catastrophic phenomena of considerable importance such as: earthquakes, landslides and collapses of structures frequented by people, floods, etc. [70–72]. In these scenarios, although civil protection agencies play a key role for those who have been affected by these disasters, often their intervention is ineffective. In these circumstances it is important to have innovative techniques and tools able to carry out an effective, precise and fast search for missing persons [73–75].

For this purpose, several algorithms for locating devices under rubble using a Femtocell-UAVs system were presented during the PhD activity. In general, the proposed algorithms involve two phase:

1. Classification of the internal/external terminals in the monitoring area;

2. Localization of the terminal within the monitoring area.

In the first phase, the focus is on discriminating the mobile terminals inside or outside the considered monitoring area. This is characterized by one or more devices under the rubble, so it is probably represented by an area of collapsed buildings. The proposed technique involves modeling the monitoring area, for simplicity, incorporated within a three-dimensional parallelepiped, of a given height h, thickness y, and depth x. The x and y parameters represent the dimensions of the area to be monitored, while with h, the rubble height is indicated within the monitoring area, or the minimum height of the floor above the monitoring area to which the drone can fly without encountering obstacles.

In the second phase, the focus is to locate the devices that were classified within the monitoring area in the first phase, in the shortest time and with high accuracy (error of about 1-3 meters).

The following sub-sections will illustrate the different localization techniques studied during the PhD period. In particular, sub-section 3.1 briey summarizes the main localization and positioning techniques followed by sub-section 3.2 which illustrates the radio mobile signal propagation and material attenuation; sub-section 3.3 describes the propounded smart femtocell-UAV data sensing system. Sub-section 3.4 describes an initial technique for classifying devices inside or outside the monitoring area. A scenario with non-ideal attenuation (no free space) is considered, where the non-ideality is due to the presence of the soil, to the height of the TX and RX antennas, multipath, positioning of the smartphone and the femtocell. In addition, the monitoring area is represented as a parallelepiped within which are the terminals to be located. Classification performance is evaluated by performing a series of simulations by varying the values of height, width and depth of the monitoring area, height of the drone and height of the terminals inside and outside the monitoring area. Sub-section 3.5 describes an first technique for locating devices within the monitoring area. The scenario considered is the same as that introduced in sub-section 3.4. The proposed technique incorporates several localization algorithms that are, subsequently, applied in a real scenario. Real power data are, in fact, collected and mapped onto a “power wall” and the device localization error is evaluated based on the introduced algorithms. Sub-section 3.6 describes the evolution of the first classification technique, considering a non-ideal propagation scenario with non-uniform material attenuation and non-uniform distribution, within the monitoring area. A series of simulations are launched in order to evaluate the classification accuracy of devices inside or outside the monitoring area by varying different parameters: material density, additional material attenuation, signal transmitting power, etc. Sub-section 3.7 describes the phase following the classification phase introduced in 3.6. Through two main localization algorithms: Proximity Method and Cluster-based Method, the localization error is evaluated varying some parameters. Sub-section 3.8 introduces a localization method that makes use of GT to optimize drone flight time, energy, and trajectory for each localized device. Also in this study, several simulations and experimental tests are carried out in order to optimize the efficiency of the femtocell-UAV system.

3.1 State of the Art on Localization Techniques

Figure 3.1 shows the current wireless localization techniques in the scale/resolution plan [76]. The area of the graph with the resolution of a meter and in an outdoor staircase (including rural and remote) is not currently managed and covered by any technique.

Some main radiolocalization methods are distinguished:

- Signal strength based (SSoA);
- AoA (/DoA/DF) (Angle of Arrival/Direction of Arrival/Direction Finding);
- ToA (Time of Arrival);
- TDoA (Time Difference of Arrival);
- Hybrid techniques (a combination of two or more of the previous ones).

The localization of a mobile terminal by RF signal is a topic often addressed in the literature [77–79]. In fact, there are several studies that deal with localizing terminals which are found in disaster areas or more generally in the presence of obstacles, through the analysis and measurement of radio signals. In [80], some different types of attenuation of RF signal were studied. The attenuation of the signal due to the presence of rubble and, therefore, of different types of materials is estimated, considering a signal at 1.8 GHz. The results of this study show, that in a post-earthquake disaster scenario, the losses, compared to free space, are 13 dB greater than the losses that occur in an indoor environment, which are equal to about 5 dB. In [81], the frequency response of the radio channel is studied on different frequency ranges, and the signal attenuation is measured for two types of material: ceramic and brick. First, the 1.8 GHz attenuation is about 4.5 dB more than 900 MHz in the case where the obstacles are arranged evenly around the receiving antenna, and 17.5 dB more when the obstacles are arranged in a less uniform way around the antenna. This implies that a higher frequency signal suffers more attenuation and that the obstacles placed in an irregular manner induce even more attenuation, compared to the case in which they are arranged in a more uniform manner.

Thanks to the features of high mobility and easy deployment, many studies use drones as a means of making on-demand communication services provision possible [82, 83].

More recently, the growing need to connect and cover areas affected by natural disasters has led to the commissioning of multiple studies concerning the use of

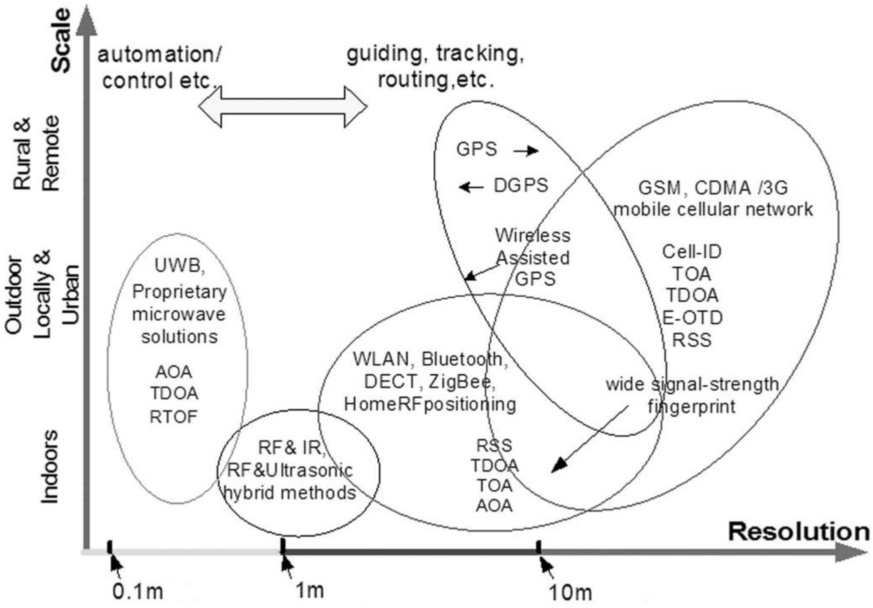

Scale
automation/ control etc.
guiding, tracking, routing,etc.
Rural & Remote
Outdoor Locally & Urban
Indoors
GPS
DGPS
Wireless Assisted GPS
GSM, CDMA /3G mobile cellular network
Cell-ID
TOA
TDOA
E-OTD
RSS
UWB, Proprietary microwave solutions
AOA
TDOA
RTOF
WLAN, Bluetooth, DECT, ZigBee, HomeRFpositioning
wide signal-strength fingerprint
RF& IR, RF&Ultrasonic hybrid methods
RSS
TDOA
TOA
AOA
Resolution
0.1m
1m
10m

Figure 3.1: Currently existing wireless localization techniques.

drone-femtocell systems as an alternative to the classic radio base stations when these are out of service [75, 84–87].

In [88], the optimal altitude of the UAV-based base station was analyzed for maximal communication coverage.

In [89], an efcient UAV 3D placement with the purpose of maximizing the covered users based on the optimal altitude was proposed. In [90], the authors studied a novel 3D UAV placement with the objective of maximizing the number of covered users according to different requirements of QoS.

3.2 Radio Mobile Signal Propagation and Material Attenuation

The transmission of signals in the mobile radio environment is obtained by means of a transmitting antenna, which emits the signal, and a receiving antenna, which receives it.

The mobile terminal, therefore, provides service within the coverage range of a base radio station to which it is docked. In general, the signal can be shielded from the environment, physical obstacles or artefacts limiting the diffusion/propagation of electromagnetic waves.

In particular, three totally different circumstances must be distinguished:

- Signal propagation in a vacuum;

- Signal propagation within the Earth's atmosphere;
- Signal propagation within the Earth's atmosphere and in the presence of materials.

3.2.1 Signal Propagation in a Vacuum

The femtocell generates an electromagnetic wave which in free space presupposes the following properties:

- Isotropic and homogeneous medium;
- Medium without losses;
- Transmission without obstacles or reflections.

In this case the only attenuation of the signal is that due to free space, and is expressed in (3.1):

$$A_0 = \left(\frac{4\pi d}{\lambda}\right)^2 \tag{3.1}$$

while the transmission power is given by the Friis's formula (3.2):

$$P_T = P_R G_T G_R \left(\frac{\lambda}{4\pi d}\right)^2 \tag{3.2}$$

where:

- P_T is the power transmitted by the femtocell;
- P_R is the power received from the mobile terminal;
- G_T, G_R are the antenna gains, respectively, in transmission and reception;
- λ is the wavelength of the electromagnetic signal;
- d is the distance between antenna in T_X (transmitter) and R_X (receiver).

Formula (3.2) is not applicable to real cases where there is a non-isotropic medium that attenuates the signal.

3.2.2 Signal Propagation within the Earth's Atmosphere

In this case, the situation is much more complex. Since the air we breathe is neither an isotropic nor a homogeneous medium, and since there are other obstacles (hills, palaces, trees, rain, fog, snow etc.) that shield an electromagnetic radio-mobile signal, the connection is often rendered possible only through reflections and diffraction.

Given that propagation conditions in the atmosphere strongly depend on the type of environment, it is very difficult to find an equation of the trafficking for mobile radio systems since, these waves, are continuously subject to attenuation, dispersion, reflection, refraction and diffraction.

3.2.3 Signal Propagation within the Earth's Atmosphere and in Presences of Materials

In this case, the scenario becomes much more complex, as the propagation takes place in the terrestrial atmosphere, where Friis's formula (3.2) is no longer valid, and through obstacles made of different types of materials, different shapes and densities.

The study of the signal attenuation effect in presence of different materials has been the subject of numerous scientific publications.

For example, [79], presents a study of the attenuation that a WiFi radio signal

Table 3.1: Attenuation for different obstacle.

Obstacle	Attenuation [dB]
Human Body	3
Cubicles	$3 \div 5$
Window, Brick Wall	2
Brick Wall next to a Metal Door	3
Glass Window	2
Office Window	3
Plasterboard Wall	3
Marble	5
Glass wall with metal frame	6
Dry Wall	4
Office Wall	6
Brick Wall	$2 \div 8$
Concrete Wall	$15 \div 20$
Metal Door	6
Metal Door in brick wall	$12 \div 13$

undergoes when it crosses various obstacles (panels) of different materials, placed at various distances. It was found that the losses depend on the distance between the transmitter and the panel, as well as the number of panels and panel material.

This dependence has a non-linear character and, therefore, it is difficult to describe it with existing methods of modelling the propagation of radio waves within a transmission medium.

Instead, an overview of the attenuation of some materials that can compose a building is provided in [91]. Also, in this study, the frequency in question is that of the 2.4 GHz WiFi signal. Table 3.1 demonstrates the attenuation presented by some types of materials.

3.3 A smart Femtocell-UAV Data sensing System

The elements of the data sensing system proposed in this studies are integrated as shown in Figure 3.2.

The system includes a drone with a femtocell aboard and a Raspberry Pi board to locally perform an intelligent analysis of the data, which are subsequently sent to a client (e.g. a tablet) to allow for visual analysis by an operator who determines the coordinates relative to the presence of a mobile terminal.

Considering the power supply and payload this system is fully sustainable, as:

- The drone allows a total payload of 2.5 kg, and a fully programmable control unit;
- The load on the drone is made up of the femtocell (550 g), a power bank to power it (550 g) and a processing unit (100 g), so the total weight of the load is 1.2 kg;
- The drone [92] supports the autonomy of 40 minutes with a load of 1.55kg and the amperage 7.67 Ah per battery (it requires 2 batteries);
- In the instance of multiple terminals or larger monitoring areas it is necessary to replace the batteries or organize multiple flights with multiple drones.

Considering the functionality of the system, the femtocell receives the power value RSRP measured by the terminal, in a common control channel. The power data, relative to each individual terminal connected, is sent to a single board computer (e.g. Raspberry Pi) via an Ethernet cable connected to the LAN port of the femtocell and to the Ethernet port of the Raspberry.

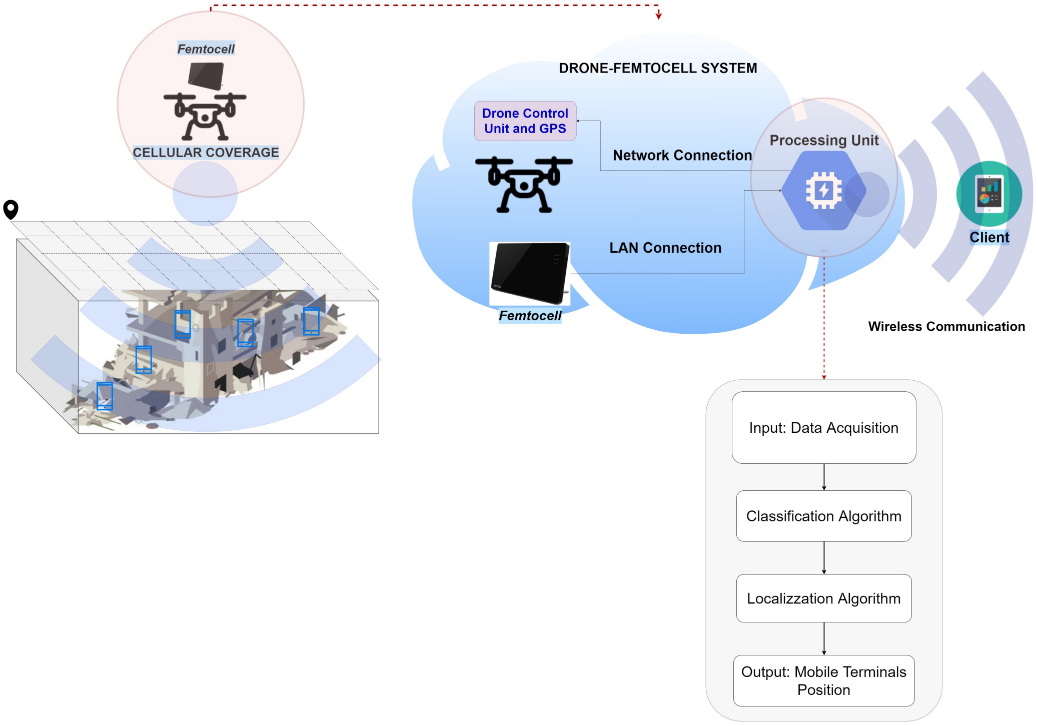


Figure 3.2: Connection/communication scheme of the proposed data sensing system.

To obtain information about the points of the area with preset GPS coordinates the drone must reach to make power measurements, it is necessary to connect the Raspberry to the drone's GPS coordinate system via network interface.

This is how it is possible to associate the position assumed by the drone with the power value that the femtocell receives at that particular point. The data collected at the end of the process are analyzed, the classification algorithm is applied to them, and, subsequently, geolocation is carried out. The analyzed information is subsequently sent to a client via a wireless network generated by the WiFi module inserted in the Raspberry Pi, which also acts as an access point for the clients.

3.4 Classification Algorithm Supposed 3D Monitoring Area in no Free Space

As studied in [93], in a generic scenario characterized by multiple monitoring areas, the first step is to verify the possible presence and the number of mobile terminals for each monitoring area, so as to select the areas of interest and immediately conduct the search by civil protection bodies.

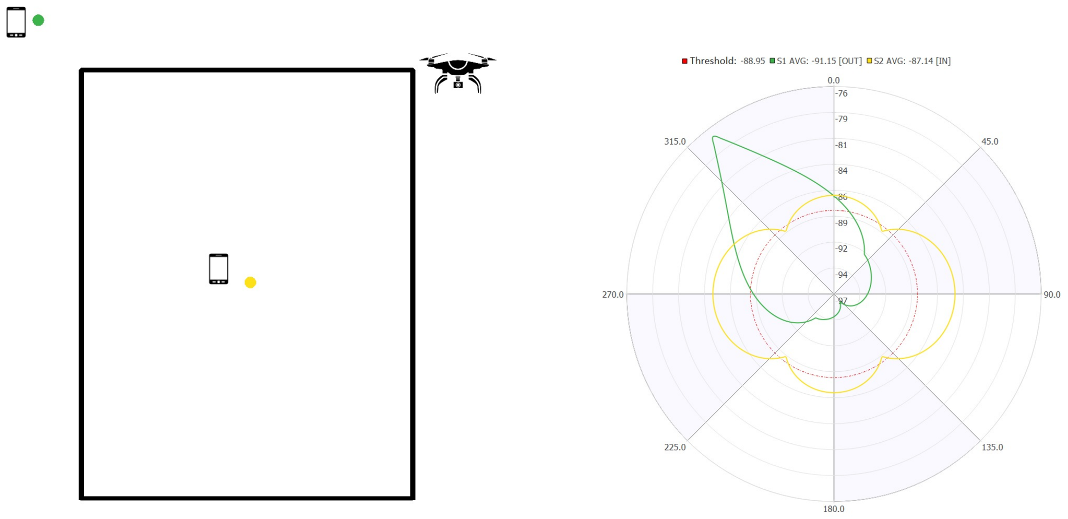


Figure 3.3: Polar diagram of the measured power relative to the $i-th$ monitoring area.

Assuming an ideal scenario, in the presence of free space, the proposed classification technique involves having the drone-femtocell system go around the entire perimeter of the $i-th$ monitoring area, measuring on the femtocell, with a step chosen a priori, the value of RSRP power received from the terminal. Figure 3.3 shows an example of a polar graph of the power levels obtained in the hypothesis of an internal terminal (yellow curve) and an external one (green curve) to the monitoring area.

The circumference in red represents the value of the discrimination threshold to distinguish whether the related terminal is external or internal to the monitoring area. In particular, the threshold is calculated considering the mean $RSRP_{Threshold}$ of the power measurements obtained by the drone-femtocell system that runs along the entire perimeter of the monitoring area, with respect to a mobile terminal placed in a corner of the monitoring area.

Given a mobile terminal, the relative average power on the perimeter of the monitoring area can be calculated, where the classification criterion is based on the following relation:

- For $RSRP_t^{AVG} < RSRP_{Threshold}$ the device is classified as OUT;
- For $RSRP_t^{AVG} \geq RSRP_{Threshold}$ the device is classified as IN.

The algorithm is applied to all the terminals that can engage the femtocell, keeping in mind that this allows determining their unique identification iteratively from the monitoring area $n = 0$ to the area $n = N$. It is important to point out that the power calculation refers to values obtained through the Friis's formula (3.2)

To verify the reliability of this technique it is necessary to apply it in real scenarios, in which the terminals (of all the monitoring areas) are subject to the non ideal conditions of the system even when they are in free space, and even worse when they are covered by rubble with more or less density. To analyze this more complex scenario, several hypotheses have been formulated.

The possible cases to be studied are the following:

- Attenuation from non-ideal free space: the non-ideality is due to the presence of the soil, to the height of the T_X and R_X antennas, multipath, positioning of the smartphone and the femtocell;
- Uniform attenuation due to a single type of material;
- Not Uniform attenuation due to a single type of material;
- Uniform attenuation (various materials).

In this sub-section, the first case is taken into consideration, analyzing the results even when they vary:

- The height, width and depth of the monitoring area;
- The height of the drone;
- The height of the terminals inside and outside the monitoring area.

Considering the complexity of the analysis to be performed, certain hypotheses have been introduced:

- The path followed by the drone, with the femtocell on board, is along the perimeter of the surface of the parallelepiped (which represents the area to be monitored) in steps of one meter;
- It is assumed that the femtocell has an isotropic propagation and an infinite coverage radius, thus allowing the coupling of all the internal and external terminals to the monitoring area;
- The height of the drone, in meters, is equal to:

$$H_D = H_R + 0.5, \tag{3.3}$$

  where H_D is the height of the drone with the femtocell on board; H_R is the maximum height level of the rubble.

In an initial phase, we used exclusively the average power measured for terminal discrimination, which is calculated in (3.4):

$$AVG = \frac{1}{N}\sum_{i=0}^{N} P_{R_i}, \tag{3.4}$$

where N is the number of measurement points taken on the perimeter of the monitoring area and P_{R_i} is the simulated power value or the one obtained from field tests. In order to simulate the power value of P_{R_i} of the $i-th$ measurement point, the Friis's formula (3.2) is used.

The choice to use only the power average parameter makes the system not very stable to the attenuation and multipath effects, for this reason two other parameters have been added:

- Normalized variance;
- First autocorrelation coefficient.

For the calculation of the first parameter a preliminary step is carried out. It works by taking the lowest power value (remember that the power in dBm has a negative sign), calling it $P_{R_{min}}$ and normalizing the other power values with respect to this, in (3.5):

$$NVAR = \frac{1}{N}\sum_{i=0}^{N-1} \left(P_{norm_i} - AVG\right)^2, \tag{3.5}$$

where P_{norm_i} is:

$$P_{norm_i} = \frac{P_R}{P_{r_{min}}}, \tag{3.6}$$

the calculation of the second parameter, first autocorrelation coefficient, is given by (3.7):

$$\rho = \frac{CORR_1}{CORR_0}, \tag{3.7}$$

where:

$$CORR_k = \sum_{k=0}^{N-1}\left\{\frac{1}{N}\sum_{t=0}^{N-k-1}\left[\left(P_{r_t} - AVG\right)\left(P_{r_{t+k}} - AVG\right)\right]\right\}, \tag{3.8}$$

to take into account the type of femtocell and the coverage levels offered, the algorithm is based on a Friis's formula "Compensated". The compensation lies in adapting the value of the variance obtained from the measurements in the field with that of the measurements obtained from the simulator that uses the Friis formula in free space. The balanced Friis formula, therefore, derives from (3.9):

$$P_r^{Comp} = \alpha P_r + (1-\alpha)\,\bar{P}_r, \tag{3.9}$$

where $\alpha = 0.52$ (the value was chosen based on the measurements made in the field); P_R is the power of Friis's formula, while $\bar{P}_r$ is the average of the values given by the Friis formula.

To this compensated formula is added an error with Gaussian distribution, which represents the non-ideality of the system (in particular the not perfectly isotropic propagation and the measurement error). The procedure underlying the generation of the Gaussian error is the following:

- A test of real measurements is considered, in which there are four terminals: two inside and two outside the monitoring area. The four terminals measure a power value (RSRP), respectively, as the femtocell position changes along the perimeter of the monitoring area;
- In the same conditions the measurements are generated, by the simulator, for each terminal;
- The difference is made between the values generated and those measured for each terminal (obviously with the same position in the monitoring area);
- The standard deviation of the differences is calculated;
- An average of the four calculated standard deviations is calculated;
- The resulting value corresponds to the range of the Gaussian error (E_g), which in our tests goes from [-1.39,1.39].

The final formula for generating power values, in non-ideal conditions, in the simulator derives by (3.10):

$$P_R^f = P_r^{Comp} + E_g, \tag{3.10}$$

Sub-section 3.4.1 describes the classification algorithm that uses a single threshold, called the critical threshold, calculated by assuming the terminal in a corner of the perimeter, and, moreover, the performances are evaluated by varying some parameters.

Similarly, in sub-section 3.4.2 the algorithm is described using two thresholds, one positioned in the corner of the monitoring area, i.e. the critical threshold, and the other positioned in the center of the monitoring area, the latter being defined as "threshold centroid".

In sub-section 3.4.3 we modify this hypothesis and we consider that the femtocell is placed on a mobile robot and that the height that can be reached varies from one to two meters.

3.4.1 Single Threshold Method

In the first phase, the threshold was considered assuming a mobile terminal placed on the ground (0 meters), in correspondence with an edge of the monitoring area. The parameters used for the implementation of the classification algorithm are:

- Power average;
- Normalized variance;
- First autocorrelation coefficient.

The algorithm is defined by the combination of the three parameters described above. If the classification parameters of the first terminal are smaller than the second then the output to the logic chain will have a Boolean variable set to TRUE, otherwise - to FALSE. The block diagram of the classification algorithm is showed in Figure 3.4:

If the output of the AND gate is TRUE it means that the terminal is correctly classified IN, otherwise, if the output is FALSE, the terminal is classified as OUT.

Performance evaluation takes place with the following parameters:

- Drone and rubble quota: please note that these two parameters are linked by the formula (3.3) expressed above;
- Monitoring area size.

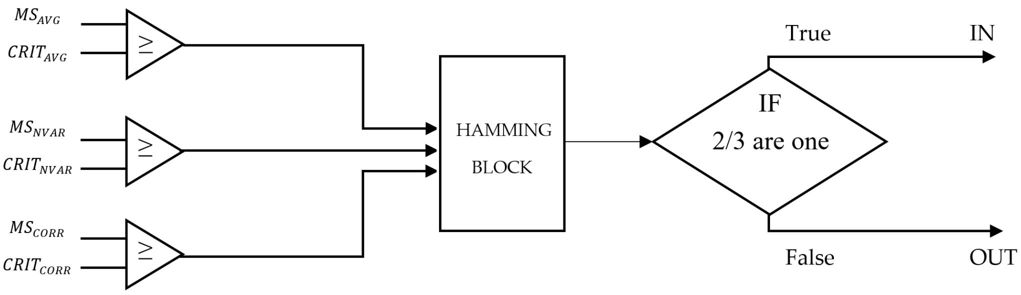


Figure 3.4: Block diagram of the critic threshold - based classification algorithm.

Before proceeding with the performance evaluation, it is important to introduce the definition of some terms:

- "Monitoring Area", is the area in which it is necessary to check how many terminals are inside and how many are outside;
- "Critic Zone", is the area outside the monitoring area where it is possible to find false positives, such as terminals that are classified as IN but which, instead, are OUT;
- "Safe Zone", represents a security zone around the monitoring area that is assumed to have no terminals;
- "Rubble Level", is the maximum level of the rubble in a disaster scenario;

Regarding the first point, the accuracy of the IN and OUT terminals is evaluated by introducing a "Safe Zone" of 10% around the "Monitoring Area", i.e. 10% of the short side of the monitoring area. Furthermore, to take into account that the radius of coverage of the femtocell, in the real case, is of finite value, we have considered the possible presence of external terminals positioned up to 50 meters from the monitoring area (Figure 3.5).

In terms of percentage accuracy of the terminals correctly classified as IN and OUT it is clear that for the different dimensions of the monitoring area the accuracy of the IN is 100%, i.e. all the terminals that are IN are classified as such. Conversely, the accuracy of the OUTs, i.e. external terminals that are classified as such, is very variable, and this depends both on the size of the monitoring area

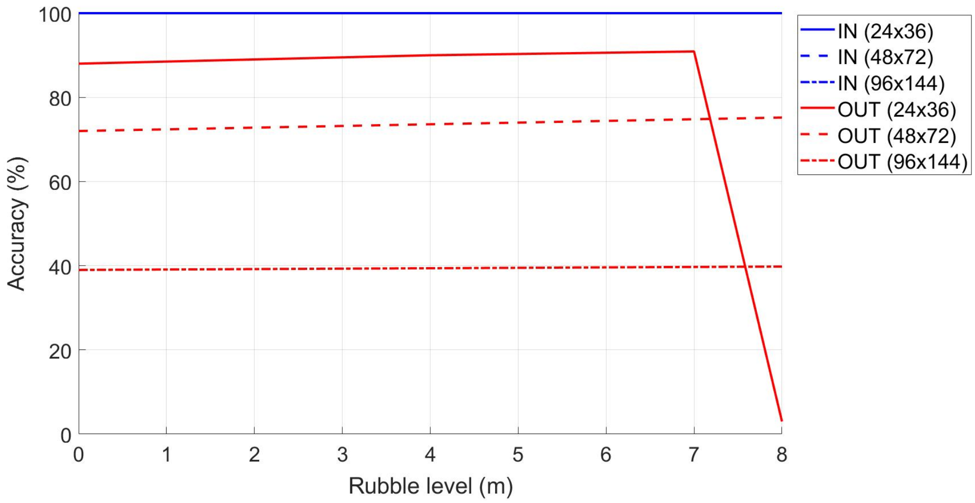


Figure 3.5: Accuracy in classification of IN and OUT terminals as the amount of rubble increases.

and on the level of the rubble. The accuracy decreases quickly after a certain flight height of the drone (or rubble) due to the greater visibility that the femtocell has from the mobile terminals outside the monitoring area (3.6).

In general, both the performance related to the extension of the "Critic Zone" and those relating to accuracy are very low, for this reason a second threshold terminal is introduced, placed at the center of the monitoring area at drone height. In this way, as we will see later, performance will improve.

Regarding the second point, as already indicated in (3.3), the hypothesis on the drone's flight height is that it exceeds the maximum height of the rubble level by half a meter, so that the drone could move freely within the whole monitoring area.

As it can see from this graph, the "Critical Zone" remains constant at 18 m and 30 m when the level of rubble varies, respectively for an area size of 48×72 m^2 and 96×144 m^2.

While, in the case of the area of 24×36 $[m^2]$, the "Critical Zone" shows a sudden increase when the rubble exceeds 7 meters in height. This means that the algorithm considers the mobile terminal as internal to the monitoring area, when in reality it is positioned externally. These false positives lead to the increase of the "Critical Zone" by about 40 meters, when the rubble level exceeds a certain threshold (in this case seven meters).

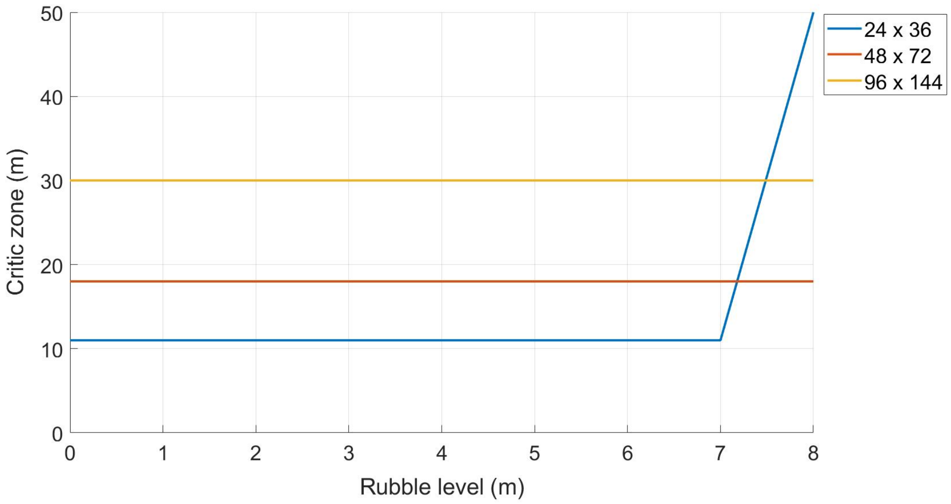


Figure 3.6: Variation of the "Critical Zone" as the amount of rubble increases.

3.4.2 Method with Hybrid Thresholds

This method has been implemented to increase accuracy performance. The classification algorithm performs the calculation of the classification parameters using separately the thresholds deduced from the analysis of the data relating to the two threshold terminals (critical and centroid) and, subsequently, applies a logical decision to establish whether the intercepted terminal is inside or outside the monitoring area. Schematically we can represent the set of logical instructions in Figure 3.7.

If at the end of the chain, the AND logic gate indicates TRUE the terminal will be classified as IN, otherwise it will be classified OUT (see Algorithm 1).

The variables "MS_{AVG}", "MS_{NVAR}", "MS_{CORR}" refer to the parameters of the target terminal. The variables "$CRITIC_{AVG}$", "$CRITIC_{NVAR}$", "$CRITIC_{CORR}$" refer to the threshold terminal at the critical point (the corner of the monitoring area) while the variables "$CENTR_{AVG}$", "$CENTR_{NVAR}$", "$CENTR_{CORR}$" refer to the threshold terminal in the centroid.

With this method the performance of the classification algorithm is improved compared to the previous method, however, there are still certain factors of non-ideality that affect its accuracy.

It is now possible to evaluate the performance with the new method as the parameters already described in the previous method vary and identify the differences and improvements.

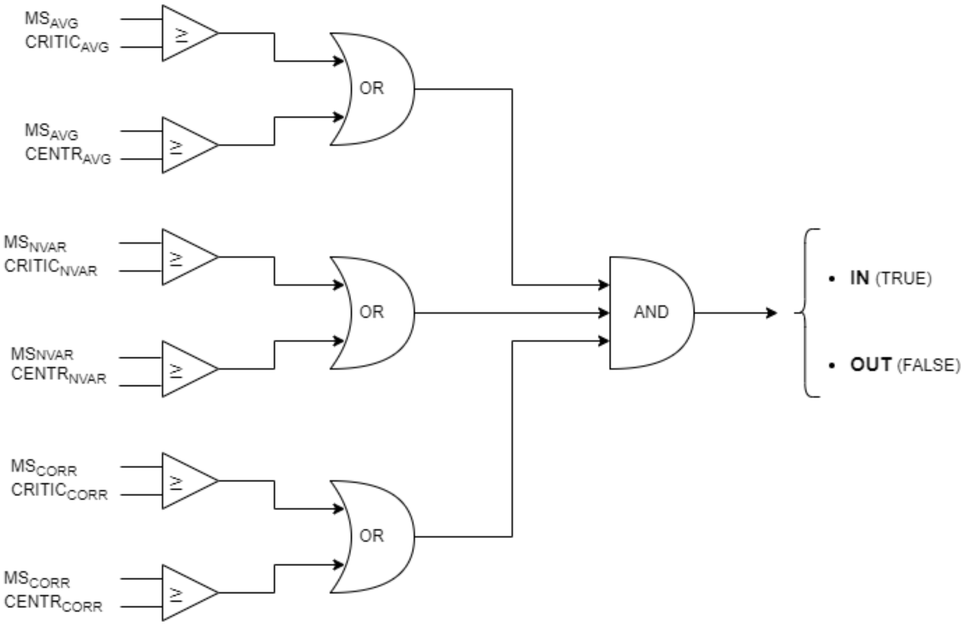


Figure 3.7: Block diagram of the hybrid threshold – based classification algorithm.

Algorithm 1 Algorithm for classification of terminals IN or OUT the monitoring area.

```
    Input: $MS_{AVG}, CRITIC_{AVG}, CENTR_{AVG}, MS_{NVAR}$
$CRITIC_{NVAR}, CENTR_{NVAR}, MS_{CORR}, CRITIC_{CORR}, CENTR_{CORR}$
    Output: $IN$, $OUT$
procedure
    $c_{AVG} \leftarrow (MS_{AVG} \geq CRITIC_{AVG}) \vee (MS_{AVG} \geq CENTR_{AVG})$
    $c_{NAVG} \leftarrow (MS_{NVAR} \geq CRITIC_{NVAR}) \vee (MS_{NVAR} \geq CENTR_{NVAR})$
    $c_{CORR} \leftarrow (MS_{CORR} \geq CRITIC_{CORR}) \vee (MS_{CORR} \geq CENTR_{CORR})$
    $c \leftarrow (c_{AVG} \wedge c_{NVAR} \wedge c_{CORR})$
    if $c \leftarrow TRUE$ then
        MS is IN.
    else
        MS is OUT.
```

As can be seen from the graph in Figure 3.8, compared to the previous method, the "Critic Zone" for each area appears to be smaller in size but grows slightly as the height of the rubble (and therefore of the drone) increases. In general, however, the performance improves, in fact, if the height of the rubble is 5 m and the area considered is 24 × 36 m, the "Critic Zone" is about 7 m, while in the previous case it is 11 m. Assuming the size of the monitoring area and the level of the rubble is known, it is possible to determine the size of the "Critical Zone" and therefore estimate a certain "Safe Zone", which reduces the possibility of false positive, i.e., mobile terminals incorrectly classified as IN.

When the femtocell is positioned at the height of the rubble (therefore depending

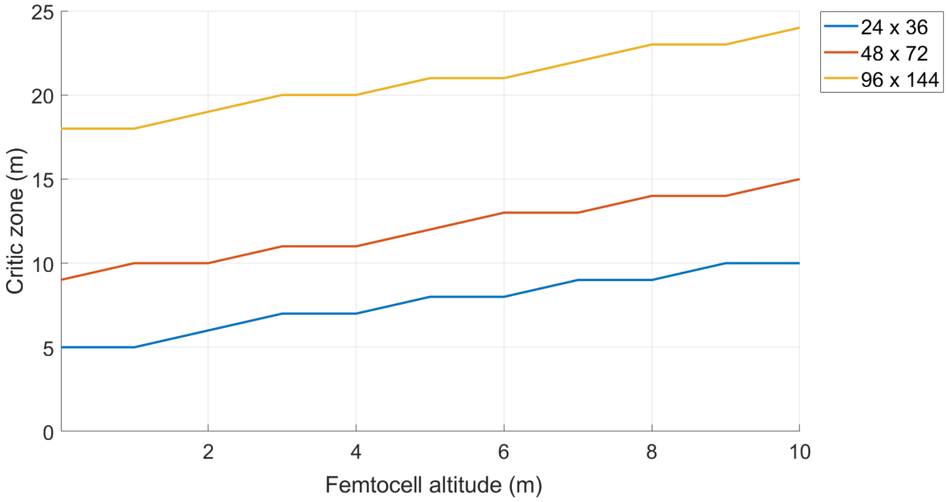


Figure 3.8: Variation of the "Critical Zone" to the increase of the femtocell altitude.

on the surrounding environment), the performance of the percentage accuracy relative to the OUTs depends not only on the actual dimensions of the "Monitoring Area" but also on the position in which the internal and external terminals are positioned. As shown in the figure, the percentage accuracy of the IN is a straight line with a slight slope (about 1% is lost every 10 m) and suffers little from the influence on the femtocell share elevation. The situation is considerably different with regard to the percentage accuracy of the OUTs which is significantly affected by the difference in altitude. In this regard, it is necessary to make an important clarification regarding what is meant by rubble quota:

- The rubble quota (indicated as H_R) identifies the maximum height of the rubble inside the "Monitoring Area";
- The OUT rubble dimension identifies the maximum height of the rubble surrounding the "Monitoring Zone" (i.e. those inside the "Critic Zone").

Graph in Figure 3.9, the maximum amount of rubble of 4 m is considered within the "Monitoring Area" while the area surrounding the "Critic Zone" has rubble at a maximum height of 2 m, the percentage of accuracy IN will be identified in the graph from the blue curve to the altitude 4 m, while the accuracy percentage OUT will be identified in the graph from the red curve at 2 m altitude.

To increase the percentage accuracy of the OUTs, a "Safe Zone" of 10% can be set up (calculated with respect to the short side of the "Monitoring Area"). In this way a significant improvement can be identified in the accuracy performance OUT (leaving unchanged those related to the classification of IN).

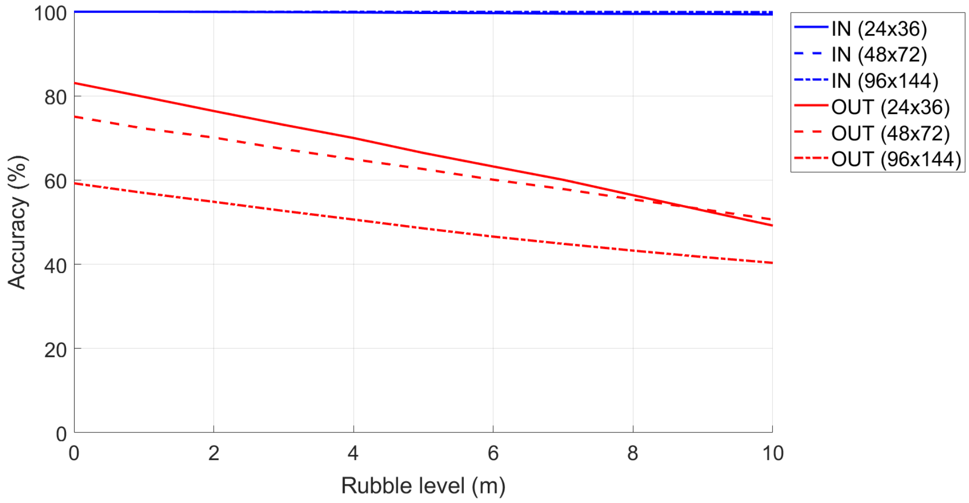


Figure 3.9: Accuracy in classification of IN and OUT terminals as the rubble level increases.

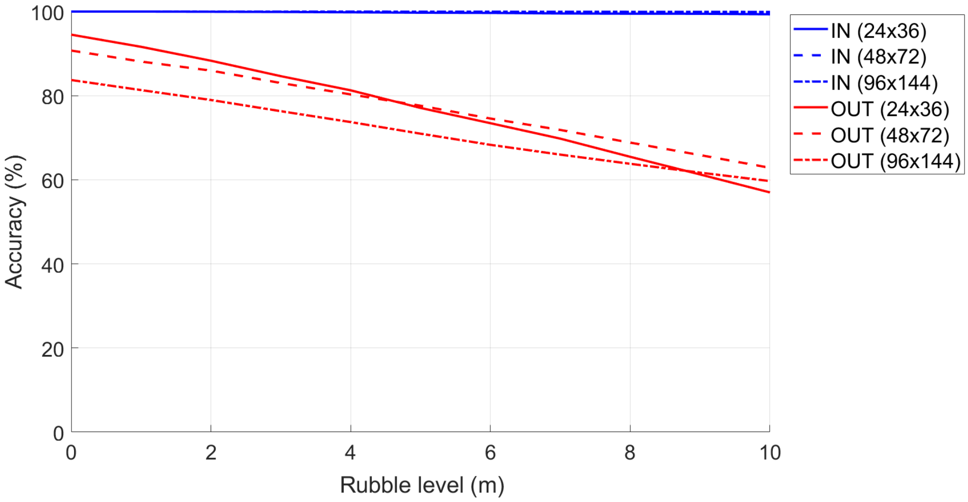


Figure 3.10: Accuracy in classification of IN and OUT terminals as the amount of rubble level increases with “Safe Zone” by 10%.

Compared to the first case, the performance of the terminals classified as OUT, for all the dimensions of the areas, turns out to be greatly improved. In fact, if we are at a rubble quota of 5 m, for the 96 × 144 m area, we have an accuracy of about 70% compared to the previous case which is 40% (Figure 3.10).

3.4.3 Method with Hybrid Thresholds and Femtocell Height at one Meter from the Ground

In the previous subsections an important hypothesis is that the femtocell is placed above the drone, where the latter flies at a certain height, given by the formula (3.3), along the perimeter of the surface of the monitoring area. In this subsection we modify this hypothesis and we consider that the femtocell is placed on a mobile robot and that the height that can be reached varies from one to two meters. The path will be along the walls of the parallelepiped.

The following are the performance scenarios with the new hypothesis. When the femtocell is positioned at a man’s height (by hypothesis 1 m), therefore independent of the surrounding environment, the performance of the percentage accuracy depends on the geometry of the “Monitoring Area” and the amount of rubble. However, in this case, the accuracy of OUT in percentage is increasing. Considering also a “Safe Zone” of 10% the trend is showed in Figure 3.11.

Comparing the two detection methods (at the rubble level and at a height of one meter) it can be observed that, with the same simulation parameters, the femtocell method at a height of one meter is more efficient. The only clarification

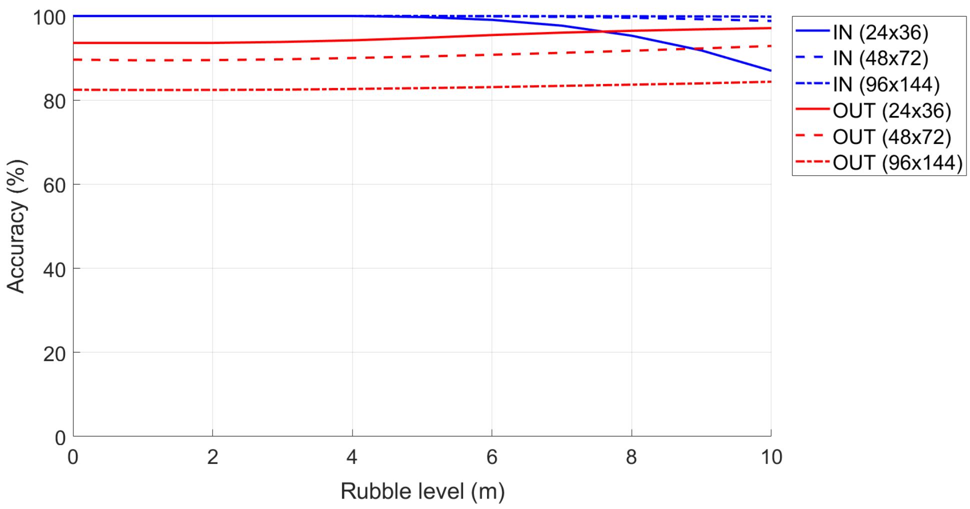


Figure 3.11: Accuracy in classification of IN and OUT terminals as the amount of rubble increases and femtocell height at one meter from the ground.

to make regards the decreasing trend of the IN accuracy is visible on the curve generated for a “Monitoring Area” of size 24×36 m. Furthermore, judging by the slight slope visible on the other two curves, one can guess that a similar behavior will be exhibited also for the “Monitoring Area” for larger rubble dimensions.

3.5 Localization Algorithm Supposed 3D Monitoring Area in No Free Space

As studied in [93], once the presence of one or more mobile terminals within a monitoring area has been verified, the second phase aims to estimate the position of the individual device. In fact the algorithm repeats the analysis for each terminal, thanks to the possibility that the femtocell offers us to be able to measure the power values unambiguously for each individual terminal.

A rectangular or cross-linked grid of uniformly distributed measurement points is considered to be placed exactly above the surface of the parallelepiped enclosing the monitoring area, where the drone, by moving, makes various measurements of the received signal strength. An example is shown in Figure 3.12, in which there is a 5×11 meter grid and the real position of the terminal is represented by the square in blue.

The power received by the terminal was subsequently mapped and interpolated, creating what has been called a “power wall”. An example of the latter is shown in Figure 3.13, which is the result of the power measured in the grid points in Figure 3.12, interpolated and mapped. The terminal is in the position marked by

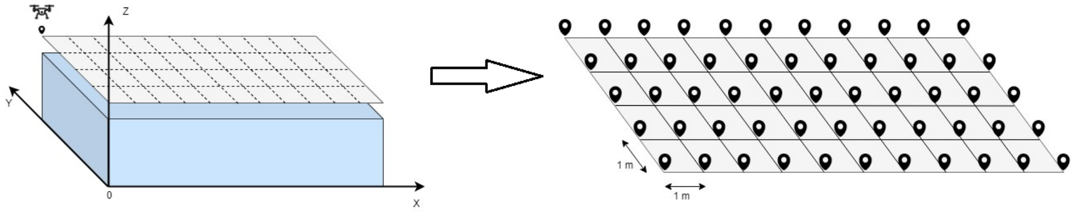


Figure 3.12: Rectangular grid of measurement points.

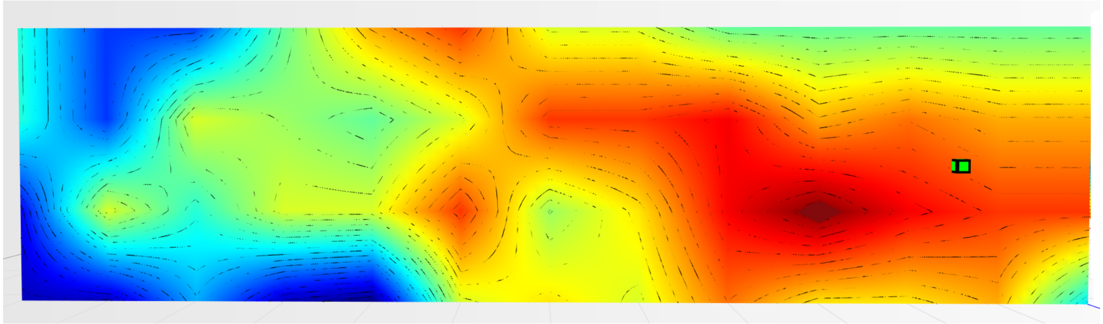


Figure 3.13: “Power wall” of the values measured in the grid points.

the green cube.

Starting from the data of the power measurements contained in a “power wall”, the proposed algorithm for estimating the position of the terminal was based on an aggregation of the following three estimation methods:

- “Method of proximity” to the highest power value;
- “Weighted distance method”;
- “Center of gravity method”.

The first estimates the position of the terminal in the position in which it has the highest power value. This algorithm is very efficient in scenarios where the terminal is in free space or when the rubble has a uniform distribution and attenuation over the entire area. In scenarios where the area does not have a uniform type of attenuation, this algorithm appears to be poorly suited to the context.

For this reason, the “weighted distance method” was introduced, which identifies the relative minimum points of the power levels. In fact, for certain types of material attenuation, the RF signal source points may be strongly attenuated, but in their surroundings still radiate the signal and, therefore, define the so-called “volcano mouths”, which, if recognized, allows for the mobile terminal that

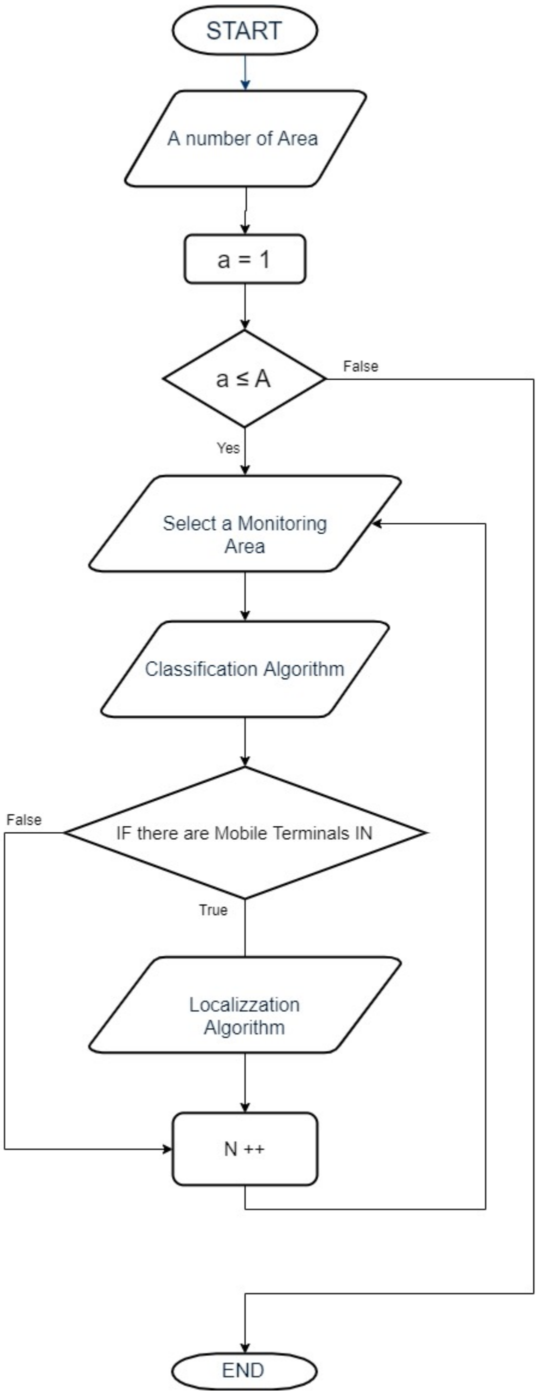


Figure 3.14: Flowchart 2D localization and classification technique.

is in the vicinity of these to be determined and not in the points where the power level is the maximum.

Finally, if the “power wall” shows three peaks of maximum power and no relative minimum, the “center of gravity method” is activated, which precisely estimates the terminals at the point defined by the formula of the center of gravity, where the three points that define the triangle are precisely the peak points of maximum power. The sub-section 3.5.1 describes the steps of the localization algorithm; the sub-section 3.5.2 describes the distance criterion; the three algorithms will be explained more specifically in sub-section 3.5.3, 3.5.4, 3.5.5. Finally, the test bed scenario is illustrated in seb-section 3.5.6. In general, the algorithm can be represented using the flow chart in Figure 3.14.

3.5.1 The Localization Algorithm Phases

Once the position of a mobile terminal has been classified as internal to the monitoring area, the next step is to estimate the exact positioning of each individual terminal within the selected area. In particular, three types of algorithms are illustrated in this second phase. The algorithm, which is based only on the method of proximity to the highest power value, as already mentioned, is not very efficient in more complex real scenarios.

In fact, due to different attenuation of the various materials around the terminal to be identified, it is likely that the mobile terminal receives more power along a lateral, and not orthogonal, propagation direction of the RF signal from the femtocell to the terminal, therefore, less power (and therefore greater attenuation) is detected in the area above the device and greater power in areas distant from the actual position of the mobile terminal. In this regard, an algorithm was designed to evaluate the position of the terminal, taking into account the presence of what we will call "mountains" and "lakes" that is, points where there are relative maxima and minima in the RF signal power values.

The concept around the definitions of "mountain" and "lake" is therefore based on the analysis of the monotony of isopotential surfaces extracted from the interpolated map of measured power levels. By grouping the surfaces and analyzing the trend of the power (from the internal to the external surface), it is possible to establish the direction of the power gradient. Depending on the direction taken, we defined:

- "Lake" when the innermost isopotential surface showed a growing monotonous tendency around a spatial position P_{L_i} called the "peak of the $i-th$ lake";
- "Mountain" when the innermost isopotential surface showed a decreasing monotonous tendency around a spatial position P_{M_j} called the "peak of the $j-th$ mountain".

Before describing the operation of the algorithm, we introduced a series of definitions and formulas necessary for calculating the terminal estimate:

- G_{max} is the maximum power value (dBm) detected in the entire matrix of points;

- G_{min} is the minimum power value (dBm) detected in the entire matrix of points;

- T is defined as "Earth level", and is obtained by (3.11):

$$T = \frac{G_{max} + G_{min}}{2}, \tag{3.11}$$

- M_{peak} is the power (dBm) related to the maximum peak of the mountain;

- L_{peak} is the power (dBm) related to the lake's peak;

- P_M is the position of the mountain's peak;

- P_L is the position of the lake's peak;

- δ_M is defined as "mountain weight" (i.e., the weight to be assigned to the global maximum point with respect to the "Earth" plan). This is defined by (3.12):

$$\delta_M = \frac{M_{peak} - T}{G_{min} - L_{peak}}, \tag{3.12}$$

- d_{LM} is the length of the segment that connects the peak of the "lake" with the peak of the "mountain";

- P_{Norm} is the normalization power, and is obtained by (3.13):

$$P_{Norm} = G_{Max} - G_{min}, \tag{3.13}$$

- D_{Norm} is the normalization distance, and is given by the diagonal of the considered rectangular surface. The formula is showed in (3.14):

$$D_{Norm} = \sqrt{w^2 + h^2}, \tag{3.14}$$

where w and h are the width and the height of the surface, respectively.

The algorithm optionally involves the selection of mountains and lakes with higher power levels. In particular, if the FLAG of the filter is active, we filter "lakes" and "mountains" with a certain percentage, defined by the operator, with respect to the maximum peak power of the "mountain" and the minimum of the "lake".

- If the number of "mountains" is equal to zero, then the algorithm does not provide any result;
- If the number of "mountains" is equal to one, then the following elements must be evaluated:
  - If there are no "lakes", the position is estimated by employing the proximity method;
  - If only one "lake" is found, the position is estimated by using the weighted distance method;
  - If several "lakes" are found, the "lake" with the lowest Mn value is selected (see (3.18)), then the terminal is estimated by using the weighted distance method;
- If the number of "mountains" found is equal to two:
  - If no "lakes" are found, the position is estimated from the midpoint of the two "mountain" peaks;
  - If only one "lake" is found, the position is estimated using the distance criterion described below;
  - If the number of selected "lakes" is greater than 1, the "lake" with the lowest Mn value is selected (see (3.18)), then the terminal is estimated using the distance criterion;
- If the number of "mountains" found is equal to three:
  - If there are no "lakes" the position is estimated using the center of gravity method;
  - If the number of "lakes" found is equal to one, the position is estimated using the distance criterion;
  - If the number of "lakes" found is greater than one, the "lake" with the lowest Mn value is selected (see (3.18)), then the terminal is estimated using the distance criterion;
- If the number of "mountains" selected is greater than three, the three highest "mountains" are selected and then instructions for the previous point are followed.

Figure 3.15 shows the positioning algorithm's flowchart.

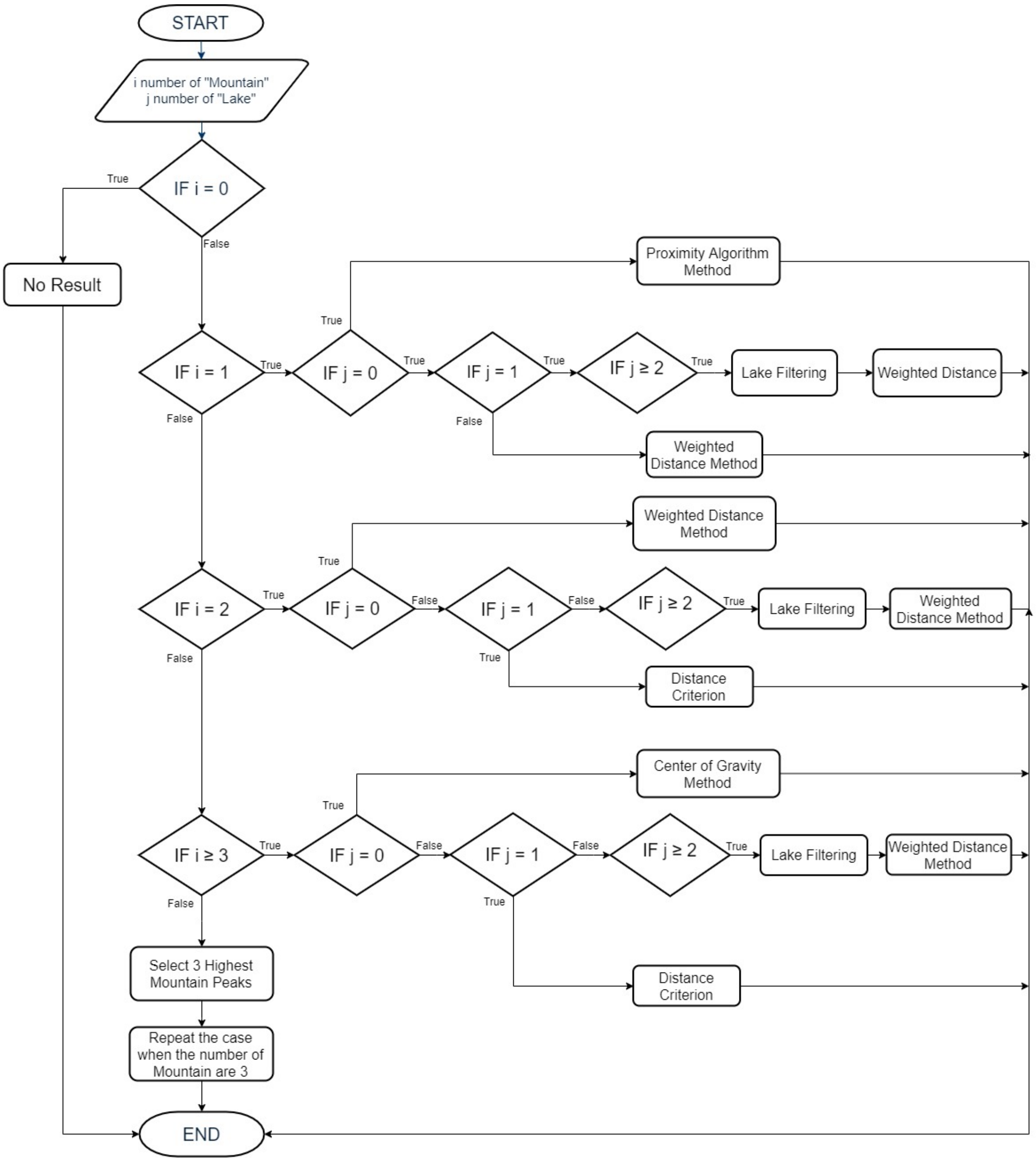


Figure 3.15: Flowchart of the positioning algorithm.

3.5.2 Distance Criterion

When there are two (or three) “mountains” and one “lake”, a pre-selection phase is activated through the distance criterion. Suppose further that the “mountain” list is sorted in decreasing power, so that $mountain_0$ is the highest power one.

- For a number of “mountains”, it is equal to three: if the selected “lake” is at a shorter distance from $mountain_0$ than in $mountain_1$ and $mountain_2$, the method of weighted distance between the “lake” and the $mountain_0$ is used;

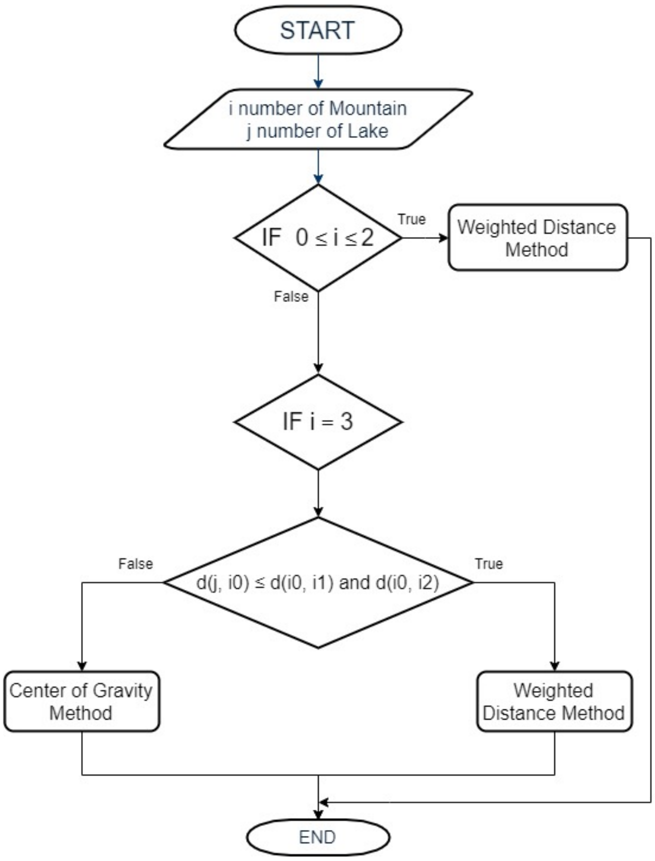


Figure 3.16: Flowchart of the distance criterion.

- If only one "mountain" (e.g., $mountain_1$) is at a shorter distance from $mountain_0$, then $P_{Estimated}$ is given by the midpoint between $mountain_0$ and $mountain_1$. If both "mountains" are at a shorter distance from $mountain_0$, then $P_{Estimated}$ is given by the "center of gravity method".

Figure 3.16 show the distance method's flowchart.

3.5.3 Proximity Method

In general, an estimate of the position based on the proximity method uses a proximity sensor to detect the presence of nearby objects without any physical contact. In particular, the proximity sensor in this paper is represented by the femtocell, which measures the power levels received from a mobile terminal.

The proximity method consists in estimating the terminal at the position P_M where there is the highest peak of power called the "peak of the mountain".

$$\begin{cases} P_x^E &= P_x^M \\ P_y^E &= P_y^M \\ P_z^E &= P_z^M \end{cases}, \tag{3.15}$$

$$P_{Estimated}\left(P_x^E, P_y^E, P_z^E\right) = P_M\left(P_m^M, P_y^M, P_z^M\right). \tag{3.16}$$

3.5.4 Weighted Distance Method

The weighted distance method is articulated in the following steps. Firstly, in order to process the data, we must look for all of the "mountains" and "lakes" on the surface. Then, we can select the highest "mountain" peak.

1. If no "mountain" is detected, no processing is possible and the routine is aborted;

2. If no "lakes" are found, but there is at least one "mountain", then the terminal is estimated in the position where the highest "mountain" peak lies;

3. If only one "lake" is found, the position is estimated using (3.17):

   $$P_{Estimated} = \delta_M \cdot d_{LM} \cdot \vec{u_{LM}}. \tag{3.17}$$

   where $(\vec{u_{LM}})$ is the directional versor that goes from the peak of the "lake" to the peak of the "mountain".

4. The more "lakes" are found, the less probable steps must be taken. To do this, the distance between the peak of the "mountain" (i.e., the highest power value) and the minimum point of each identified "lake" must be calculated and the one with the lowest M_n value must be chosen, where M_n is given by (3.18):

   $$M_n = 0.5 \cdot \left(\frac{D_{L_{i-M}}}{D_{Norm}}\right) + 0.5 \cdot \left(\frac{D_{L_{peak}}}{P_{Norm}}\right), \tag{3.18}$$

   where $D_{L_{i-M}}$ is the distance between the $i-th$ "lake" and the selected peak of the "mountain", and is obtained by (3.19):

   $$D_{L_{i-M}} = \left\| P_{L_i} - P_M \right\|. \tag{3.19}$$

Once a single "lake" is selected, proceed following the instructions outlined in point 2.

3.5.5 Center of Gravity Method

This method is applied when only three "mountains" are found. In particular, this consists of considering the three peaks of "mountain" as the three vertices of a triangle and calculating the center of gravity (3.20):

$$\begin{cases} P_x^E & = \frac{P_x^{M_0}+P_x^{M_1}+P_x^{M_2}}{3} \\ P_y^E & = \frac{P_y^{M_0}+P_y^{M_1}+P_y^{M_2}3}{} \\ P_z^E & = \frac{P_z^{M_0}+P_z^{M_1}+P_z^{M_2}}{3} \end{cases}, \tag{3.20}$$

3.5.6 The Test Bed Scenario

The following hardware and software tools were used for simulations and laboratory tests:

The hardware and software tools described in the Appendix A were used in the testbed for simulations and laboratory tests; in addition part of the technological center of the University of Catania was used during the algorithm validation phase. The choice was dictated by the possibility of having a structure characterized by different types of material such as concrete blocks, glass windows, external lava stone, wooden, and iron doors. Each material, as is known, has a different attenuation coefficient, and therefore it has been possible to simulate situations of isotropic and non-isotropic propagation, obtaining cases with mountains only (power peaks), but also in cases with the presence of lakes (signal sources' RF that radiate laterally due to the high attenuation of the material along the direction orthogonal to the position of the mobile terminal to be geolocalized). In particular, for the geolocation tests, the mobile terminals were positioned inside the laboratory rooms along an external edge of the structure. The measures were carried out through a trestle adjustable in height where the femtocell was positioned, and considered a grid of 4 × 12 points equidistant to $1m$ to cover the external walls. The data obtained from the power measurements were interpolated by obtaining images representing the so-called power walls used as input to the classification and geolocation algorithms described in the previous sections. In order to evaluate the effectiveness of the algorithm, two adjacent walls of the technological pole building of the University of Catania were considered (see Figure 3.17), and the terminal estimation algorithm was applied to the obtained data (Figures 3.18 and 3.19).

It was decided to indicate the real position of the mobile terminal as a green cube, its estimated position in white, the peak of the "lake" in red, and, finally, the peak of a "mountain" in blue. This alternating choice of colors serves to discriminate, in the image, the position of the peak of the "mountain" and the peak of the "lake". It is important to note that the alternating choice of colors does not affect the color that indicates the highest and lowest power. In fact, in

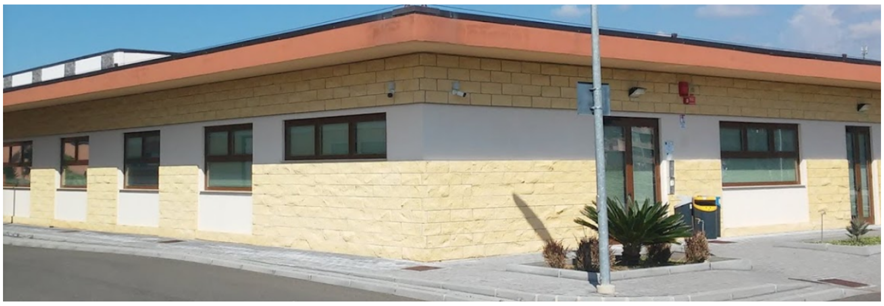


Figure 3.17: Technological pole building.

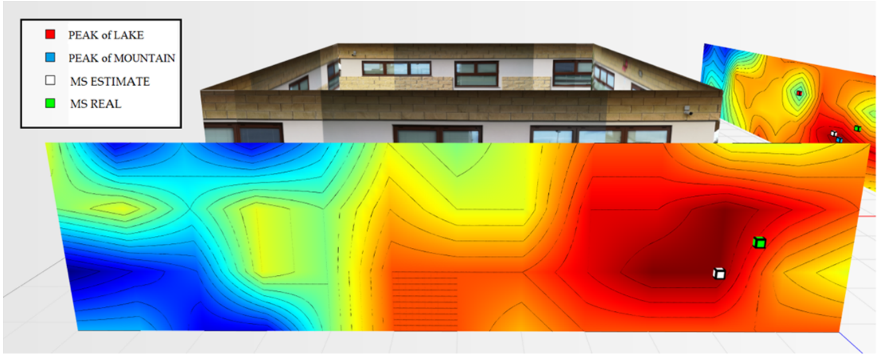


Figure 3.18: Power levels for wall 1 relative to the terminal located in the room on the left at 1.5 m.

the image, the area in red indicates a higher received signal strength than the blue zone.

Figure 3.18 shows the estimate of a “mountain” for wall 1. The terminal was at a height of 1.5 m and is represented by a green cube. It is important to point out that for all the tests conducted, the terminal was always placed in the same position and the measurement was read using an app (TeamViewer), with screen sharing on a PC desktop. This solution was introduced because the human body near the mobile terminal could make oscillations and variations in the received

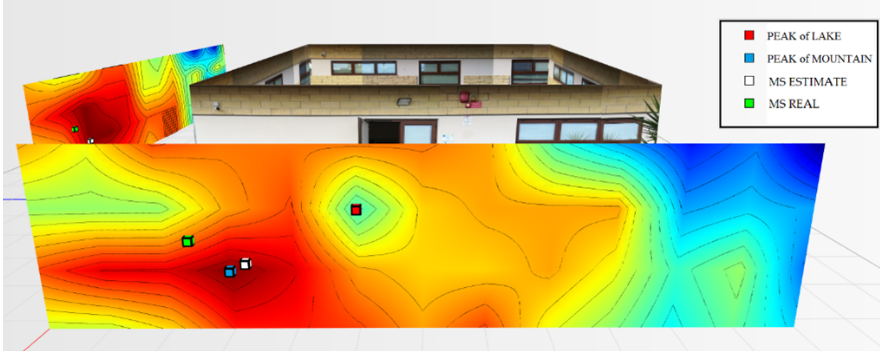


Figure 3.19: Power levels for wall 2 relative to the terminal located in the room on the left at 1.5 m.

power. The application of the terminal estimation algorithm, in this case, exploits the proximity method to estimate the terminal in the peak of power characterized by the "mountain". The $P_{Estimated}$ was estimated to be 0.74 m away from the actual position.

As for wall 2 (Figure 3.19), only one "lake" was found and a maximum peak of the "mountains". In this case, the algorithm applied the weighted distance method between the "lake" and the "mountain". The $P_{Estimated}$ was estimated to be 0.94 m away from the actual position.

Excellent localization accuracy could be observed in both cases. Taking into consideration the second wall, if an operator is present, who evaluates the power wall derived from the values, the point where the terminal is likely to be is the peak power point identified as the "mountain". To improve the localization estimate, therefore, it is possible to activate filters that allow, based on the assigned percentage, to take into account, or not, additional "mountains" or "lakes". In particular, the percentage is set at 50% by default (i.e., it must exceed 50% of height/depth with respect to the maximum/minimum peak).

If in this case, we apply a filter for the "lakes" of 70%, only a peak maximum power will be found. Therefore, the "proximity method" will be applied and the terminal will be estimated with an error of 0.78 m, with respect to the real position of the terminal. This can be observed in Figure 3.20.

Figure 3.21 shows the overview of the two power walls relating to the terminal located in the room to the left at 1.5 m.

The second test was carried out by putting the terminal inside the room to the right (wall perspective 2) of the technological pole building, at a height of 1.4 m. Figure 3.22 graphically represents the power levels mapped on the two walls. Additionally, in this case, the new algorithm was applied. These results show that for wall 1, the algorithm used the proximity method and estimated the position

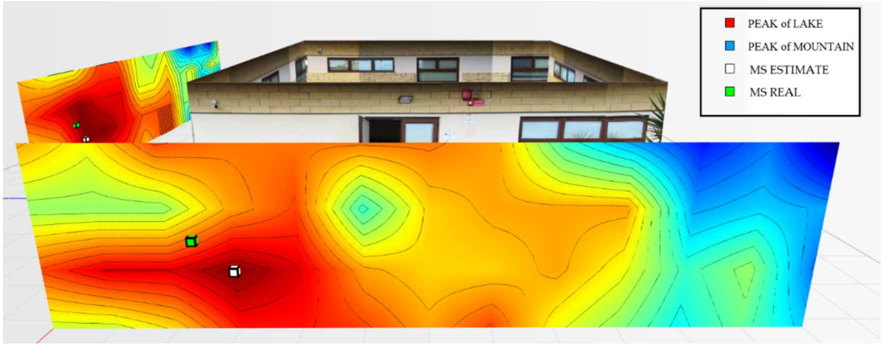


Figure 3.20: Power levels for wall 2 relative to the terminal located in the room on the left at 1.5 m with the filter.

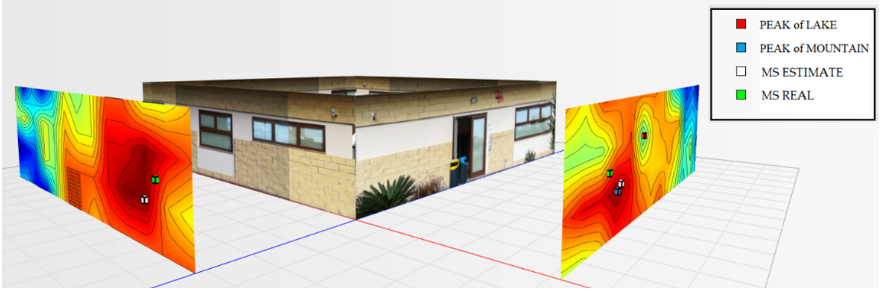


Figure 3.21: Overview of the power levels for walls 1 and 2, room terminal on the left at 1.5 m.

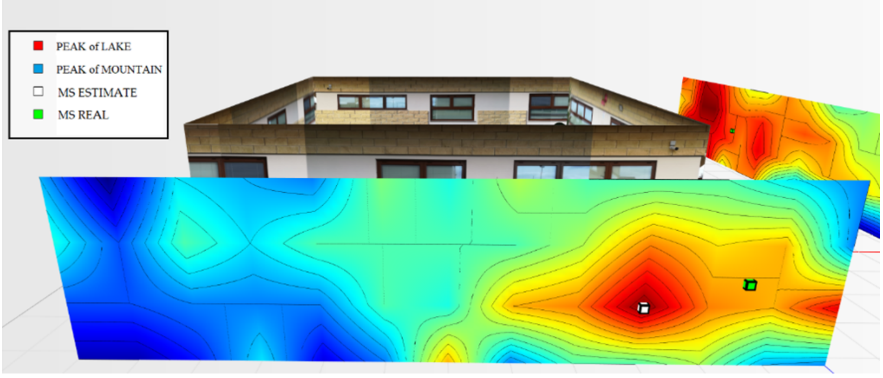


Figure 3.22: Power level for wall 1 related to the terminal located in the room to the right at 1.4 m.

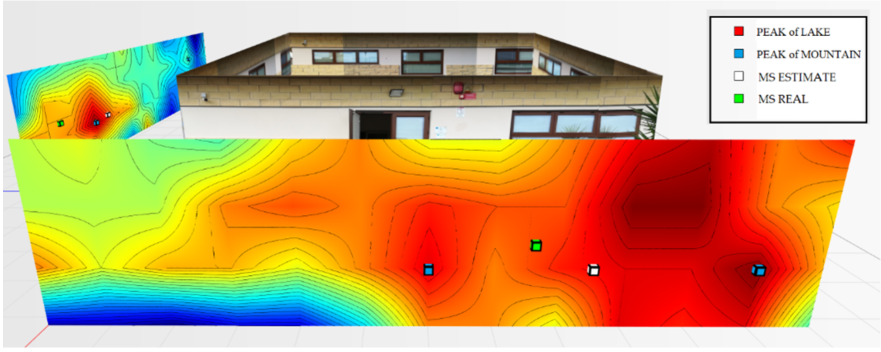


Figure 3.23: Power level for wall 2 related to the terminal located in the room to the right at 1.4 m.

of the device at the peak of the "mountain" at 1.6 m from the real position. Thus, a "lake" was found.

As for wall 2 (Figure 3.23), two "mountains" and no "lakes" were detected. The algorithm used the distance criterion, estimating the terminal at 0.98 m away from the actual position.

In general, even the fourth test calls for excellent precision concerning the estimate of the terminal position. Figure 3.24 shows an overview of the two power

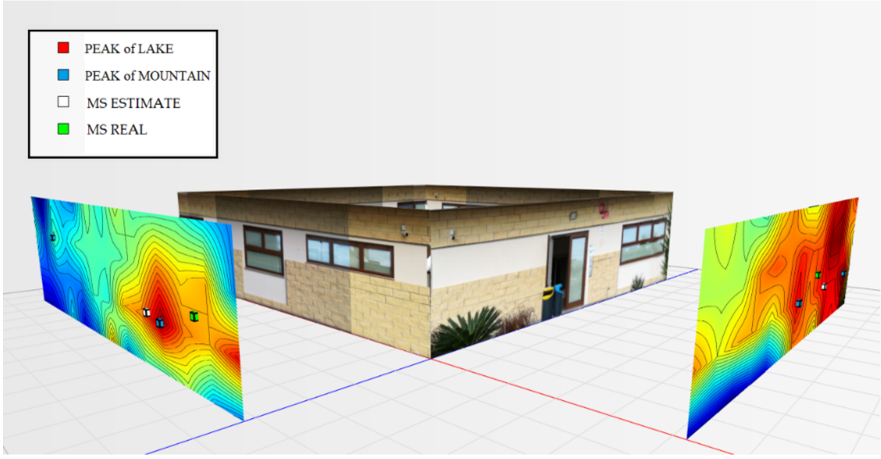


Figure 3.24: Overview of power levels for walls 1 and 2, room terminal to the right at 1.4 m.

walls relating to the terminal located in the room to the right at 1.4 m.

The proposed geolocation algorithm identified a mobile terminal through a 2D analysis of a power wall, but of course, it is possible to combine and aggregate the data to obtain a 3D geolocation that under certain circumstances can be decisive for the geolocation of a missing person.

3.6 Classification Algorithm considering 3D Monitoring Area in No Free Space - No Isotropic Attenuation

As studied in [94], the first step in identifying missing persons under the rubble is to classify the terminals within the selected monitoring area.

It is therefore assumed that the area in which the collapse occurred was incorporated into a parallelepiped of A_X, A_Y, H_R (length, width and height). Figure 3.12 shows the supposed monitoring area. The hypotheses underlying the classification algorithm are the following:

1. The size of the monitoring area is A_X, A_Y, H_R, meters, the monitoring area is formed by $S = \frac{A_X \cdot A_Y \cdot H_R}{27}$ under areas of size $3 \times 3 \times 3\ m^3$;

2. The flight of the drone takes place at height H_R, equal to the height of the rubble;

3. The drone takes different positions, which we will indicate with the term "$D_k^{Position}$". The $k-th$ positions are defined at the center of each sub-area

S at height H_r with $k = 1, ..., (\frac{A_X}{3} \cdot \frac{A_Y}{3})$;

4. The mobile terminals, defined as MS_i for $i = 1, ..., S$, are positioned at the points $C_i = (C_i^X, C_i^Y, C_i^Z)$, or in the centroid of the $i - th$ sub area S_i;

5. For each $k - th$ position of the drone, power received by the $i - th$ MS is given by (3.21):

$$P_T = P_R + G_T + G_R - A_0(d_{i-k}) - D_r \cdot d_{i-k} \cdot A_r - r_{i-k}, \tag{3.21}$$

   where d_{i-k} [m] is the distance between the position of each terminal and the $k - th$ position of the femtocell; A_0 is the attenuation in dB in free space (3.1); D_r is the density of the material in the interval $[0, 1]$ (expressed in percentage in the graphs); A_r is the material attenuation coefficient [dB/m]; r_{i-k} is a random value in dB to simulate an additional non-isotropic attenuation between the the terminal position and drone position; given by formula in (3.22):

$$r_{i-k} = rand[0, (d_{i-k} \cdot A_r \cdot W \cdot (0.1 + 0.9 \cdot D_r)], \tag{3.22}$$

   where W is a default weight set to 0.5 (50%).

6. The terminals that will be hooked by the femtocell will be those whose power value will be above a certain threshold, hereinafter referred to as $P_{th} = -120[dBm]$. The connected terminals will therefore be all terminals whose power is greater than P_{th}: if $P_{i-k} \geq P_{th}$ then $P_{i-k} \in P_{hoocked}$.

Once the working hypotheses have been defined and the terminals are generated, it is possible to introduce the classification algorithm. We define the set of terminals hooked P_{hooked}, without repetition and for each position taken by the drone within the monitoring area, such as $\{A\}$.

Some terminals are hooked by the femtocell but are not internal to the monitoring area, so these terminals represent the "false positive" value. So to reduce false positives it was thought to turn the femtocell around the perimeter of the monitoring area with a distance from the perimeter equal to its coverage radius. This further enables hooking all the terminals present outside the monitoring area, creating the set $\{E\}$, that is the set of terminals outside the monitoring area.

In the set $\{E\}$ there will be terminals that are also present in the set $\{A\}$.

Thus the intersection of the two sets creates the set $\{X\}$. This set is therefore formed by (3.23):

$$X = A \cap E \tag{3.23}$$

The set $\{X\}$ must be subtracted from the set $\{A\}$ to have the set of terminals considered as internal to the monitoring area. Therefore, the set $\{I\}$ is given by (3.24):

$$I = A - X \tag{3.24}$$

Once the terminals inside the monitoring area have been defined, it is possible to proceed to the localization phase.

To better understand the classification phase and related performance it is possible to evaluate the performance's accuracy to vary some parameters. The hypotheses made for the performance simulation are:

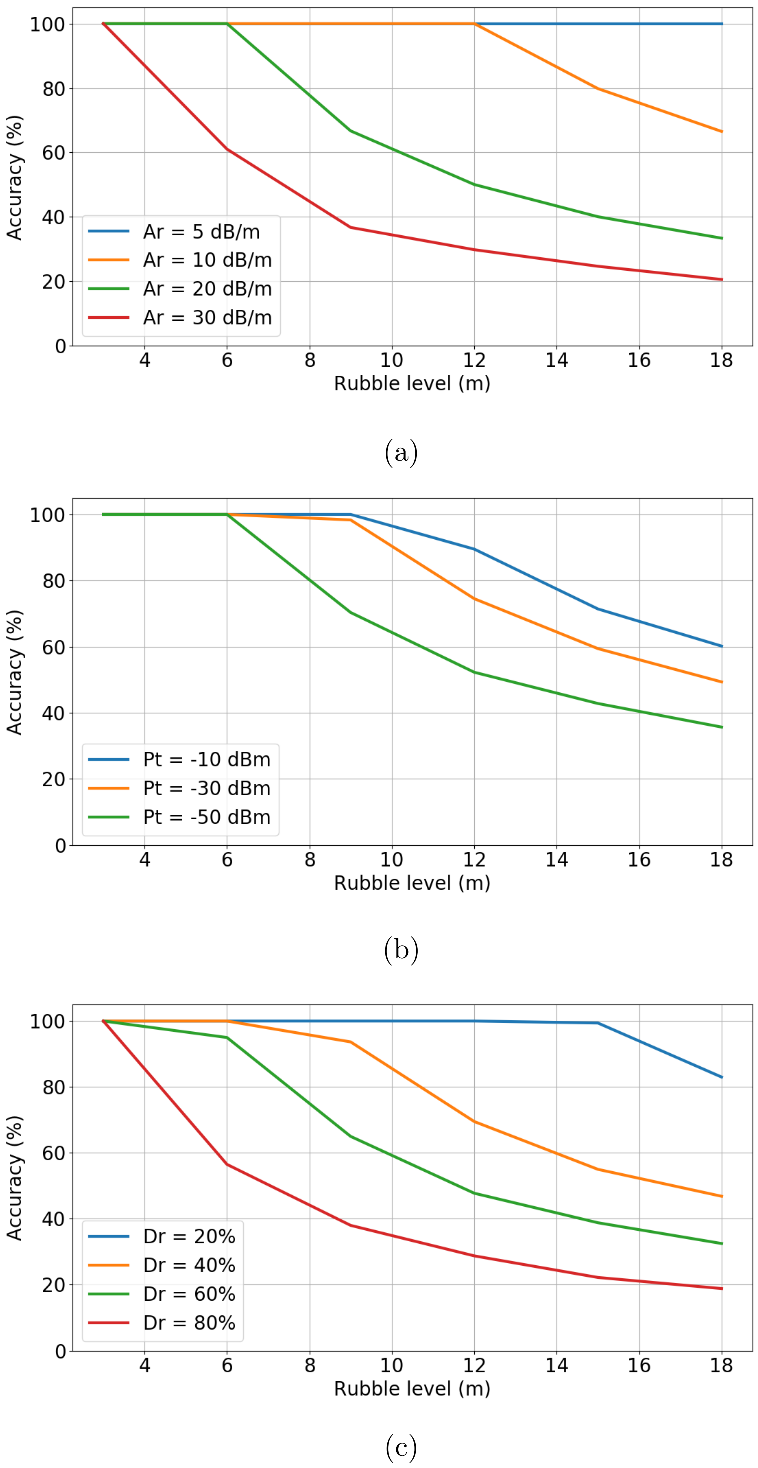
- Area Size: 30×30 m^2;
- Weight (W) of 5%;
- Transmission power P_T of -57 dBm;
- Power threshold P_{th} of -120 dBm;
- Sub area resolution of 3 m^3.

The Figures 3.25a, 3.25b and 3.25c represent the accuracy of the classification as the level of the rubble increases and change respectively A_r, P_T e D_r.

In this graph the density of the material D_r is assumed to be at a value equal to 50% and the curves are shown relatively to the different levels of additional attenuation of the material A_r. It may be noticed that as the quantity of rubble increases the accuracy visibly decreases, especially, in cases where there is an attenuation due to the material greater than 5 dB/m.

In this graph, obtained by assuming $A_r = 20dB/m$ and the density of the material at 50%,represents the curves relating to the different transmission power levels P_t. Also in this case it is clear that accuracy decreases with the increase of rubble, and, in particular, that the curve defined by a transmission power level of -30 dBm decreases more than that relative to the transmission power level equal to - 10 dBm.

In this graph, obtained by considering $A_r = 20dB/m$, the curves are represented as the density of the material D_r changes. Also in this last case the accuracy decreases as the rubble increases, and, particularly, where the density of the



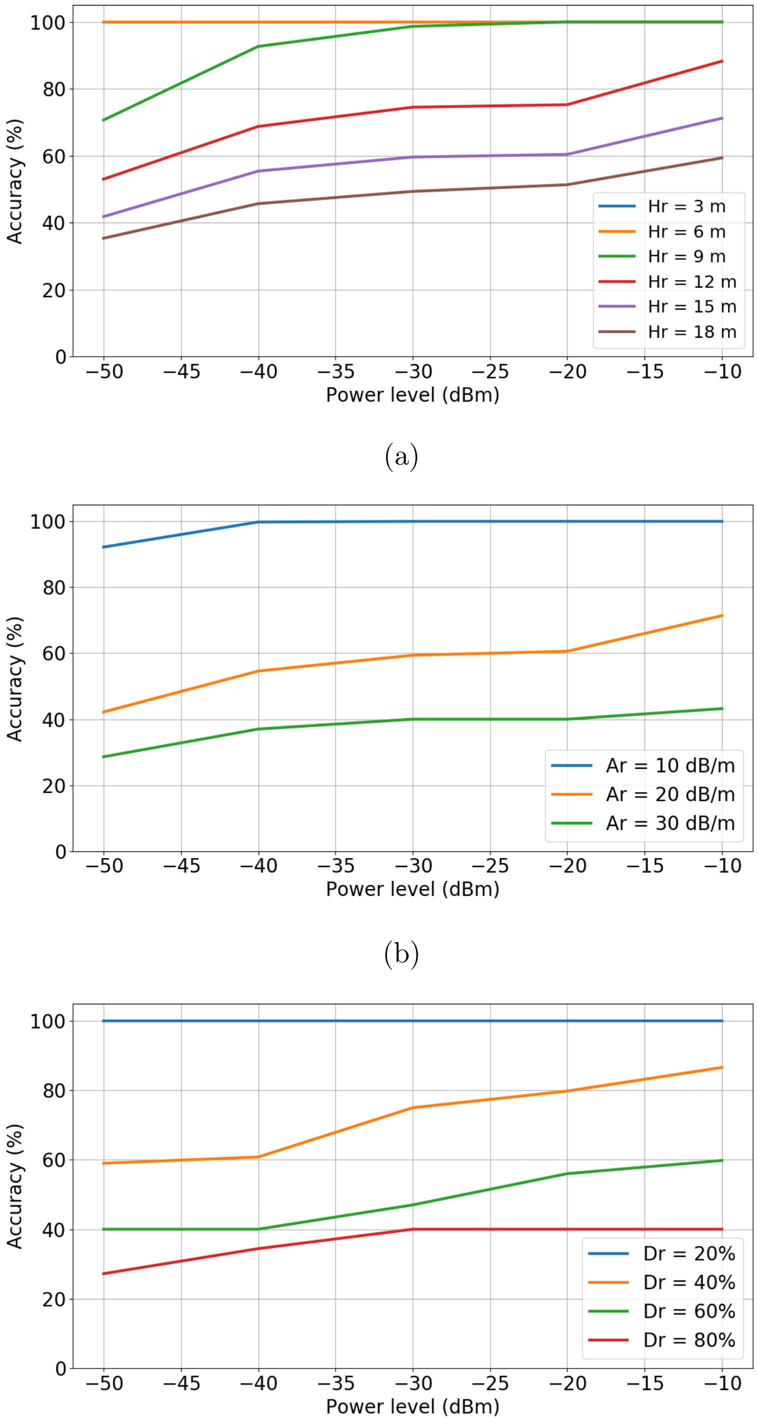
(a)

(b)

(c)

Figure 3.25: Accuracy versus rubble level: (a) Varying A_r and $D_r = 50\%$, (b) Varying P_t and $D_r = 50\%$, (c) Varying D_r and $A_r = 20dB/m$.

material is higher the accuracy will be very low, compared to the case in which the material has lower density. This is because with the same transmission power, with more material, there is more additional attenuation and, therefore, the number



(a)

(b)

(c)

Figure 3.26: Average classification accuracy varying several parameter: (a) Accuracy versus power level ($A_r = 20dB/m, D_r = 50\%$), (b) Accuracy versus power level ($A_r = 20dB/m, D_r = 50\%$), (c) Accuracy versus power level ($H_R = 20dB/m, A_r = 20dB/m$).

of terminals hooked up will be lower.

The Figure 3.26a, 3.26b and 3.26c represent the accuracy of the classification as

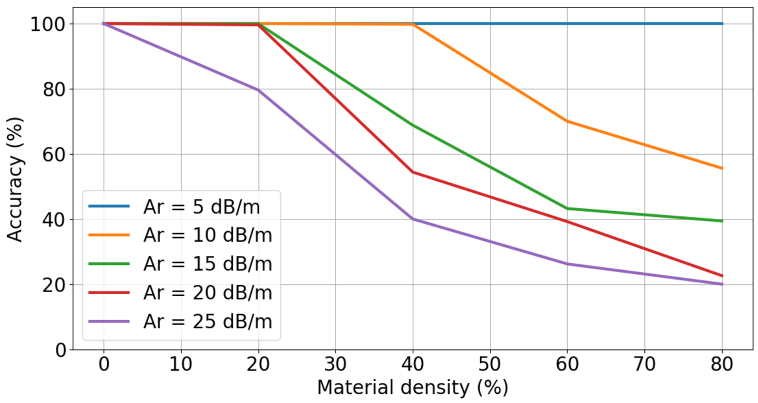


Figure 3.27: Accuracy versus material density ($H_R = 15m$).

the level of the transmission signal's strength increases P_T and varies according to H_R, A_r e D_r.

Graph in Figure 3.26a is obtained by assuming a material density of 50% and a $A_r = 20dB/m$, the curves are shown relative to the different heights of the rubble. It may be noticed how as the signal power level increases in transmission, the level accuracy grows visibly. In particular, for a drone height of 3 and 6 meters accuracy is 100%. This means that the femtocell manages to hook all the terminals present in the monitoring area and classify them as “IN”.

Graph in Figure 3.26b is obtained by assuming the density of the material to be at 50% and the height of the drone A_z at 15 m, the curves for the different levels of additional attenuation are represented A_r. Also in this case, it is clear that the accuracy increases with the increase in power, and, in particular, that the curve defined by an attenuation level of 10 dB/m leads to a much higher accuracy than that relative to $A_r = 30dB/m$.

Finally, graph in Figure 3.26c is obtained with the hypothesis of $A_r = 20dB/m$ and H_R at 15 m, the curves are represented as the density of the material changes D_r. Also in this last case the accuracy increases with the increase of the power level. In particular, where the density of the material is higher, the accuracy will be very low, compared to the case in which the density of the material is lower, thus, allowing more terminals to be hooked.

Figure 3.27 below represents the accuracy of the classification as the density of the material increases D_r and to vary the A_r.

From the graph it can be seen that by presuming the drone's flight height to be at 15 m, accuracy decreases as the density of the material increases and, also, the additional attenuation increases.

3.7 Localization Algorithm Supposed 3D Monitoring Area in No Free Space - No Isotropic Attenuation

As studied in [94], the localization phase can be implemented using two different algorithms:

1. Proximity algorithm (sub-section 3.7.1);
2. Cluster-based fast proximity algorithm (sub-section 3.7.2).

3.7.1 Proximity Algorithm

The localization algorithm is based on the hypotheses made previously, in the classification phase. The localization phase involves estimating the position of the MS_i terminal in the $MS_i^{Estimated}$ point, on the Z plane (in 2D), where this terminal has greater power.

In fact, after defining the terminals classified as IN, having memorized, in the first phase, the P_{i-k} power measurements for each $D_k^{Position}$ it is possible to obtain a power grid in the surface of the monitoring area for each terminal, with which it is possible to estimate the position of the terminal at the grid point where maximum power is obtained for that terminal.

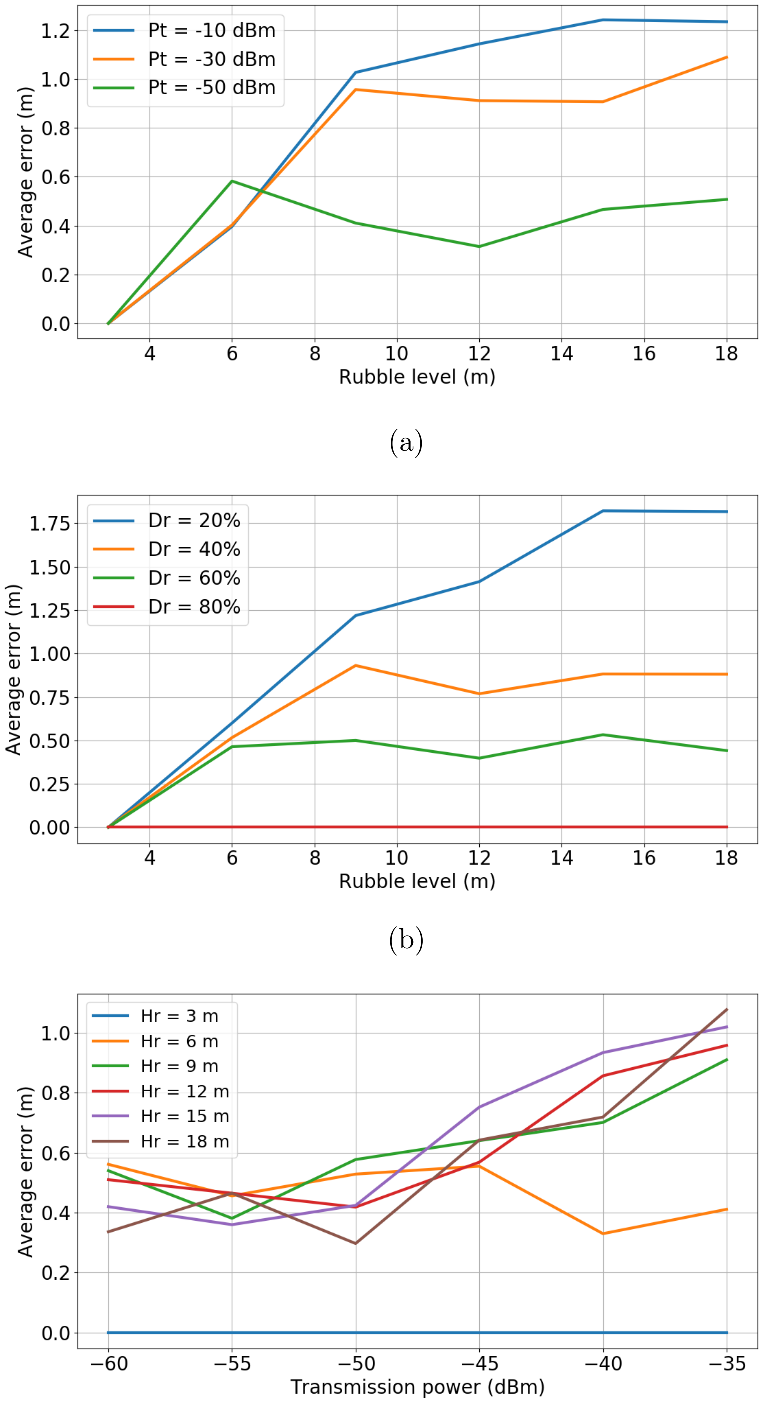
The localization error is expressed in meters and is estimated by calculating the distance between the two vertical axes passing through the positions of MS_i and $D_k^{Position}$ in which the drone measures the maximum power received by the mobile terminal MS_i, then assuming that the error along the Z axis is zero. The error is based on (3.25):

$$\in = d_{vert}\left[MS_i, D_k^{Position}\right] \tag{3.25}$$

where $d_{vert}\left(Point_i, Point_k\right)$ represents the distance between the vertical lines passing through $Point_i$ and $Point_k$.

To better understand the “Proximity Algorithm” and related performance it is possible to evaluate the average error of the estimated position.

In Figure 3.28a, 3.28b and 3.28c the average error is represented by varying one of the 3 parameters H_R, P_T and A_r. In Figure 3.28a the additional attenuation is assumed at 20 dB/m and the density of the material at 50% and the curves relating to different values of P_T are represented. In this case, the average



(a)

(b)

(c)

Figure 3.28: Average error varying several parameter: (a) Average error versus rubble level ($A_r = 20dB/m, D_r = 50\%$), (b) Average error versus rubble level and density level ($A_r = 20dB/m$), (c) Average error versus transmission power ($A_r = 20dB/m, D_r = 50\%$).

localization error increases with increasing rubble and, more importantly, with increasing transmission power.

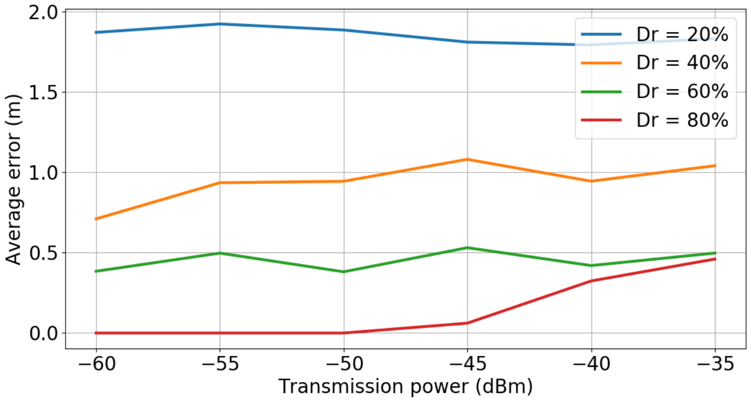


Figure 3.29: Average error versus transmission power and D_r ($A_r = 20dB/m, H_R = 15m$).

This occurs because as P_T increases, more terminals are hooked on which the localization algorithm must then be performed, so having more terminals, in the presence of attenuation and 50% material density, an error will occur at a higher location, but still between 0 and $1.3m$.

In Figure 3.28b, a material with an attenuation coefficient of 20 dB/m is assumed, and the curves relating to the different levels of material density are represented. Again, in this case, the average error increases as the rubble grows and the material density decreases. The average error decreases as the density of the material increases, because this increase leads to a situation of isotropic attenuation, and, therefore, reduces the localization error.

Figure 3.28c shows the average error as both the transmission power and the Hr increase, assuming an A_r attenuation of 20 dB/m and the density of the material at 50%. As can be seen from the figure, when we are at the height of the rubble at $3m$, by varying the transmission power, it is possible to make zero error. As the height of the rubble increases, for transmission powers between $[-60dBm, -50dBm]$ there is an error from 40 to 50 cm. For the remaining power values, the average error increases more when we are at different heights. The increase in error as transmission power increases is due to the fact that the femtocell hooks up more terminals, i.e. those deeper down.

Finally, in Figure 3.29, as in the previous one, the average error is evaluated as the transmission power increases. In particular, the level of the rubble is assumed to be at $15m$ and the additional attenuation at 20 dB/m and the curves relating to the different levels of material density are represented.

The average error, in this case, again, increases as the transmitted power increases and the density of the material decreases.

3.7.2 Cluster-based Fast Proximity Algorithm

In the real post-earthquake scenarios of a lost person localization times are very important. It is of fundamental importance to be fast in the rescue times to minimize the drone's energy consumption [95, 96] and that of the mobile devices.

The reduction in flight time and energy consumption has been studied by several researchers. In [97], the propulsion energy consumption model of the fixed-wing UAV was derived and an efficient trajectory maximizing the UAV's energy efficiency was designed.

In addition, energy efficient schemes have attracted wide attention due to the battery technology limitation of mobile devices and UAVs. The authors in [98], studied the minimization problem of the weighted sum energy consumption of the UAV and users. The computation resource scheduling, the bandwidth allocation and the trajectory of the UAV were optimized in the minimization problem.

In this paper the proposed method to reduce energy consumption is based on the optimization of the drone flight. It will be positioned on certain points according to the algorithm that will be hereinafter illustrated.

It is supposed that terminals are not distributed in a non-uniform way within the monitoring area. There will be an area with higher density of terminals unlike another area with lower density. For this reason it is possible to apply the "Cluster-based Fast Proximity Algorithm" to select one sub-area, rather than another, where larger quantities of terminals with greater power are detected. The algorithm includes the following steps:

1. Determination of the minimum resolution of the location error to be obtained (1 m, 2 m, 3 m, etc.);

2. Estimation of the monitoring area so that it can be divided into 4 quadrants for each phase;

3. Selection of 9 grid positions where the drone is stationed. 4 of these positions are represented by the vertices of the monitoring area and another 4 by the median positions of each side, finally the last position is represented by the centroid. From the second phase on wards, the points are reduced to 5 (the middle 4 of the sides of the subarea and the relative centroid).

4. Identification of the number of terminals for each sub-area. This is done by evaluating which terminals, at the different points defined above, have the highest power value;

5. Once the number of terminals identified in the various sub-areas has been defined, it is possible to intervene in the point with several terminals and therefore the whole procedure is repeated, starting from point 2, for the sub-area in question;
6. The phases are F and depend on the size of the monitoring area.

The algorithm uses the subdivision in phases for the optimization of the trajectory of the drone, which is defined as “2” or “serpentine”. In particular, hypotheses are made for simplicity:

- The grid must be an $M \times M$ matrix in which:

$$M = 2^n + 1, n = 2, 3, ..., N \tag{3.26}$$

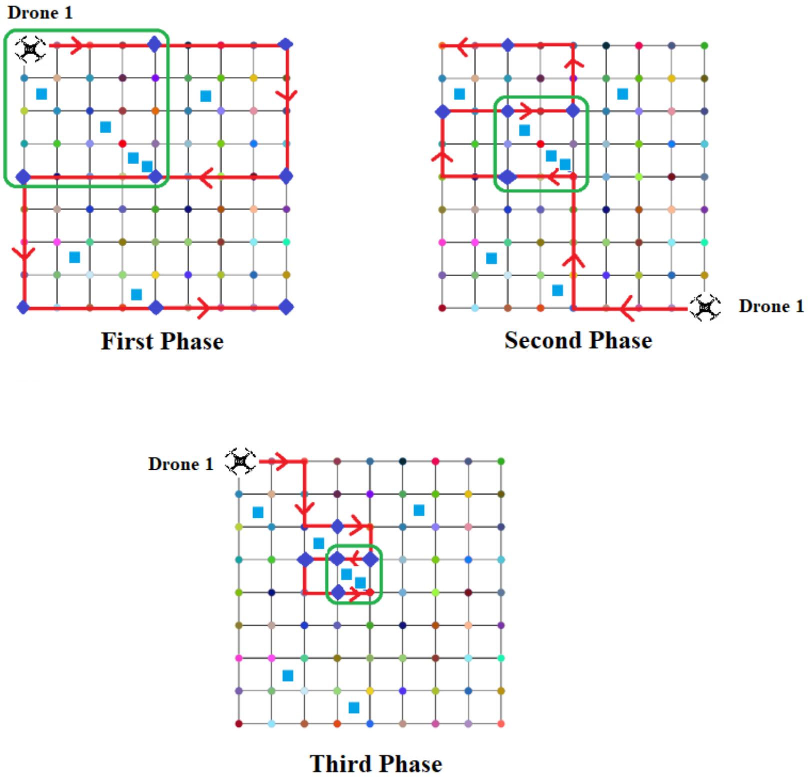
- The matrix must not be 2×2 or 3×3, since there would be no optimization of the trajectory and therefore no energy/time saving;
- The number of iteration phases of the algorithm must be given by (3.27):

$$F = M - 1 \tag{3.27}$$

From this formula it can be understood that $n = F$.

Once the initial hypotheses are defined, it is possible to provide a series of definitions:

- E_{tot}^{NO} is the total energy not optimized;
- E_{tot}^{O} is the total energy optimized;
- T_{P-tot}^{NO} is the total non-optimized point processing time;
- T_{P-tot}^{O} is the optimized total point processing time;
- T_{V-tot}^{NO} is the total flight time of the non-optimized drone;
- T_{V-tot}^{O} is the total flight time of the non-optimized drone;
- P it is the power used by the drone to stay in flight;
- t_p it is the processing time, i.e. the time required by the mobile terminal to detect the power received by the;



First Phase

Second Phase

Third Phase

Figure 3.30: Optimized drone rout.

- δ_t is the time it takes the drone to fly from one point on the grid to another;

It is now possible to define the non-optimized total energy as:

$$E_{tot}^{NO} = P \cdot \left(T_{P-tot}^{NO} + T_{V-tot}^{NO}\right) \tag{3.28}$$

where:

$$\begin{gathered} T_{P-tot}^{NO} = t_p \cdot M^2 \\ T_{V-tot}^{NO} = \delta_t \cdot \left(M^2 - 1\right) \end{gathered} \tag{3.29}$$

While the total optimized energy can be expressed in the following way:

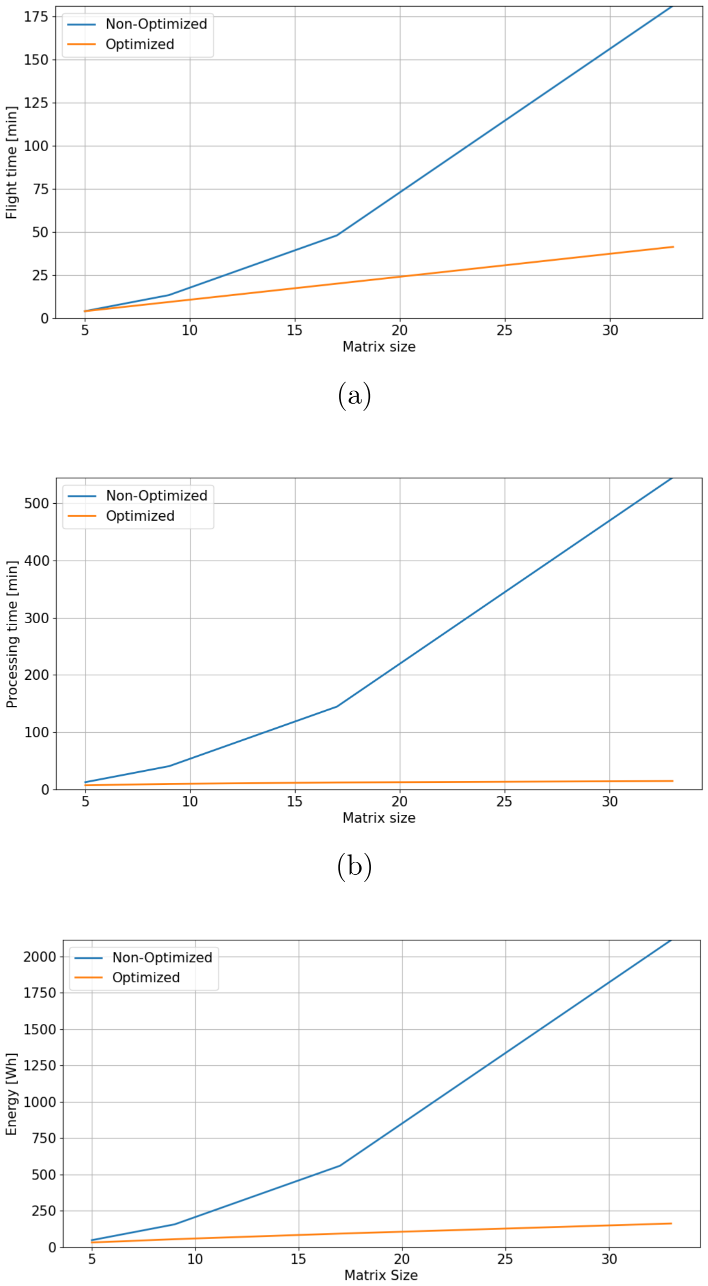
$$E_{tot}^{O} = P \cdot \left(T_{P-tot}^{O} + T_{V-tot}^{O}\right) \tag{3.30}$$

where:

$$\begin{gathered} T_{P-tot}^{O} = t_p \cdot (5 \cdot F + 4) \\ T_{V-tot}^{O} = \delta_t \cdot \left[2^F \cdot \left(4 + \sum_{F=2}^{F_{max}} \frac{2}{2^{F-2}}\right)\right] \end{gathered} \tag{3.31}$$

The flight and total processing time depend on the route iteration phases and on the size of the matrix (also linked with the number of phases). Moreover, to define the previous formulas, we adopted the worst scenario, in which the selected quadrant (in which the terminal is present) is on the opposite side to the last position taken by the drone. Figure 3.30 represents an example of the process of iteration and path optimization.

To better understand the “Cluster-based Fast Proximity Algorithm” and re-



(a)

(b)

(c)

Figure 3.31: Comparison between the curve defined by: (a) T^{O}_{P-tot} and T^{NO}_{P-tot}, (b) T^{O}_{V-tot} and T^{NO}_{V-tot}, (c) E^{O}_{tot} and E^{NO}_{tot}.

Table 3.2: Percentage energy optimization.

M	Reduction to [%]	Reduction [%]
5	70.70	29.29
9	38.70	61.30
17	19.05	80.95
33	5.53	94.47

lated performance it is possible to evaluate optimized/non-optimized flight time, processing time and compare each other.

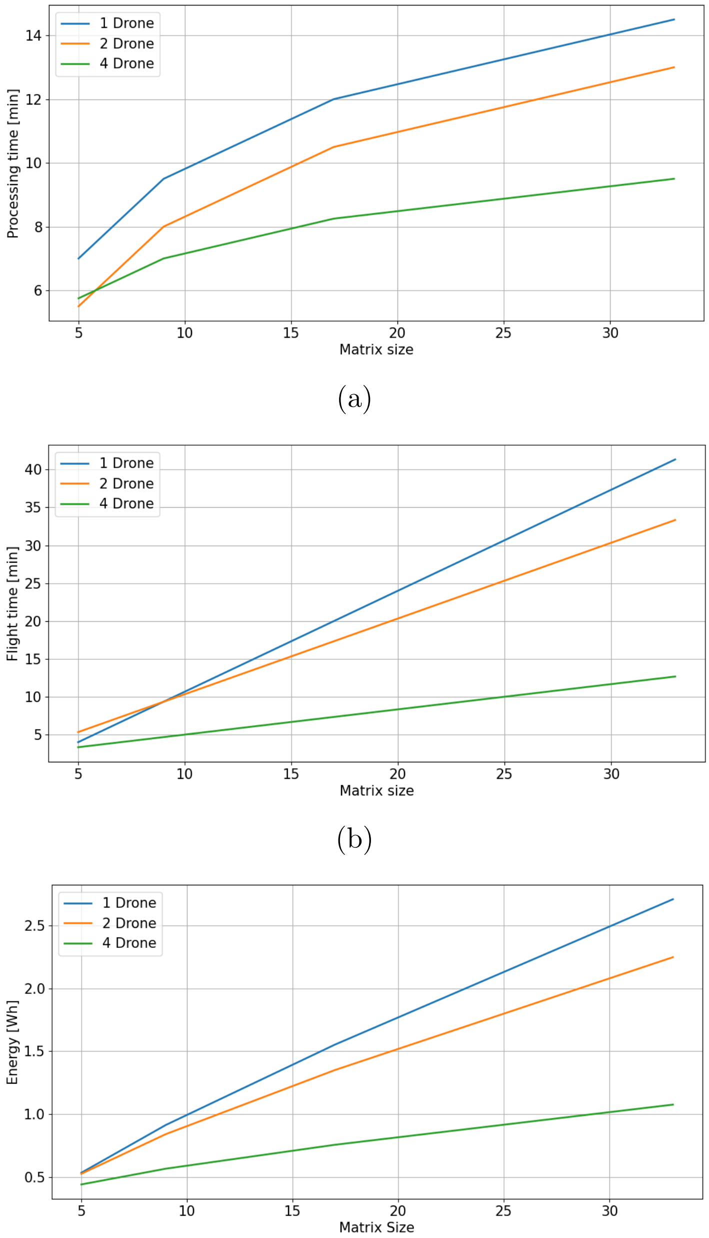
Placing ourselves in the conditions described in the previous section, in which this algorithm is introduced and illustrated, we show graphs that relate the optimized and non-optimized flight time and to processing time, and therefore also the optimized and non-optimized energy expenditure, to vary of the size of the monitoring area.

Figure 3.31a, 3.31b and 3.31c show that this trajectory optimization algorithm provides considerable energy saving. Table 3.2 shows, as the size of the matrix varies, the “reduction to” and the “reduction in”, in percentage, of energy.

This method could be extended considering the use of two or more cooperating drones, optimizing flight time and areas to be covered. This mechanism leads to a reduction in the energy consumed by drones and also in rescue times. Therefore, by using multiple drone-femtocell systems, the need arises to remodel the algorithm for optimizing the drone flight time, in order to intelligently cover each sub-area of the monitoring area. To this end, once the optimization algorithm is applied, the graphs relating to processing times, flight times and energy expenditure are obtained as the number of drones used and the size of the matrix that defines the monitoring area vary. To apply the algorithm, the following constraints were introduced:

- Coverage radius of the femtocell on board the drone equal to half the diagonal of the starting grid;
- The terminals hook onto the first femtocell they detect;
- The drones depart from the edges of the grid with a time lag of one minute, to prevent them from passing through the same point at the same time;
- Uneven distribution of terminals.

Processing time, flight time and energy, in the case of two and four drones are given by (3.32) and (3.33), respectively:



(a)

(b)

(c)

Figure 3.32: Comparison between 1, 2, 4 drones when matrix size varying: (a) Processing time, (b) Flight time, (c) Energy.

$$\begin{gathered}
T^{O}_{P-tot} = t_p \cdot (5 \cdot F_{max} + 1) \\
T^{O}_{V-tot} = \delta_t \cdot \left[20 + 2^{F_{max}} \cdot \left(\sum_{F=2}^{F_{max}} \frac{3}{2^{F-2}} \right) \right] \\
E^{O}_{tot} = P \cdot \left(T^{O}_{P-tot} + T^{O}_{V-tot} \right)
\end{gathered} \tag{3.32}$$

$$
\begin{aligned}
T^{O}_{P-tot} &= t_p \cdot \left[5 \cdot \left(\frac{F_{max}+1}{2} \right) + 4 \right] \\
T^{O}_{V-tot} &= \delta_t \cdot \left[2^{F_{max}} \cdot \left(2 + \frac{3}{2^{F_{max}-2}} \right) \right] \\
E^{O}_{tot} &= P \cdot \left(T^{O}_{P-tot} + T^{O}_{V-tot} \right)
\end{aligned}
\tag{3.33}
$$

Using (3.31)(3.30), (3.32) and (3.33) it is possible compare the trend of the curves relating to processing time, flight and energy consumption, based on the use of 1, 2 or 4 drones.

Figures 3.32a, 3.32b and 3.32c represent, respectively, the trend of the curves. These graphs confirm what has been said for a 9 × 9 matrix. In fact, as regards the processing time, the greatest reduction is obtained by passing from 1 to 4 drones. In terms of flight time and energy, there is a sharper decrease from 2 to 4 drones. Once the number of drones has been fixed, processing times, flight times and energy increase hand in hand with the increase in the size of the matrix, but with a different trend. The processing time increases linearly, while the flight time and energy grow according to an exponential trend.

3.8 Localization Algorithm Used Game Theory Algorithm

The use of UAV-Femtocell systems for the geolocation of mobile terminals has recently been introduced in the literature as an alternative and effective method in civil protection scenarios and applications. In this context, one of the important aspects concerns the optimization of geolocation times, bearing in mind the limited duration of the flight time of drones. This sub-section proposes a game theory-based approach to geolocation using UAV-Femtocell systems.

Over the past few years, GT has played a considerable role in different application contexts within the telecommunication system [99, 100].

In [99], two main aspects of GT are used to control the power of mobile terminals.

In [100], the authors seek the optimal placement of two UAVs providing backbone communications to a community of mobile terminals by applying the GT technique. Drones are seen as the "players" and the set of possible positions they can take are the "strategies".

Also in [100], the authors suggest the use of GT methods where the drones (i.e. the "players") exhibit non-cooperative behavior in order to optimize the position of the UAVs.

As studies in [101, 102] in this sub-section we want to apply the concepts of GT in an application scenario different from those described above. In fact, the application scenario corresponds to an earthquake zone with missing people who must be located under the rubble via their mobile terminals by applying UAV-Femtocell systems, or a system equivalent to a BS of a mobile radio network that is not fixed, but can move around the territory by approaching the mobile terminals which, on the other hand, in this particular scenario are in fixed positions.

In this sub-section it's propose a localization technique based on the use of two UAV-Femtocell systems and GT techniques in order to define the optimal position in which the drones must move in order to save flight time, reduce energy and locate the largest number of mobile terminals that connect. The results of the optimized flight time and energy is compared with the "Serpentine" method studied in [94], and described in sub-section 3.7.2.

In this study is introducing a number of improvements to the GT algorithms and extending the performance and comparative analysis to a scenario closer to the real one. In particular, the main features introduced in this work are:

- N number of mobile terminals distributed over the monitoring area;
- Device position generation within the monitoring area occurs randomly;
- Different dimensions of the monitoring area have been evaluated in order to provide a more complete analysis of the performance in the simulation phase. The number of mobile devices within the area, in this case, is proportional to the size of the monitoring area;
- During the game, in each "round", instead of performing a single iteration, *i=10000* iterations are performed in order to provide greater statistical significance of the results and to verify the robustness ensuring greater guarantee of achieving the Nash equilibrium;
- Evaluation of the work through a series of simulations aimed at assessing time and energy gain according to the variation in different parameters, i.e. size of the monitoring area, number of mobile devices generated, different data processing and drone flight speeds;
- Three Utility Functions (UFs) have been introduced to allow obtaining the Nash equilibrium at all times. A performance comparison of the three UFs has been made;

- The study introduces a mechanism for unlocking the game if Nash equilibrium is not achieved;
- A series of simulations have been introduced:
  - Comparison of UFs with respect to time (flight + processing) and energy gain according to the number of mobile terminals N generated and the size of the monitoring area;
  - Calculation of the percentage of times Nash equilibrium does not occur for each of the three UFs as the number of devices and the size of the monitoring area vary;
  - Evaluation of the entire game simulation times as UFs vary;
  - Having determined the monitoring area, the best utility function and the number of devices, a performance comparison at different data processing and drone flight speeds has been made.
- Finally, in order to account for the different effects on the power attenuation of rubble related to the collapse of a structure in a realistic post-earthquake scenario, tests were carried out assuming a non-isotropic propagation of electromagnetic waves within the monitoring area according to the formula presented in [94]. This formula introduces the existing Friis attenuation on a random additive component considering the type and density of the material present in the rubble in post-earthquake disaster scenarios.

3.8.1 Game Strategy

In this sub-section is illustrated the geolocation method and UAV energy efficiency method using GT; the main concept of the GT is illustrated in Appendix B. Figure 3.33 shows an example of a rectangular-shaped monitoring area over which $D = 2$ drones intervene. In this study, as in [101], the "players" are represented by drones, while the movements along the grid points of the monitoring area represent the possible "strategies".

The initial position of the drones is defined from the two centroids that characterize two equal partitions of the monitoring area. Conversely, the mobile M devices are randomly distributed. Unlike the "Serpentine" method in which drones move from one lattice point to the adjacent one, in the present study the drones start moving from the center of a macro-cell consisting of four basic cells towards its four vertices, which represent possible strategies. At the beginning of each

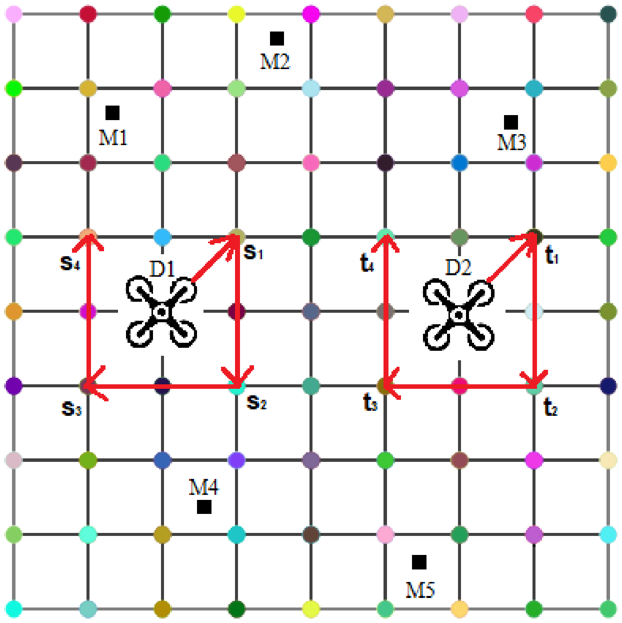


Figure 3.33: Example of simulated scenario.

stage of the game, the drones, based on their position, identify the mobile terminals with the highest power and proceed with several game rounds until the terminals are located. At the end of every round the winning strategy is determined (i.e. the vertex on which the drone will have to move) calculating the UFs and, therefore, the Nash equilibrium. By defining the first drone as D_1 and the second drone as D_2, we can define the set of possible strategies of the two drones, according to (3.34). The UFs of the two drones will be defined in the Cartesian product $A_{D_1} \times A_{D_2}$, so each of them can take 16 values.

$$\begin{aligned} A_{D_1} &= \{s_1, s_2, s_3, s_4\}, \\ A_{D_2} &= \{t_1, t_2, t_3, t_4\}. \end{aligned} \tag{3.34}$$

Next, we will generalize the proposed method to a generic number of devices, i.e. mobile terminals, within the monitoring area and the newly introduced UFs: let M_m be the m-th mobile terminal present in the monitoring area, with m = *1, ... M*, where M represents the total number of mobile terminals to be geolocated. In (3.35), $M_{\overline{m1}}$ and $M_{\overline{m2}}$ are defined as the terminals for which the drones measure the maximum power at the s and t points, respectively:

$$\begin{aligned} M_{\overline{m1}} &= m : P_{D_1}^{M_m}(s) = Max, \\ M_{\overline{m2}} &= m : P_{D_2}^{M_m}(t) = Max, \end{aligned} \tag{3.35}$$

where the powers $P_{D_1}^{M_m}$ and $P_{D_2}^{M_m}$ are those of the terminal m-th measured by the drone D_1 and D_2, respectively. The proposed UFs have been structured to be maximized in situations where the drone is approaching the reference terminal to be geolocated. For convenience we will denote “UF_1” (Utility Functions 1) the ‘UFs” that have been formulated to be maximized when the power measured by the relative drone increases (indicating that the drone is approaching the terminal to be geolocated) and, at the same time, when the power difference from that measured by the second drone increases (with the aim of minimizing the overall movements of the drones and, therefore, reducing geolocation times and decreasing the overall duration of the flight); “UF_2” (Utility Functions 2) the utility functions related to the power ratio of the mobile terminals measured by the two UAV-Femtocell systems; “UF_3” (Utility Functions 3) the utility functions related to the power differences of the mobile terminals measured by the two UAV-Femtocell systems. The three types of UFs are expressed respectively as (3.36) and (3.37), (3.38) and (3.39), (3.40) and (3.41).

- UF_1:

$$u_1(s_i, t_k) = P_{D_1}^{M_{\overline{m_1}}}(s_i) * [P_{D_1}^{M_{\overline{m_1}}}(s_i) - P_{D_2}^{M_{\overline{m_1}}}(t_k)], \tag{3.36}$$

$$u_2(s_i, t_k) = P_{D_2}^{M_{\overline{m_2}}}(t_k) * [P_{D_2}^{M_{\overline{m_2}}}(t_k) - P_{D_1}^{M_{\overline{m_2}}}(s_i)]; \tag{3.37}$$

- UF_2:

$$u_1(s_i, t_k) = \frac{P_{D_1}^{M_{\overline{m_1}}}(s_i)}{P_{D_2}^{M_{\overline{m_1}}}(t_k)}, \tag{3.38}$$

$$u_2(s_i, t_k) = \frac{P_{D_2}^{M_{\overline{m_2}}}(t_k)}{P_{D_1}^{M_{\overline{m_2}}}(s_i)}; \tag{3.39}$$

- UF_3:

$$u_1(s_i, t_k) = P_{D_1}^{M_{\overline{m_1}}}(s_i) - P_{D_2}^{M_{\overline{m_1}}}(t_k), \tag{3.40}$$

$$\begin{aligned} u_2(s_i, t_k) = P_{D_2}^{M_{\overline{m_2}}}(t_k) - P_{D_1}^{M_{\overline{m_2}}}(s_i), \\ i, k = 1, .., 4. \end{aligned} \tag{3.41}$$

A stage of the game ends when the drone measures less power than in the previous round, according to the criterion described in sub-section 3.8.2. Once the two terminals are located, the game resumes. The drone that geolocalizes a terminal first, waits for the completion of the localization phase of the other drone, in order to synchronize the beginning of the next stage. If only one mobile device remains, the drone that detects the greatest power will proceed to its localization.

The proposed geolocalization technique does not foresee that two drones can compete for the same terminal as only in the initial set-up phase of the localization procedure a single drone is associated with the mobile terminal with greater power. In this phase, shall a drone see more terminals with the same power, a random association is carried out. On the other hand, if the two drones see the same terminal $M_{\overline{m_k}}$ with the same power, the maximum powers of the two successive mobile terminals $P_{D_1}^{M_{\overline{m_x}}}(s)$ are compared and $P_{D_2}^{M_{\overline{m_y}}}(s)$ measured by the two drones. At this point, the terminal $M_{\overline{m_k}}$ is associated with the drone D_1 if it measures a power on the second minor terminal, that is, if $P_{D_1}^{M_{\overline{m_x}}}(s) < P_{D_2}^{M_{\overline{m_y}}}(s)$. In the final phase, the management of the geolocalization of the last mobile terminal to be geolocated is performed by the same drone it is associated with, the second drone eventually moving to a different monitoring area.

3.8.2 Game Theory-based Geolocation

The present section will describe the localization phase of mobile terminals. In particular, an example of the geolocalization of two terminals in a certain number of rounds is illustrated in Figure 3.34. Not knowing a priori the positions of the mobile terminals, the starting positions of the UAV-Femtocell systems coincide with the centroids of a partition with $D = 2$ zones of rectangular shape.

R_{D_1} and R_{D_2} denote the rounds in which the drones D_1 and D_2 locate the terminals M_1 e M_2, respectively. $r_{D_1} - th$ and $r_{D_2} - th$ stand for the rounds that the drones must perform in order to locate the mobile devices. Each of the drones D_1 and D_2 performs three movements with the three respective new power measures $P_1(r_{D_1})$, $P_2(r_{D_1})$, $P_3(r_{D_1})$ and $P_1(r_{D_2})$, $P_2(r_{D_2})$, $P_3(r_{D_2})$. So, the geolocation conditions in round r_{D_1} of terminal M_1 carried out by D_1 and round r_{D_2} of terminal M_2 carried out by D_2 are as follows:

1. if the UAV-Femtocell system has not dropped the terminal with respect to the previous round;
2. if

$$\begin{aligned} D_1 &: \begin{cases} P_1(r_{D_1}) < P_{MAX}(r_{D_1} - 1) \\ P_2(r_{D_1}) < P_{MAX}(r_{D_1} - 1) \\ P_3(r_{D_1}) < P_{MAX}(r_{D_1} - 1) \end{cases} \\ D_2 &: \begin{cases} P_1(r_{D_2}) < P_{MAX}(r_{D_2} - 1) \\ P_2(r_{D_2}) < P_{MAX}(r_{D_2} - 1) \\ P_3(r_{D_2}) < P_{MAX}(r_{D_2} - 1) \end{cases} \end{aligned} \tag{3.42}$$

where $P_{MAX}(r_{D_1} - 1)$ and $P_{MAX}(r_{D_2} - 1)$ represents the maximum value of the power measured in the previous round by D_1 and D_2, respectively. The round

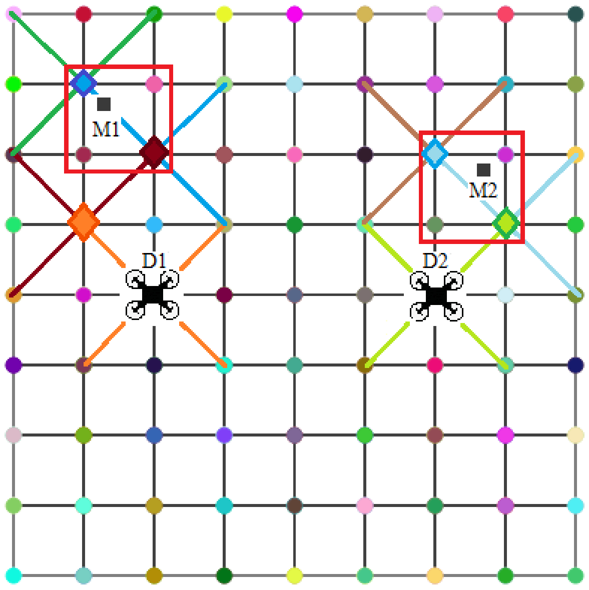


Figure 3.34: Geolocation of two terminals in a certain number of rounds.

$r_{D_1} = R_{D_1}$ for which the previous conditions occur is the one in which the geolocation of M_1 occurs, the same condition is true for the localization of the terminal M_2 performed by D_2.

In the proposed example, taking into account the previously established conditions, terminal M_1 is geolocated in $R_{D_1} = 4$ represents by the square (highlighted in the figure by a red box) characterized by the two opposite vertices with the last two maximum power values. Similarly M_2, as shown in Figure 3.34, is geolocated in $R_{D_1} = 3$ rounds. The definite conclusion of the procedure is defined by the round in which the last mobile terminal is connected. The end of the geolocalization procedure is therefore determined by the location of all mobile terminals.

3.8.3 Unlocking Mechanism in the Case of Non-Existing Nash Equilibrium

This section is devoted to the description of a mechanism for unlocking the game in case Nash equilibrium is not reached. Section II described the concept of GT and Nash equilibrium. The theory tells us that in a discrete game where the solutions are finite, Nash equilibrium cannot be always guaranteed. Keeping in mind that for the utility function introduced in [101] there might be cases Nash equilibrium is not reached, as a solution to a possible implementation of the proposed technique in a real application scenario, the present section proposes an unblocking mechanism that allows the game to continue until the affected terminals are located.

The mechanism is shown to be transparent as the results indicate that it has a negligible effect on the performance. The operation of the unlocking mechanism is examined by using the drone D_1, as a reference, and bearing in mind that the established criterion also applies to the drone D_2.

As described in the previous section, the drone moves to the 4 vertices of the square (game strategies: s_1, s_2, s_3 and s_4, as can be seen in Figure 3.33) following the perimeter of the unit square in the clockwise direction. For each game strategy, the power values received by the terminals M_1 and M_2 are measured, and, based on the proposed UFs, the Nash equilibrium is calculated.

The result obtained from the Nash equilibrium is the position (one of the vertices of the square) to be taken up by the drone. If the Nash equilibrium cannot be found, the drone will position itself in the strategy (vertex of the square) that requires the least waste of energy and time, i.e. the fourth strategy s_4 (last vertex of the square). In this way, the drone D_1 does not have to move from the last position where it made the last power evaluation. Following the round in which both drones D_1 and D_2, are assigned the positions s_4 and t_4, respectively, the game continues until the terminals are located without interruptions caused by non-equilibrium.

The experimental results section will establish numerically (in percentage) the lack of Nash equilibrium out of 10000 iterations performed for each utility function.

3.8.4 Time and Energy Evaluation

In this section, we will describe how the flight time, processing time, and total energy used by drones to search for mobile terminals is estimated. The flight time, δ_f, expressed in seconds, is the time it takes the drone to move along one side of a basic cell. The processing time, δ_p, is the time it takes the drone to acquire a stable power value. These values, which depend on the Femtocell UAV technology employed, were chosen to be 10 and 30 seconds, respectively, according to the experimental evidence and technology adopted in the previous study [101].

In order to obtain the total flight time for the entire game, it is necessary to add, for each device to be located, the flight and processing time taken by the drone in each round. For each round, the optimal path for the measurement of the powers at the four vertices of a macro-cell starting from the center is given by the semi-diagonal path from the center to a vertex and the other three sides (see Figure 3.33).

Then we must add the path to reach the vertex V_N identified by the Nash equilibrium, and, therefore, a path that can take three different values: the null value (if the last vertex coincides with V_N), equal to one side (if V_N is an adjacent vertex) or the diagonal (if the vertex V_N is the opposite one).

The T_{tot}^{GT} is, therefore, given by the sum of the total flight and processing time, according to the equation below (3.43):

$$T_{tot}^{GT} = \sum_{m=1}^{M} \Big[[\sqrt{2} + 2(N_s - 1) + J]\delta_f + N_s\delta_p \Big] R_m, \qquad J = 0,\ 2,\ 2\sqrt{2}, \tag{3.43}$$

where R_m represents the number of rounds needed to localize the m-th device, N_S is the number of strategies (i.e., the four positions that each drone can take in a single round), $J\delta_f$ corresponds to the time needed for the final displacement that the drone makes in one round to reach the vertex of the macro-cell that maximizes the Nash equilibrium. From the total time we can derive the total energy consumed by the drone by (3.44):

$$E_{tot}^{GT} = P_d \frac{T_{tot}^{GT}}{3600} [Wh], \tag{3.44}$$

where P_d is the power employed by the drone and is equal to about 700 Watts [W], as can be inferred from technical datasheets related to UAVs used in similar application scenarios.

The times and energy employed by the “Serpentine” method used for comparison in this study, were derived from the formulas found in [94].

3.8.5 Simulation Results

This section discusses the main results obtained from the simulations. Particularly, a series of simulations related to scenarios with isotropic propagation (illustrated in sub-section 3.8.5) and anisotropic propagation (illustrated in sub-section 3.8.5) were carried out. The achieved results are expressed in terms of time gain, energy gain, and percentage of times Nash equilibrium does not occur. Time and energy gains, in terms of savings, were expressed through (3.45) and (3.46), respectively:

$$G_{time} = 1 - \frac{T_{tot}^{GT}}{T_{tot}^{Serp}}, \tag{3.45}$$

$$G_{energy} = 1 - \frac{E_{tot}^{GT}}{E_{tot}^{Serp}}. \tag{3.46}$$

Isotropic Signal Attenuation

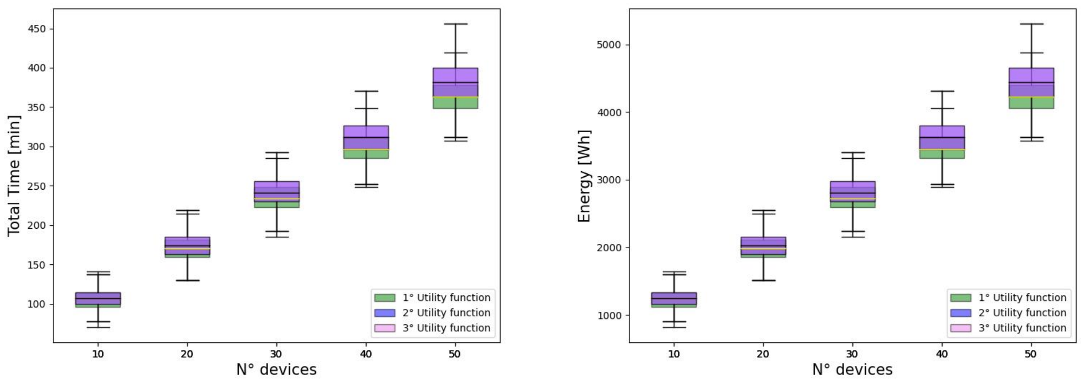
This section shows the results of a series of simulations first considering an isotropic attenuation in an empty space within the monitoring area. The tests performed are as follows:

i. Time and energy gain evaluation varying the three UFs described and the number of terminal $N = 10, 20, 30, 40, 50$;

ii. Time and energy gain evaluation varying the three UFs described and the size of the monitoring area. In [101] the monitoring area size was 24×24 meters, now the authors introduced a several performance test considering other two size: 12×12 meters and 48×48 meters. In this case, for simplicity reasons, the number of devices generated for each monitoring area was fixed to constant density: 4, 16, 64 devices for 12×12, 24×24, 48×48 areas, respectively;

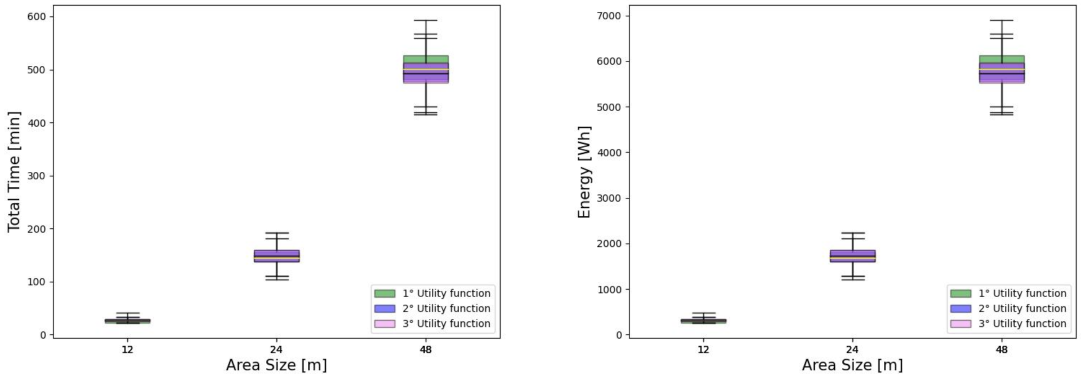
iii. The calculation of the percentage of times Nash equilibrium does not occur was performed separately for each of the three UFs as the number of devices and the size of the monitoring area vary;

iv. Once fixed monitoring area size and utility function: evaluation of the total time and energy used by the drone to operate over the monitoring area; comparison of total time and energy between the proposed method and the



(a) (b)

Figure 3.35: Isotropic Attenuation: Time (a) and Energy (b) gain comparison between differents UFs and varying number of terminals.



(a) (b)

Figure 3.36: Isotropic Attenuation: Time (a) and Energy (b) gain comparison between differents UFs and varying monitoring area size.

Table 3.3: Isotropic Attenuation: UFs compared - energy and time gain when varying the number of devices and size of monitoring area.

	Area Size [m]: 12 × 12		Area Size [m]: 24 × 24		Area Size [m]: 48 × 48	
N° Devices	UFs Compared	Time/Energy Gain [%]	UFs Compared	Time/Energy Gain [%]	UFs Compared	Time/Energy Gain [%]
10	$UF_1 > UF_2$	1.3 ± 15.9	$UF_2 > UF_1$	0.6 ± 17.9	$UF_2 > UF_1$	5.8 ± 21.3
	$UF_1 > UF_3$	1.2 ± 15.8	$UF_3 > UF_1$	0.6 ± 17.8	$UF_3 > UF_1$	5.6 ± 21.5
	$UF_3 > UF_2$	0.7 ± 11.6	$UF_3 > UF_2$	1.6 ± 17.8	$UF_3 > UF_2$	1.5 ± 19.4
20	$UF_1 > UF_2$	3.2 ± 12.1	$UF_1 > UF_2$	1.6 ± 13.3	$UF_2 > UF_1$	3.9 ± 15.2
	$UF_1 > UF_3$	3.2 ± 12.1	$UF_1 > UF_3$	1.4 ± 13.4	$UF_3 > UF_1$	4.3 ± 14.8
	$UF_3 > UF_2$	0.5 ± 9.4	$UF_3 > UF_2$	1.0 ± 13.7	$UF_3 > UF_2$	1.3 ± 13.4
30	$UF_1 > UF_2$	5.6 ± 10.2	$UF_1 > UF_2$	2.8 ± 11.3	$UF_2 > UF_1$	2.4 ± 12.1
	$UF_1 > UF_3$	5.6 ± 10.2	$UF_1 > UF_3$	2.6 ± 11.2	$UF_3 > UF_1$	2.6 ± 12.4
	$UF_3 > UF_2$	0.3 ± 8.3	$UF_3 > UF_2$	0.9 ± 11.9	$UF_3 > UF_2$	0.8 ± 11.0
40	$UF_1 > UF_2$	7.0 ± 9.2	$UF_1 > UF_2$	3.7 ± 10.1	$UF_2 > UF_1$	2.1 ± 10.1
	$UF_1 > UF_3$	7.1 ± 9.2	$UF_1 > UF_3$	3.7 ± 9.9	$UF_3 > UF_1$	2.4 ± 10.6
	$UF_3 > UF_2$	0.2 ± 7.3	$UF_3 > UF_2$	0.6 ± 10.8	$UF_3 > UF_2$	0.7 ± 9.6
50	$UF_1 > UF_2$	8.2 ± 8.5	$UF_1 > UF_2$	4.5 ± 9.1	$UF_2 > UF_1$	2.3 ± 9.7
	$UF_1 > UF_3$	8.2 ± 8.5	$UF_1 > UF_3$	4.2 ± 9.0	$UF_3 > UF_1$	2.2 ± 9.7
	$UF_3 > UF_2$	0.3 ± 6.7	$UF_3 > UF_2$	0.8 ± 10.2	$UF_3 > UF_2$	0.3 ± 8.8

"serpentine" method is described; evaluation of the gain, intended as time and energy savings, between the two methods at the variation of the number of devices, flight time and processing.

Regarding points (*i.*) and (*ii.*), the graphs in Figure 3.35 and Figure 3.36 show, respectively, the trends of time and energy gain as the number of devices and the size of the monitoring area vary in relation to the three UFs described in this study.

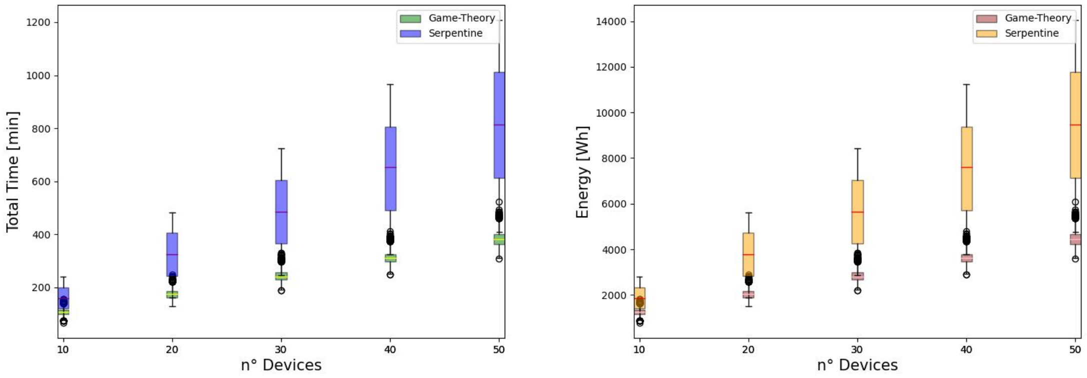
The images outline that there are no major differences in time and energy gain among the three UFs; in fact, there is a percentage difference between about

Table 3.4: Isotropic Attenuation: Percentage of non-existing Nash Equilibrium varying number of devices and monitoring area size.

N° Devices	Non-Existing Nash Equilibrium [%]								
	Area Size [m]: 12 × 12			Area Size [m]: 24 × 24			Area Size [m]: 48 × 48		
	UF_1	UF_2	UF_3	UF_1	UF_2	UF_3	UF_1	UF_2	UF_3
10	1.08	0	0	0.88	0	0	0.3	0	0
20	1.99	0	0	1.67	0	0	0.35	0	0
30	2.65	0	0	2.39	0	0	0.7	0	0
40	3.16	0	0	3.91	0	0	0.78	0	0
50	3.55	0	0	5.21	0	0	0.78	0	0

1% and 8%. To process these differences more analytically, Table 3.3 highlights the difference (in percentage) between the UFs in terms of time and energy gain as the number of devices and the size of the monitoring area vary. Regarding point (*iii.*), Table 3.4 shows the percentage of times that Nash equilibrium does not occur as the number of devices and monitoring area vary, for each utility function.

Having fixed the size of the monitoring area to 24 × 24 meters, point *(iv.)*, a comparison of total time and energy between the proposed method (using UF_3) and the "Serpentine" method is shown in Figure 3.37a and Figure 3.37b. In addition to the average reduction in the values of the proposed method by up to 41%, the comparison, highlights strong reduction in the variability of the time and energy values around the average, compared to the "Serpentine" method (to process the data analytically, including utility functions UF_1 e UF_2, see Table 3.5).



(a)

(b)

Figure 3.37: Isotropic Attenuation: Comparison between GT (UF_3) and "Serpentine" method. (a) Total Time in minute. (b) Energy in Wh.

Table 3.5: Isotropic Attenuation: Time and energy gain - GT VS Serpentine method.

N° Devices	Time and Energy Gain [%]		
	UF_1	UF_2	UF_3
10	26.0 ± 26.1	26.3 ± 25.8	26.0 ± 25.5
20	42.1 ± 19.1	40.5 ± 20.0	40.6 ± 20.1
30	46.9 ± 17.5	45.4 ± 18.1	45.1 ± 18.3
40	49.3 ± 16.6	46.7 ± 17.6	47.3 ± 17.5
50	50.4 ± 16.2	48.1 ± 17.1	48.5 ± 17.1

Table 3.6: Isotropic Attenuation: Time and energy gain varying δ_f and δ_p ($M = 50$ devices) - UF_3.

$\delta_p[s]$	$\delta_f[s]$			
	5	10	15	20
10	57.4±14.1%	62.7±11.8%	67.7±11.4%	66.7±13.6%
20	38.0±16.7%	52.9±14.3%	59.7±16.0%	65.9±12.9%
30	28.2±20.1%	41.4±14.4%	48.5±17.1%	58.6±13.7%
40	25.6±20.3%	33.0±19.3%	45.0±14.0%	50.8±15.7%

Finally, using UF_3, Table 3.6 illustrates differences in the gain when δ_f and δ_p are varied in the case of $M = 50$ devices. Moreover, the table shows that the gain increases significantly by using the proposed method as the processing time decreases, especially for low time-of-flight values.

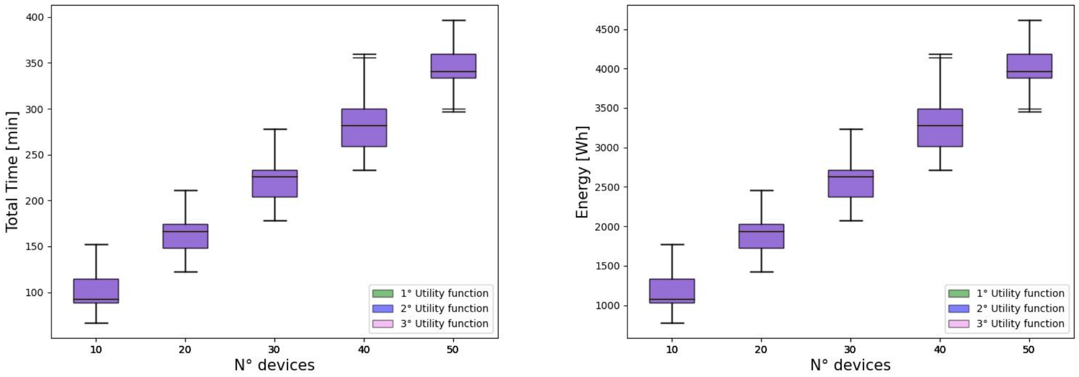
Non-isotropic Signal Attenuation

This section will repeat some of the simulations conducted and illustrated in the previous sub-section, considering an attenuation of the non-isotropic type within the monitoring area. The tests performed are as follows:

i. Time and energy gain evaluation varying the three UFs described and the number of terminals $N = 10, 20, 30, 40, 50$;

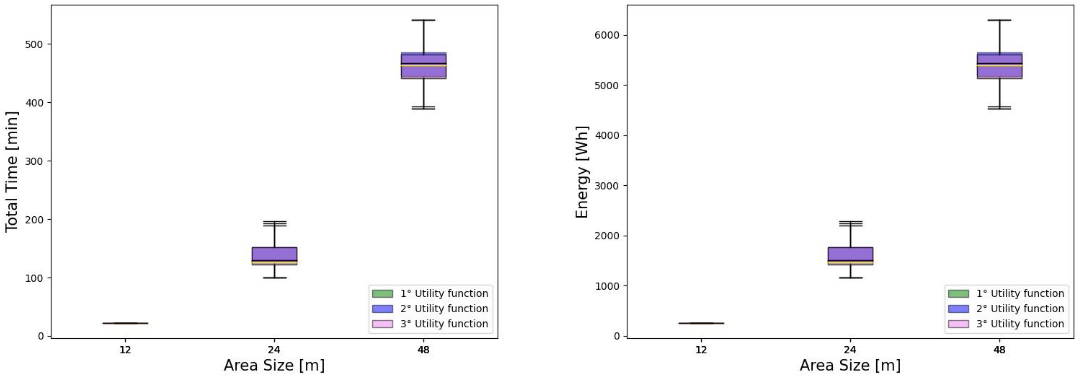
ii. Time and energy gain evaluation varying the three UFs described and the size of the monitoring area. The monitoring area size was 24 × 24 meters, but the authors have additionally introduced several performance tests considering other two monitoring area sizes: 12 × 12 meters and 48 × 48 meters;

iii. The calculation of the percentage of times Nash equilibrium does not occur was performed separately for each of the three UFs as the number of devices and the size of the monitoring area vary;



(a) (b)

Figure 3.38: Non-isotropic Attenuation: Time (a) and Energy (b) gain comparison between differents UFs and varying number of devices.



(a) (b)

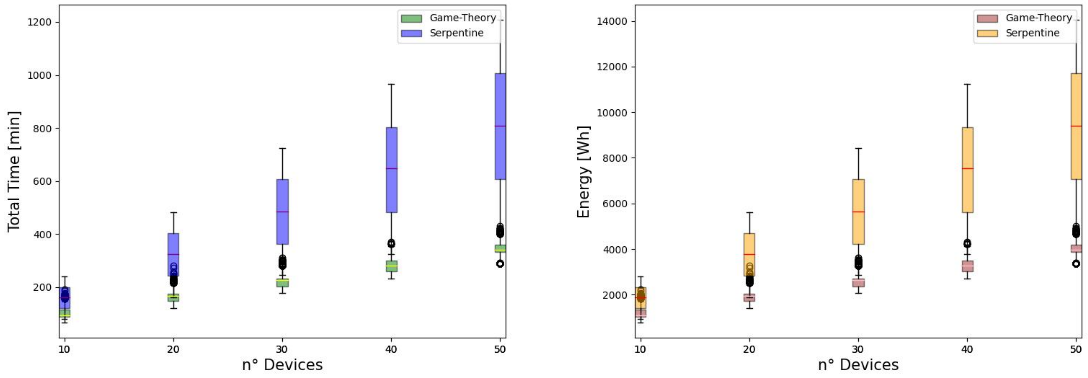
Figure 3.39: Non-isotropic Attenuation: Time (a) and Energy (b) gain comparison between differents UFs and varying monitoring area size.

iv. Once fixed monitoring area size and utility function was fixed, the evaluation of the total time and energy used by the drone to operate over the monitoring area was performed; comparison of total time and energy between the proposed method and the "Serpentine" method was described.

Regarding the points (i.) and (ii.), the graphs in Figure 3.38 and Figure 3.39 show, respectively, the trends of time and energy gain as the number of devices and the size of the monitoring area vary in relation to the three UFs described in this study. Similarly to the previous case (isotropic attenuation), the images outline that there are no major differences in time and energy gain among the three UFs (between 1% and 7%). To process these differences more analytically, Table 3.7 shows the difference (in percentage) between the UFs in terms of time

Table 3.7: Non-isotropic Attenuation - UFs compared: Energy and time gain when varying the number of devices and size of monitoring area.

	Area Size [m]: 12 × 12		Area Size [m]: 24 × 24		Area Size [m]: 48 × 48	
N° Devices	UFs Compared	Time/Energy Gain [%]	UFs Compared	Time/Energy Gain [%]	UFs Compared	Time/Energy Gain [%]
10	$UF_2 > UF_1$	0.1 ± 3.7	$UF_2 > UF_1$	6.4 ± 35.3	$UF_2 > UF_1$	4.9 ± 38.6
	$UF_3 > UF_1$	0.1 ± 4.5	$UF_3 > UF_1$	6.1 ± 35.1	$UF_3 > UF_1$	5.7 ± 38.3
	$UF_3 > UF_2$	0.0 ± 2.8	$UF_3 > UF_2$	5.1 ± 34.9	$UF_3 > UF_2$	7.6 ± 39.9
20	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_2 > UF_1$	1.9 ± 20.5	$UF_2 > UF_1$	2.3 ± 23.3
	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_3 > UF_1$	1.8 ± 20.6	$UF_3 > UF_1$	2.1 ± 23.6
	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_3 > UF_2$	2.0 ± 20.6	$UF_3 > UF_2$	2.4 ± 23.9
30	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_2 > UF_1$	0.9 ± 14.6	$UF_2 > UF_1$	1.0 ± 16.8
	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_3 > UF_1$	1.0 ± 14.5	$UF_3 > UF_1$	0.4 ± 15.8
	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_3 > UF_2$	1.2 ± 14.6	$UF_3 > UF_2$	0.9 ± 16.6
40	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_2 > UF_1$	0.7 ± 11.2	$UF_2 > UF_1$	1.8 ± 13.6
	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_3 > UF_1$	0.6 ± 11.2	$UF_3 > UF_1$	1.7 ± 13.5
	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_3 > UF_2$	0.5 ± 11.0	$UF_3 > UF_2$	0.8 ± 13.6
50	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_2 > UF_1$	0.5 ± 9.0	$UF_2 > UF_1$	1.0 ± 11.3
	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_3 > UF_1$	0.5 ± 9.0	$UF_3 > UF_1$	0.4 ± 11.1
	$UF_1 \approx UF_2 \approx UF_3$	0	$UF_3 > UF_2$	0.5 ± 9.1	$UF_3 > UF_2$	0.1 ± 11.2



(a)

(b)

Figure 3.40: Non-isotropic Attenuation: Comparison between GT (UF_3) and "Serpentine" method. (a) Total Time in minute. (b) Energy in Wh.

and energy gain as the number of devices and the size of the monitoring area vary. Regarding point (iii.), the simulations show that all three UFs always have Nash equlibrium.

Having fixed the monitoring area size to 24 × 24 meters, a comparison of the total time and energy between the proposed method (UF_3) and the "Serpentine" method is shown in Figure 3.40a and Figure 3.40b. In addition to an average reduction in the values of the proposed method by up to 46%, the comparison highlights strong reduction in the variability of the time and energy values around the mean, compared to the "Serpentine" method (to process the data analytically, including for the utility functions UF_1 and UF_2, see Table 3.8).

Table 3.8: Non-isotropic Attenuation: Time and Energy gain - GT VS Serpentine method.

N° Devices	Time and Energy Gain [%]		
	UF_1	UF_2	UF_3
10	32.5 ± 27.3	33.2 ± 26.9	33.2 ± 26.6
20	44.6 ± 19.5	44.8 ± 19.6	44.8 ± 19.6
30	49.6 ± 17.0	49.2 ± 17.2	49.4 ± 17.1
40	52.0 ± 15.9	51.9 ± 15.8	51.8 ± 16.0
50	53.8 ± 15.1	53.6 ± 15.1	53.6 ± 15.2

3.8.6 Discussion and Conclusion

Analyzing the results obtained from the simulations based on different assumptions in this study, the following considerations can be made:

- GT is confirmed to be a suitable technique for drone energy and flight time optimization in terminal geolocalization scenarios for civil protection purposes;

- The introduction of new UFs allows for a better identification of the utility function with greater robustness rather than reaching 100% Nash equilibria due to the fact that the percentage of times the Nash equilibrium is not reached is so low that there is no effect on performance;

- The percentage of cases in which Nash equilibrium of the UF_1 utility function does not exist varies from about 1% to 5% as the size of the monitoring area increases. The impact of the release mechanism on the performance of UF_1 is, in general, negligible.

- The introduction of scenarios with non-isotropic propagation assumptions actually confirms the results of the ideal scenario characterized and modeled with the simple Friis Transmission equation of isotropic propagation;

- In particular, while in an isotropic propagation scenario utility function UF_1 performs better in smaller areas, in real cases of non-isotropic propagation UF_3 performs better, it is especially true in cases of reduced number of devices to be geolocated;

- Slightly larger gain in the non-isotropic case is due to the smaller number of rounds (in general, the observed reduction was 10%) required for geolocalization, given the unavoidable larger error that the non-isotropic case presents;

- The small differences between the three UFs that emerge from this study may suggest the possibility of applying dynamic fitting methods, which would certainly result in superior performance.
- In a more realistic scenario, such as that of mobile terminals attenuated by post-earthquake rubble, characterized by radio-frequency signals propagating in a non-isotropic manner, GT always finds the Nash equilibrium and increases the energy savings and, thus, the duration of the flight by about 10%, at the expense of a consequent slight increase in geolocalization error.

Chapter 4

Multimodal rainfall classification systems based on Convolutional Neural Networks

The sudden climate change, which has taken place in recent years, has generated calamitous phenomena linked to hydrogeological instability in many areas of the world. The main existing rainfall level measurement methods employ rain gauges, weather radars and satellites [103–105]. The rain gauge is probably the most common rainfall measurement device, as it is able to provide an average accurate estimate of the rainfall with a precise temporal resolution; in fact, rain gauges continuously record the level of precipitation even within short time intervals. Modern tilt rain gauges consist of a plastic manifold balanced on a pin. When it tips, it actuates a switch which is then electronically recorded or transmitted to a remote collection station. Unfortunately, tipping buckets tend to underestimate the amount of rainfall, particularly in snowfall and heavy rainfall events. Moreover, they are also sensitive to the inclination of the receiver and different types of dirt that may clog the water collection point.

Many studies have been carried out regarding the classification of rainfall levels using alternative methods, parameters, and signals such as video, audio, and radio signals [106–108].

This study propose a wider set of rain levels, including the “No rain” class and adding the “Shower” and “Cloudburst” rain classes. In places without a power grid (e.g., agriculture and smart roads), this system can be powered by a low-power photovoltaic panel, enabling recording at a rate proportional to the rain level, thus preserving the average lifetime of the electronic and microphone sensor components. Statistics on the average weather that characterizes a rainfall event in the territory allow us to state that the reliability and average duration of an audio rain gauge is comparable to that of a UE. The operating temperatures of an acoustic rain gauge, in fact, are the same as those of a UE. Furthermore, the microphones are designed for large temperature and humidity levels. The introduction of an acoustic rain gauge is, thus, justified, particularly, in contexts where it may be necessary to reduce the risks caused by sudden "showers" or "cloudbursts" with low operational investment, management and maintenance costs. Some application contexts concern smart cities for which is foreseen the integration of an audio sensor inside street lamp, precision agriculture [109], with the advantage of being able to adapt the irrigation flows in a complementary way to different rain levels, as well as highway safety, by minimizing the risks of aquaplaning [110, 111].

In particular, during the PhD period, new algorithms for classifying rainfall levels using different types of signal and new deep learning were studied. The results related to rainfall classification and detection, obtained from such studies, are promising and quite accurate. In sub-section 4.1, the study [112, 113] related to rainfall classification using audio signal and deep learning techniques is illustrated. This study has evolved to a real-time embedded implementation [114] (see sub-section 4.2). In sub-section 4.3 is related to the use of video recordings of rain as an input signal for the identification and classification of rainfall levels [115]. Finally, sub-section 4.4 proposes a first study on rain intensity classification using LTE signal radio parameters and deep learning techniques [116].

4.1 Classification of Rainfall Estimation Using Audio Signal

The possibility to classify rain using systems based on the classification of the audio signal could be especially useful in the context of smart cities or in precision agriculture [109] and highway safety [110, 111].

In this sub-section, studies conducted on this topic during the PhD period will be described. The sub-section is organized as follows: sub-section 4.1.1 briefly summarizes the main studies regarding the rainfall classification based on audio signal, followed by sub-section 4.1.2 that described the dataset created for this study; sub-section 4.1.3 described the proposed method to rainfall classification using audio signal and CNN nentwork, including: acquisition system setup and labeling algorithm; sub-section 4.1.4 shows the results obtained using the CNN neural network; and, finally, there are the discussion and conclusions.

4.1.1 Related Work

In [45], rain is estimated through acoustic sensors and Android smartphones. Using this system has several advantages: access to the data collected via the device's microphone or camera; data can be sent via radio functions (WI-FI, GSM, LTE, etc.); and the cost-effectiveness of Android devices compared to high-precision meteorological instruments. The audio data collected by the smartphone microphone are then processed to extract the fundamental parameters to be compared with the critical thresholds, and exceeding these thresholds, translates into sending an "alarm". In addition, historical data are stored on a web server allowing remote access to information. From the audio sequence sampled at 22.05 KHz the calculation of the signal strength (dB) is performed every 5 s. To verify the validity of the results obtained with this method, measurements were carried out simultaneously with the aid of a tilting rain gauge so that the data of the two solutions can be compared.

The study carried out in [117], aims at the digital representation of rain scenes as a function of the real sounds produced by the precipitation. The research is based on the study of rainfall phenomena proposed by Marshall and Palmer, providing the frame for describing the probability distribution of the number and size of raindrops in a space of known volume, evaluating the intensity of the event, and the fact that the speed of a water drop is only related to its size.

Based on these considerations, it is possible to extract parameters relating to raindrops, useful for creating an animated digital scene. A rainy phenomenon can be modelled according to the Marshall–Palmer distribution which allows determining the relationship between the number and size of the drops in a volume as a function of the level of intensity (mm/h) obtainable from the audio recordings of actual rainfall. These were obtained using a smartphone and employing the same previously described procedure [45], but with sampling at a frequency of 44.10 KHz and, simultaneously, carrying out measurements using a rain gauge. In order to extract the frequency characteristics, various rainfall recordings of different intensity were taken and, for each, the Fourier transform was performed. One of the most evident results is that the samples in the spectrum present a high level of correlation and the high frequency components increase as the intensity of the precipitation increases. This can be interpreted by the Marshall–Palmer distribution, in fact, the number of occurring raindrops per unit of volume increases as the intensity of the rain increases, the average landing time of the drop is reduced and, consequently, the frequency of sounds increases.

The input sounds were taken from a data set, randomly selecting eight 1-s sound segments for each data, with 15 types of rainfall intensity. Once the level of rainfall has been determined, it is possible to trace the following parameters: size of the rain particles, number of drops, and falling speed.

Another technique used, was a machine learning approach of analysis and classification in which the sound classes to be analyzed are defined in advance [118].

Aiming to overcome some limitations present in traditional techniques to signal classification, in [112] we proposed an acoustic rain gauge based on convolutional neural network (CNN), in order to obtain accurate classification of rainfall levels. The paper presents a classification algorithm for the acoustic timbre produced by the rain in four intensities, i.e., "Weak rain", "Moderate rain", "Heavy rain", and "Very Heavy rain"; also the paper studies and compares the performance of an acoustic rain gauge in four different types of materials used to cover the microphone. In this study, the sound of rain was derived from the impact of water drops on a material covering the microphone.

In [119] authors were able to subdivide precipitation detected by a series of piezoelectric sensors into intensity classes through SVM, KNN and Naïve-Bayes machine learning algorithms. The quantities involved in the study are sound pressure and rainfall height.

4.1.2 Rainfall Audio Dataset

The database was created ad hoc using the acquisition system represented in Figure 4.1. This system consists of a microphone and a tilting rain gauge connected to a processing unit. The processing unit houses an algorithm for labeling the audio files that allows aggregating different audio sequences in the corresponding class of precipitation intensity. Once the audio sequences are labeled in their respective intensity classes, they are entered in the database. The acquisition system, samples audio sequences at a frequency of 22.05 KHz at 16 bit (PCM format). The database consists of seven precipitation intensity categories, defined in Table 4.1; for each category there are 10 audio sequences each lasting 30 s.

The categories were decided upon by taking the national classification scales [120] as a reference, and slightly modifying some ranges in order to obtain a number of adequate and homogeneous examples corresponding to different classes.

The current database was created by recording the rain on five different days, some of which were characterized by the presence of different wind and environmental noise, taken at different locations, such as gardens, terraces, and countrysides, obtaining examples of all seven rainfall level classes. This implies a certain robustness of the system, since the obtained results take into account the addi-

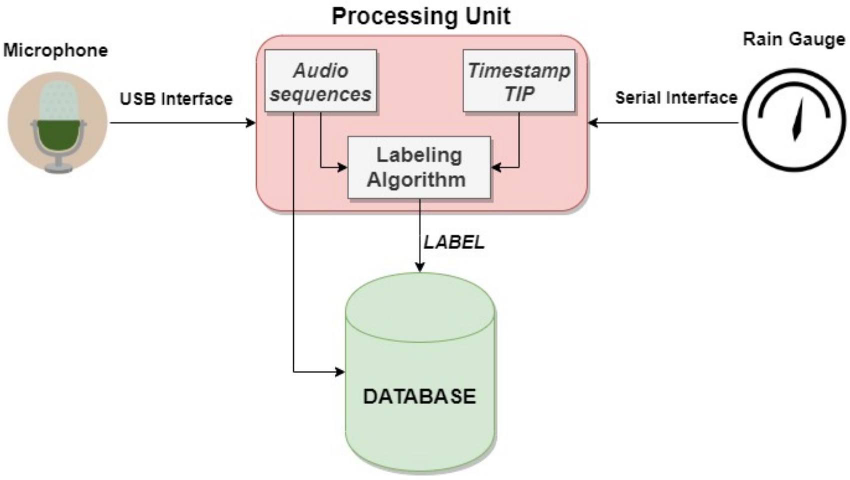


Figure 4.1: Block scheme of audio acquisition system.

Table 4.1: Classification and rainfall intensity range.

Classification	Acronym	Rainfall Intensity [mm/h]
No rain	nr	<0.5
Weak	w	[0.5 ÷ 2]
Moderate	m	[2 ÷ 6]
Heavy	h	[6 ÷ 10]
Very heavy	vh	[10 ÷ 18]
Shower	s	[18 ÷ 30]
Cloudburst	c	>30

tional external variable noise. Moreover, during the rain recording phase, continuous checks were carried out so that no dirt was created in the pan rain gauge used for labeling the rain level. Finally, through the simultaneous audio-video recording, the database was verified and cleaned up by eliminating sequences with wrong levels, through a repeated listening and contextual video verification phase.

The audio sequences were provided, 10 for each class of seven different rainfall levels. Subsequently, each audio sequence, lasting 30 s, is cut out, using the slide window algorithm, for the creation of CNN input files.

In particular, the offset chosen is 100 milliseconds while the duration of the time window is 3 s. Once the audio sequences are cropped, 16,746 one-second sequences are obtained, corresponding to the seven levels of rain (including “No rain” level).

Once the crop is obtained, 70% of these sequences are placed in the training set and 30% in the test set. In this way the dataset contains 11,722 elements in the training set and 5024 elements in the test set.

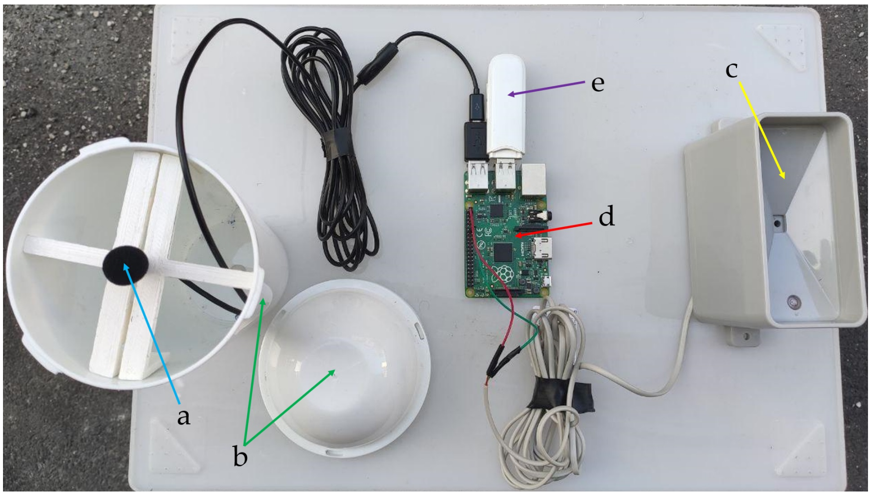


Figure 4.2: Hardware components of the audio acquisition system.

Once the dataset is created, the audio signal is under-sampled at 22.05 kHz. In this phase the signal is normalized with mean and standard deviation and is fed as input to the CNN network. The percentage of probability corresponding to each individual class will appear in the output.

4.1.3 Proposed Method

This sub-section is devoted to defining and describing the key elements of our study, such as: labeling algorithm, and testing procedure.

Acquisition System Setup

The tests were conducted using mainly two instruments related to the measurement of rainfall intensity: a microphone inserted into a rigid plastic shaker. The signals related to the various timbres of the rain sounds are used as the input variables of a pattern recognition system based on a CNN classifier. The scenario is characterized by real rain which, for the total duration of the test, covered the seven rainfall intensities shown in Table 4.1.

The system implemented and studied in this paper is characterized by a device capable of detecting the rain audio data, when it falls on a plastic surface. In particular, we are interested in obtaining a more efficient and faster classification of the different levels of rainfall intensity. The acoustic rain gauge, see Figure 4.2, is characterized by the following components:

- Microphone (a);
- Plastic shaker (b);

- Tipping bucket rain gauge (c);
- Raspberry Pi, used for data processing (d);
- 4G dongle for data transmission in the cloud (e).

The microphone sensor is connected via USB cable to the Raspberry Pi, where a processing phase of the collected data is carried out. The processing unit, in fact, implements the labeling algorithm described above, enabling the generation of 30-s audio sequences, which correspond to a certain class of rainfall intensity. The obtained data are sent via a cellular connection (4G dongle) to the cloud, entered in the database and subsequently given as input to the neural network for the training phase.

The rain gauge tray is therefore used for the algorithm that labels rainfall levels allowing, in turn, to create the testing and training database for the neural network. It is connected to the processing unit via an RJ11 cable and is managed ad hoc through a software interface capable of detecting and counting the "interruptions" generated by the rain gauge tray every time a tilt occurs. Subsequently, to obtain the estimate in mm/h, the acquired value was multiplied by a factor dependent on the time span required in order to estimate the level of rainfall in an hour, assuming constant rain distribution.

Every minute the obtained values are sent to an IoT platform, via the publish/-subscribe protocol. The values taken from the tray are used to label the audio signal at different rain intensities. The audio files labeled in such way define the dataset for the training and test phase of the convolutional neural network, specifically structured for this specific application context.

The general scheme of the proposed system is shown in Figure 4.3.

Labeling Algorithm

Once the structure and characteristics of the adopted database are defined, it is possible to describe the labeling algorithm used for the recognition and classification of the intensity of the rain. The preliminary requirements of the algorithm are:

- Audio file recorded in steps of X seconds;
- Timestamp of the tip, of the tipping rain gauge, stored in the database.

The characteristic parameters of the algorithm are defined below:

- T_i is the final instant of the $i-th$ time window of the audio signal;

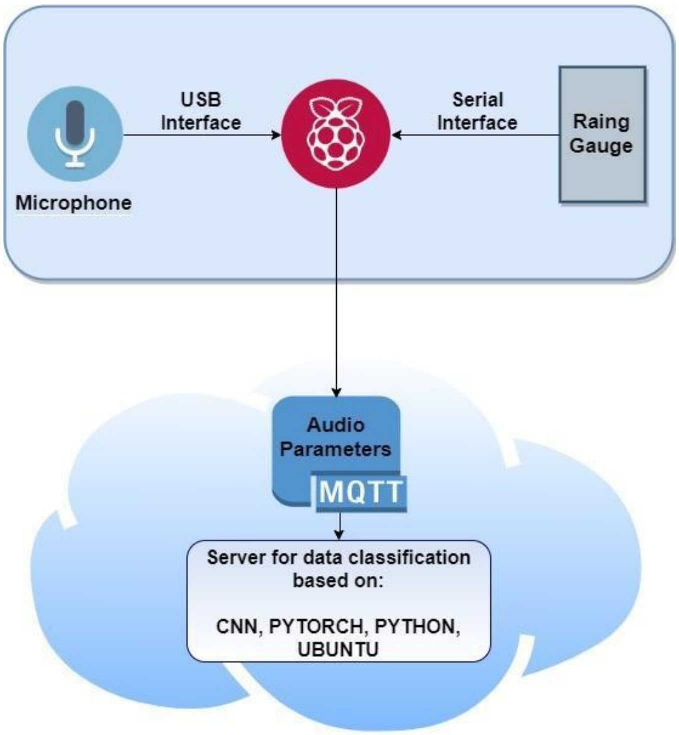


Figure 4.3: Conceptual scheme of the audio acquisition system.

- C is the capacity, in millimeters, of the tipping bucket rain gauge for water collection;
- R_k is the instant of the $k-th$ bucket rotation following T_i;
- $R_{(k-1)}$ is the instant of $k-th-1$ bucket rotation, prior to T_i;
- T_s is the time interval between R_k and $R_{(k-1)}$;
- T_h is the hour expressed in seconds, i.e., 3600 s.

The labeling algorithm is responsible for calculating the T_s and subsequently, using (4.1), we obtain the estimate of the level of precipitated rain, expressed in millimeters.

The first phase of the labeling algorithm is, therefore, the calculation of the T_s which may be performed differently, based on different cases:

i. If $R_k = 0$ and $R_{(k-1)} = 0$, we will have $T_s = 0$;

ii. If $R_{(k-1)} = 0$ and $R_k \neq 0$, we will have $T_s = R_{k-T_i}$;

iii. If $R_{(k-1)} \neq 0$ and $R_k \neq 0$, we will have $T_s = R_{k-R_{(k-1)}}$.

So, once T_s, is obtained, the labeling algorithm is as follows:

i. If $T_s > T_h$ the minimum estimate in mm/h is equal to 0; and

ii. If $T_s \leq T_h$ the estimate in mm/h is calculated with the following formula:

$$e = C \cdot \frac{T_h}{T_s} \tag{4.1}$$

The result of (4.1) is compared with the rain classes, defined in Table 4.1, for the labeling of the audio files in the database.

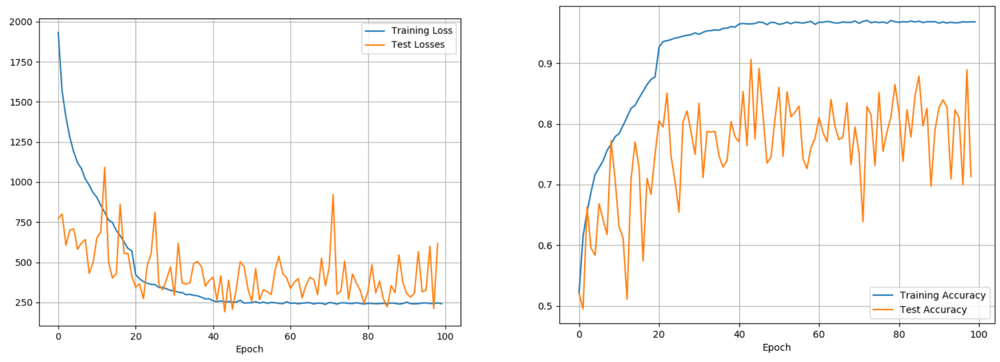
4.1.4 Performance Evaluation

In this section we will analyze the results obtained by applying the machine learning technique, with audio signals placed as input to the CNN (described in Appendix C). The results obtained when we input the audio sequences in our database (audio sequences concerning all rainfall categories) are visible in the Figure 4.4.

Figure 4.4a shows the progress of the training losses (blue curve) and the test losses (orange curve); both curves decrease with increasing epochs. Conversely, Figure 4.4b shows that the trend of training accuracy and test accuracy increases with increasing epochs. The two graphs are complementary, as accuracy increases, the loss for training and testing decreases. This implies that the neural network is performing an accurate classification.

For greater understanding of the results obtained by the neural network, see the confusion matrix in Figure 4.5.

Figure 4.5 shows the confusion matrix obtained by processing the test audio sequences inputted to the CNN network. It is possible to observe good classification



(a) (b)

Figure 4.4: Rainfall audio - training and test phase: (a) training and test losses trend. (b) training and test accuracy trend.

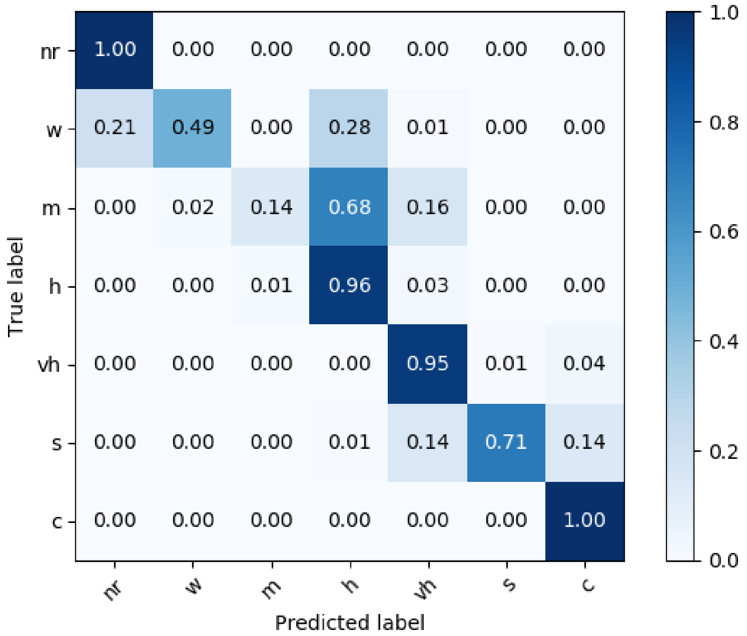


Figure 4.5: Rainfall audio - confusion matrix.

of the seven levels of rainfall intensity with an average percentage of 75%, which rises to 93% if the miss-classifications of the adjacent classes are not considered. In particular, we have an excellent classification for “No rain” (100%), “Heavy” (96%), “Very heavy” (95%), and “Cloudburst” (100%). The level of accuracy in percentage can also be improved by inserting a post-processing block represented for example by a median filter.

4.1.5 Discussion and Conclusion

This study presents a technique that allows classifying the different levels of rainfall intensity accurately and fast. In particular, we have shown how it is possible to classify the spectral and statistical parameters of an audio signal by means of the machine learning technique known as the convolutional neural networks (CNN).

Possible future studies to improve and extend the classification technique are the following:

- Divide the “Cloudburst” level into further sub-categories and apply the same analysis presented in this paper to have a good classification;
- As regarding spectral analysis, consider reducing the sampling frequency so as to send a smaller number of parameters to the input of the CNN;

- Add the spectral and/or statistical parameters to a DNN input, which is a neural network more suited to manage a reduced set of audio parameters;
- Study the "signature" of rain sound based on patterns interpretation in hidden CNN layers from spectrograms of the rain.

The paper proposes an innovative acoustic rain gauge based on CNN. An analysis of the different statistical and spectral characteristics of the acoustics produced by the rain at various rainfall levels is presented. In particular, the system is very simple, being based on a plastic shaker, a microphone, and a low-cost/low-power signal processing unit. The performance is very good in terms of accuracy and ability to adapt to sudden changes in precipitation intensity. It should be borne in mind that, especially in low rainfall, the peak rain indicator used to label the database has a low temporal resolution. Thus, taking into account the typical micro-variances of rainfall intensity, it is possible to consider an average accuracy of 93%, assuming that overall system performance does not include miss-classification between adjacent classes. The new acoustic rain gauge exceeds the limits of traditional ones, having no mechanical parts and requiring no maintenance. In general, the proposed solution is adequate for precipitation level monitoring service, with its major advantage being a totally electronic system that can be easily integrated on existing platforms and systems.

4.2 Classification of Rainfall Estimation Using Audio Signal: Implementation on Embedded Board

In this sub-section the precedent studies is extended in order to proposes a new pipeline for the realization of a system for the classification of rain in real-time. In particular, a different CNN [121] has been implemented in this pipeline providing better accuracy than our previous study [112, 113] (see sub-section 4.1). Furthermore, thanks to the use of the new pipeline, inference times are significantly reduced. This is done by converting the Keras model into TensorFlow (TF) Lite optimized specifically for execution in embedded devices.

4.2.1 Rainfall Classification in Real-time

The proposed study implements a new type of CNN that allows the introduction of a real-time rain intensity classification system through the audio signal. The audio signal is, in fact, generated by the acoustic effect that the rain generates

Table 4.2: Classification and rainfall intensity range: 5 Classes

Classification	Acronym	Rainfall Intensity [mm/h]
No rain	nr	<0.5
Moderate	m	[0.5 ÷ 6]
Heavy	h	[6 ÷ 10]
Shower	s	[10 ÷ 30]
Cloudburst	c	>30

when it hits a hard plastic surface, under which a microphone connected to a Raspberry Pi model 4 is located. With the same mechanism described in [113] we created a dataset based on audio rainfall record (this time of two seconds with a sample rate of 16KHz). This audio was previously recorded by a microphone and a mechanic rain gauge in order to assign this label registered to mechanic rain gauge to the relative audio signal (see sub-section). Then we have trained the CNN network described in "*Rainfall Audio Classification with CNN in Keras TensorFlow and TensorFlow Lite*" sub-section and using TensorFlow Lite library we transferred the model to Raspberry Pi in order to record two seconds of audio record in loop and send it to the pretrained CNN in order to predict the rainfall label.

Audio dataset and features

The recorderd clips, which are of different duration, are cut out, using the slide window algorithm, for the creation of CNN input files, the same used in [113].

In particular, the offset chosen is 100 ms while the duration of the time window this time is 2 seconds and a sample rate of 16KHz. Once the audio files are cropped, 70915 one-second files are obtained, corresponding to the 5 levels of rain (see Table 4.2)

The following files have been obtained:

- 39600 audio files belonging to *nr* classes;
- 22411 audio files belonging to *m* classes;
- 4976 audio files belonging to *s* classes;
- 3241 audio files belonging to *h* classes;
- 687 audio files belonging to *c* classes.

Once the crop is obtained, 70% of these files are placed in the training set and 30% in the test set. With this process we have 70915 files belonging to 5 classes and we are using 49641 files for training and 21274 files for validation.

Rainfall Audio Classification with CNN in Keras TensorFlow and TensorFlow Lite

Convolutional neural networks process data that have a well-known grid topology. The data can be, for example, time-series data or image data that can be represented, respectively, as a 1-D and 2-D grid [121]. In this paper we used the CNN architecture shown in Figure 4.6 used in [121] for speaker recognition with pre-processed audio with FFT.

As we can see in the following snippet of code.

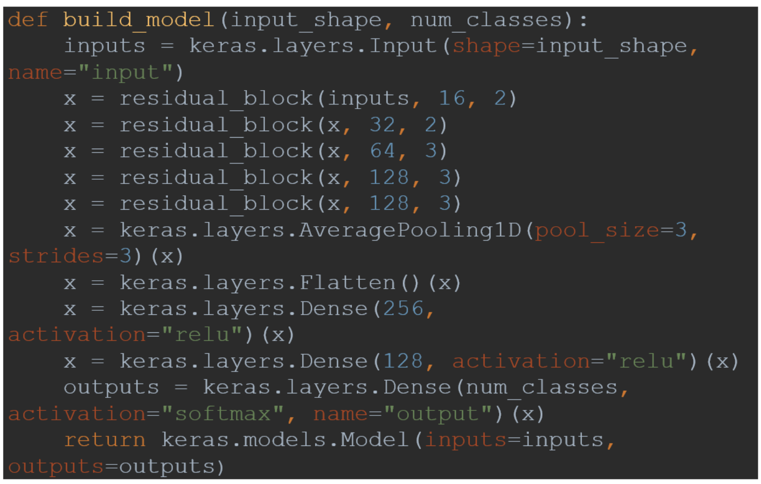
This CNN is composed of the first layer with an input shape of $(Sampling_{Rate} \cdot Rec_{Duration}, Num\_of\_Channel)$ that in this case is $(16000 \cdot 2, 1)$.

Then there are 4 different layers of residual block defined from the following code snippet, Figure 4.7.

Here the residual block is defined starting from input parameters that are:

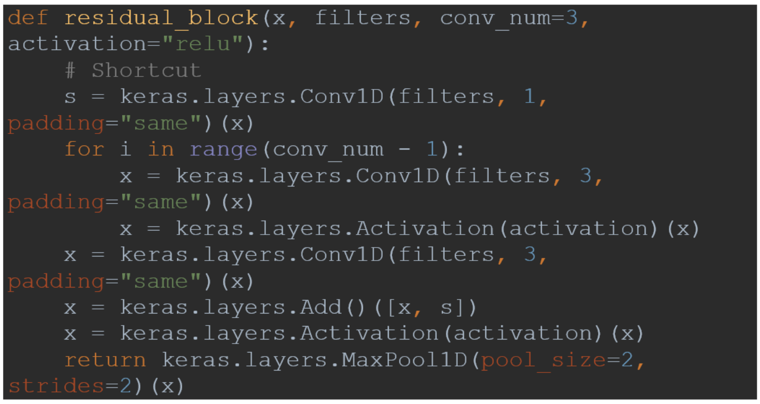
- Input of residual block: x
- Number of filter inside the convolution;
- Number of convolution inside the residual block.

Once the network is defined, as Keras require we compile it with an Adam optimizer and use sparse categorical crossentropy as loss function. In order to save



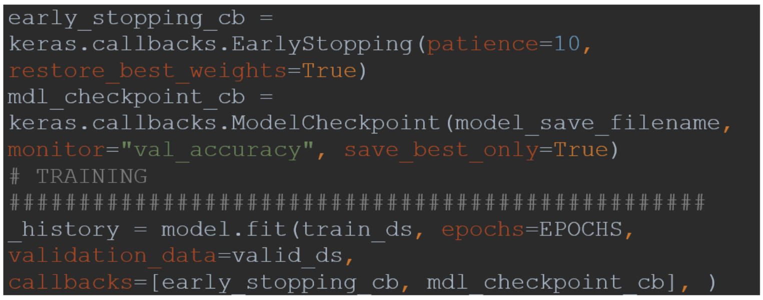
```
def build_model(input_shape, num_classes):
    inputs = keras.layers.Input(shape=input_shape,
name="input")
    x = residual_block(inputs, 16, 2)
    x = residual_block(x, 32, 2)
    x = residual_block(x, 64, 3)
    x = residual_block(x, 128, 3)
    x = residual_block(x, 128, 3)
    x = keras.layers.AveragePooling1D(pool_size=3,
strides=3)(x)
    x = keras.layers.Flatten()(x)
    x = keras.layers.Dense(256,
activation="relu")(x)
    x = keras.layers.Dense(128, activation="relu")(x)
    outputs = keras.layers.Dense(num_classes,
activation="softmax", name="output")(x)
    return keras.models.Model(inputs=inputs,
outputs=outputs)
```

Figure 4.6: Audio classification CNN architecture - Code.



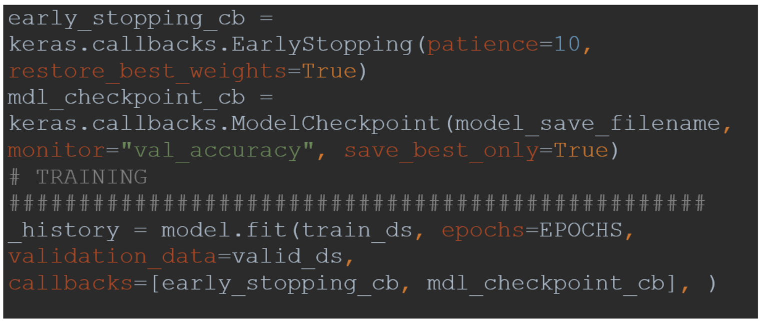
```
def residual_block(x, filters, conv_num=3,
activation="relu"):
    # Shortcut
    s = keras.layers.Conv1D(filters, 1,
padding="same")(x)
    for i in range(conv_num - 1):
        x = keras.layers.Conv1D(filters, 3,
padding="same")(x)
        x = keras.layers.Activation(activation)(x)
    x = keras.layers.Conv1D(filters, 3,
padding="same")(x)
    x = keras.layers.Add()([x, s])
    x = keras.layers.Activation(activation)(x)
    return keras.layers.MaxPool1D(pool_size=2,
strides=2)(x)
```

Figure 4.7: Residual block of 1D audio classification CNN.



```
early_stopping_cb =
keras.callbacks.EarlyStopping(patience=10,
restore_best_weights=True)
mdl_checkpoint_cb =
keras.callbacks.ModelCheckpoint(model_save_filename,
monitor="val_accuracy", save_best_only=True)
# TRAINING
####################################################
_history = model.fit(train_ds, epochs=EPOCHS,
validation_data=valid_ds,
callbacks=[early_stopping_cb, mdl_checkpoint_cb], )
```

Figure 4.8: Callbacks to save the best model during training.



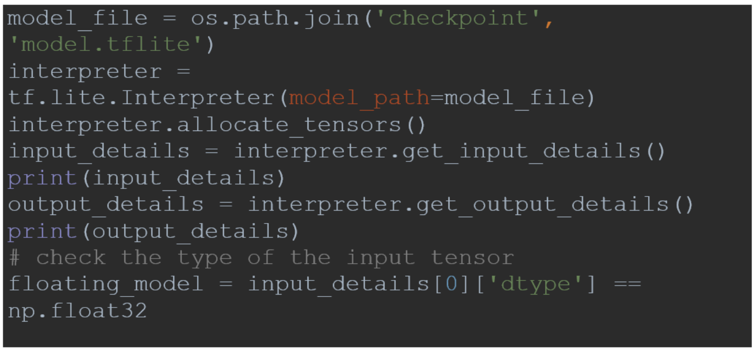
```
early_stopping_cb =
keras.callbacks.EarlyStopping(patience=10,
restore_best_weights=True)
mdl_checkpoint_cb =
keras.callbacks.ModelCheckpoint(model_save_filename,
monitor="val_accuracy", save_best_only=True)
# TRAINING
####################################################
_history = model.fit(train_ds, epochs=EPOCHS,
validation_data=valid_ds,
callbacks=[early_stopping_cb, mdl_checkpoint_cb], )
```

Figure 4.9: Model conversion snippet from Keras to Tensor Flow Lite.

only the best train epochs we define the following callbacks inserted in function fit used for training in Keras, Figure 4.8:

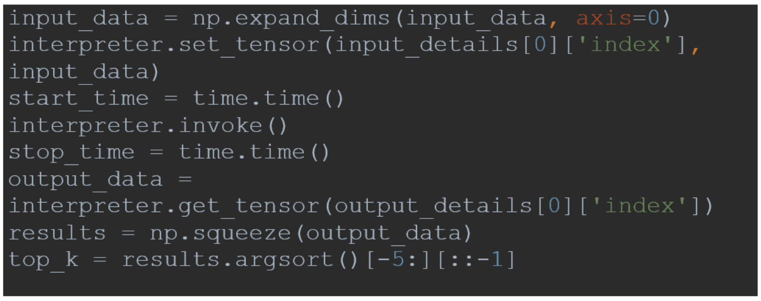
Once the network is trained, we use TensorFlow Lite converter in order to convert the Keras Model *(.h5)* into a TensorFlow Lite model *(.tflite)*. The snippet, in Figure 4.9, of code takes the model in *.h5* format and loads it into a Keras model and then converts this model into a Lite model. This operation is performed in order to reduce the complexity representation of the model converted to the Lite version.

As shown in the experimental result this improves the execution time as a



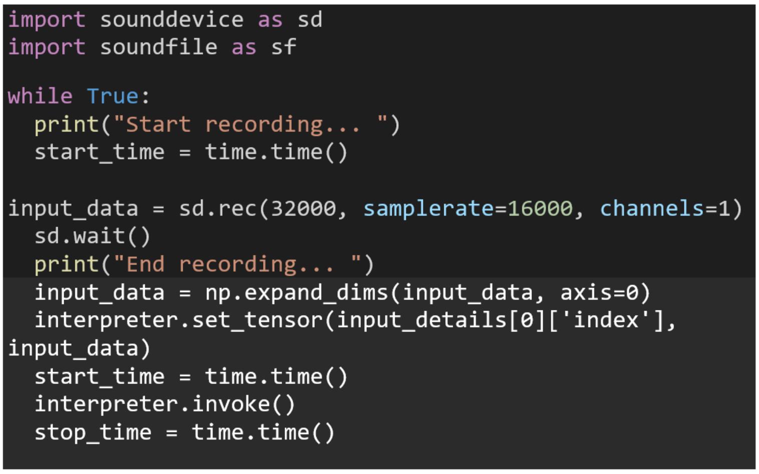
```
model_file = os.path.join('checkpoint',
'model.tflite')
interpreter =
tf.lite.Interpreter(model_path=model_file)
interpreter.allocate_tensors()
input_details = interpreter.get_input_details()
print(input_details)
output_details = interpreter.get_output_details()
print(output_details)
# check the type of the input tensor
floating_model = input_details[0]['dtype'] ==
np.float32
```

Figure 4.10: Lite model loading into interpreter.



```
input_data = np.expand_dims(input_data, axis=0)
interpreter.set_tensor(input_details[0]['index'],
input_data)
start_time = time.time()
interpreter.invoke()
stop_time = time.time()
output_data =
interpreter.get_tensor(output_details[0]['index'])
results = np.squeeze(output_data)
top_k = results.argsort()[-5:][::-1]
```

Figure 4.11: Lite model inference on data and time benchmark.



```
import sounddevice as sd
import soundfile as sf

while True:
  print("Start recording... ")
  start_time = time.time()

input_data = sd.rec(32000, samplerate=16000, channels=1)
  sd.wait()
  print("End recording... ")
  input_data = np.expand_dims(input_data, axis=0)
  interpreter.set_tensor(input_details[0]['index'],
input_data)
  start_time = time.time()
  interpreter.invoke()
  stop_time = time.time()
```

Figure 4.12: Lite model inference on data and time benchmark (part2).

benchmark of the real-time prediction system. Once the model is converted into TF Lite version it is possible to transfer and run it into a Raspberry Pi. Installing the module via [122] it is possible to run it via the following code snippet, Figure 4.10.

The module interpreter loads the previously trained *tflote* model and for every 2 seconds of recorded audio it makes the inference via invoke() method, Figure 4.11. Here there are the first top 5 network predictions so if we want to use the mode in real-time USB microphone has to be connected to Raspberry Pi. Once device is connected we need to take 2 seconds of audio recording with a sample rate of 16KHz via sound device python library. Then we add a new dimension because the recorded audio has a shape $(32000, 1)$ whereas network input requires $(None, 32000, 1)$. The script continues with the same line of snippet in Figure 4.12.

4.2.2 Test Bed Scenario

In this paper we propose two different types of testbed scenario. The first one is based on a new training of a different CNN used than in [113].

The second scenario is based on the benchmark analysis of inference time using the same model but first employing the original model and the second time - the lite model. This experiment is done in order to establish performance quality for IoT device, applied in the real-time rainfall classification context.

It's implemented a temporal benchmark on the inference of the same architecture (Audio Classification 1D CNN) but with two different models. Starting from the Keras model trained as per scenario one, the latter has been converted into a TensorFlow Lite model and executed in the same machine in order to compute the results with the same computational resources (i7-9th gen, 32GB RAM, GeForce GTX 1650).

4.2.3 Performance Evaluation

In the first test bed scenario was trained the Audio Classification 1D CNN for 40 epochs. This training has led to a result that is better than that shown in the previous work in [113].

On the training set the performances turn out to be about 95% leading to an improvement of 2% compared to the previous CNN [113] performance which on a dateset with plastic timbre had an accuracy of about 93%.

Table 4.3 provides a comparison in terms of accuracy in the training and testing phase, the number of epochs and size of the dataset for the two networks.

As a result of the second scenario we noticed that the running time of an in-

Table 4.3: Performance comparison between two CNN networks.

Model Architecture used	Training Epochs	Best Train Accuracy	Best Test Accuracy	Test Set Size
CNN in [113]	40	96%	93%	506
New CNN in [121]	40	98%	95%	21274

ference for a Keras model is around 350 400ms for a single 2 seconds audio recording example. Running the same architecture with the same weights but with an optimized TensorFlow Lite model results in an inference time ranging from 50 100ms. This experiment shows a drastic improvement in performance with the same accuracy. This optimization also provides the possibility to have different neural networks running in an IoT device of medium computational power, such as Raspberry Pi 4.

Tests conducted on Raspberry Pi 4 show an inference time of around 100 120ms. As can be seen, there are no performance differences compared to running a device equipped with today's GPU, thus suggesting the possibility of using more complex networks or different approaches in these devices.

Table 4.4 shows the comparison between the inference times according to the type of platform and device used, as previously mentioned.

One of these possibilities is to merge two CNNs in a multimodal way for the classification of the same rain phenomenon but using audio and images to increase the robustness of the system.

Table 4.4: Performance comparison between the inference times.

Platform	Devices	Inferences Time
TensorFlow (Keras)	i7-9th gen, 32GB RAM, GeForce GTX 1650	350 ∼400ms 2s audio
TensorFlow Lite	i7-9th gen, 32GB RAM, GeForce GTX 1650	50 ∼100ms
TensorFlow (Keras)	RPI4B+: Quad core(ARM v8) 64-bit SoC@1.5GHz 4G LPDDR4-3200 SDRAM	400 ∼420 ms
TensorFlow Lite	RPI4B+: Quad core(ARM v8) 64-bit SoC@1.5GHz 4GB LPDDR4-3200 SDRAM	100 ∼120ms

4.3 Classification of Rainfall Estimation Using Video Signal

This sub-section presents studies conducted on rainfall intensity recognition using deep learning techniques and images extracted from video recordings of rain. In particular, sub-section 4.3.1 gives an overview of the state of the art related to techniques for rain and rain intensity recognition using video and/or images. In the subsections to follow, the study conducted during the PhD period on rain intensity recognition and classification through video signal (and images) and neural networks is explained step by step.

4.3.1 Related Work

This sub-section provides a general overview of new rainfall level estimation methods by analyzing videos recorded over the rain. The intensity of rain, in fact, can undergo tremendous variations even for short distances, less than one kilometer. It is therefore necessary to implement a prevention system especially in areas with high hydrological risk. To obtain measurements with large spatial resolution, it is necessary to distribute a substantial number of instruments such as rain gauges or weather stations, which may, nonetheless, have a major adverse impact in economic terms.

A number of studies on the classification of rainfall levels have been conducted by researchers employing various alternative methods and signals [108, 112, 113, 116, 118, 123–128]. These different approaches could represent a valid alternative to the typical solutions used in current meteorology. Furthermore, these innovative systems lead to improved performance in terms of time, flexibility and robustness. In particular, there are numerous papers in which the possibility of classifying the level of rainfall through the attenuation of the electromagnetic signal [108, 116, 123] and many others using the audio signal [108, 112, 113, 118] is studied. In this sub-section, will focus on the use of the video signal as a tool for classifying rainfall levels and therefore this section deal with previously conducted studies in this regard.

In [126] the Electronics and Telecommunications Department of the University of Mumbai developed an algorithm that allows determining the intensity of rainfall based on a single photo only. Data processing takes place implementing a technique denominated ERAUIP, using a high-resolution slow-motion camera, which acquires the image placed on a black background with a minimum speed of 1200 FPS. The camera acquires a color image which, once filtered by noise employing a clustering technique, is converted to gray scale. Subsequently, the image is divided into rows and columns, and therefore further divided into even and odd content. After which the conversion from the gray scale image to the binary image takes place and, finally, the size of the raindrops are measured thanks to the use of a morphological filter. This technique allows to reach high precision levels, even up to 90%. The performance will depend on the characteristics of the camera used; a full-bodied database of excellent quality images and videos is required in order to achieve better results.

A method for evaluating the raindrops present in images has been proposed in [127]. In this case image processing is performed using the 2D dual density Discrete Wavelet Transform, with the help of 2 cameras with advanced features, so that the 2 captured images have the same central point. This allows obtaining such information as the maximum and minimum size of the raindrop and the number of raindrops from the given image.

In the automotive sector, more and more sophisticated driver assistance systems are being developed to improve safety conditions behind the wheel, especially during adverse rain conditions. An approach used to detect raindrops present on a transparent screen has been developed in the context of research concerning the ITS intelligent transport algorithm, as reported in [128]. This technique detects the fuzzy raindrops that collect on the windshield, differentiating from the previously illustrated techniques as it is not necessary to adjust the focus on the windshield on which the raindrops fall.

Classification of rain levels through the acquisition of images can also be achieved by exploiting the learning abilities of a neural network [124]. In this case, a neural network was trained in order to obtain the detection of the glomerulus in the segments of the renal tissue present in digitized slides, and a classification considering such parameters as heterogeneity, staining, composition within the kidney composition etc. This is done by applying CNN in order to provide faster and more reliable diagnoses. Nevertheless, enormous amount of computational power is needed to train the network in an appropriate time frame.

The previously mentioned studies outline significant growth in interest concerning the replacement of the classic measurement methods with new instruments which can deliver better performance, guaranteeing a set of information that can cover a large area of the territory.

In [125] the authors propose a method for remote detection of rain via fixed cameras, using CNN. In particular, in the pre-processing phase the Sobel algorithm is used for detecting the edges and subsequently DCT is applied to the

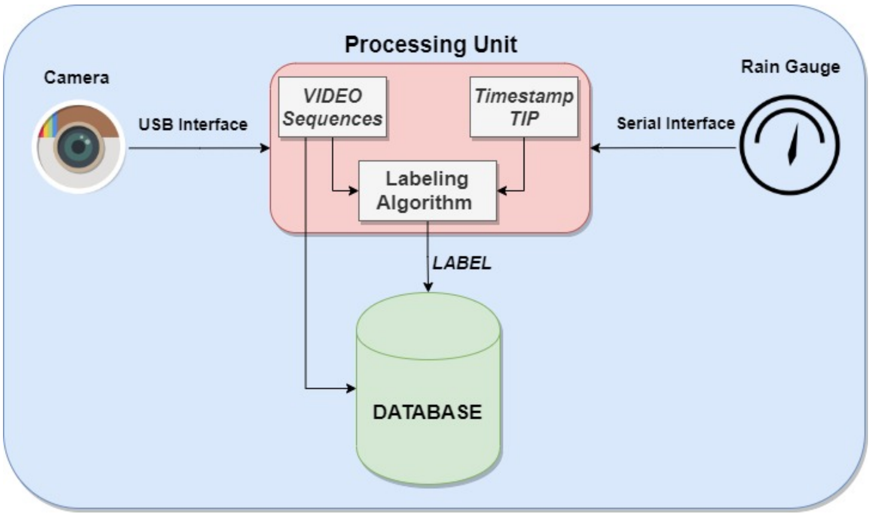


Figure 4.13: Block scheme of video acquisition system.

obtained images for further experiments in order to improve the classification scores. The classification results using the Sobel algorithm and the DCT are respectively 91.97% and 79.89%. There are only 2 classes considered: “Rain” and “No Rain”.

Unlike all the studies present in the state of the art, our method consists in detecting and classifying, as accurately as possible, seven different levels of rainfall intensity. The proposed technique, therefore, allows to set up a rainfall measurement system with the advantage of not having mechanical parts subject to breakdown, good accuracy and high temporal resolution.

4.3.2 Rainfall Video Dataset

The proposed system consists of a camera connected to a processing unit. The labeling algorithm (described in sub-section 4.1.3) is located in the processing unit. It is responsible for assigning the video sequences to the corresponding intensity classes thanks to the labeling performed by a traditional tipping rain gauge. After labeling, the video sequences are inserted into the database. Figure 4.13 shows the block scheme of the video acquisition system.

Video sequences are sampled with FPS equal to 30 frames of 640 × 480 pixels. Table 4.1 lists the intensity classes that make up the dataset. The database was created by recording natural rain characterized by all seven rainfall levels indicated in Table 4.1. Furthermore, during the rain recording phase, continuous checks were carried out on the tipping rain gauge used for labeling, so as to preserve it from dirt. An image processing phase, before inserting the input data to the neural network, was applied to the video recordings: extraction of the frames

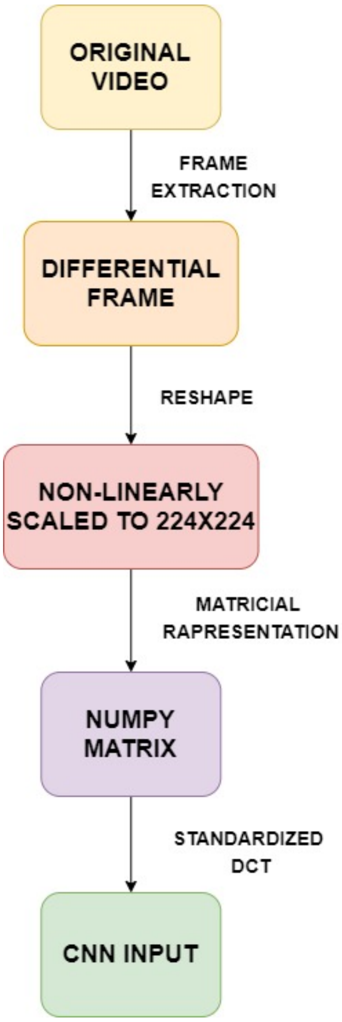


Figure 4.14: Image processing flowchart.

with frame-rate equal to 30 FPS and offset equal to 1 frame; subtraction of the resulting frames from each other. These frames will be referred to as “differential images”.

Differential images are subjected to DCT on 16 × 16 sub-blocks. The data obtained from the application of the DCT are standardized and input data to the neural network, described in Appendix D.

The tests were conducted using a camera inserted in a rigid plastic shaker with a transparent lid. The differential frames, relating to the various categories of precipitation, are used as input variables of a deep learning system based on a CNN classifier.

Figure 4.14 shows the general “Best practices” for inserting standardized matrices into the neural networks. The first block represents the original video, on which the extraction of the frames takes place at a frame-rate of 30 FPS. Then, the extracted frames are brought in grayscale and subsequently, differentiated by moving between one frame and the next with an offset of 1 frame.

The difference between two images is very simple to obtain, as we know that the images are nothing more than matrices containing numerical values in the RGB (or Black & White - BW) scale of size $M \times N$. In this regard, it is possible to obtain what we define “differential images” by applying a subtraction between two matrices of the same size. Therefore, since $F_1 = M \times N$ is the first extracted frame and $F_2 = M \times N$ the second extracted frame, the first differential image is given by (1).

From the matrix point of view, let f^1_{ij} be the elements of the matrix F_1, f^2_{ij} the elements of the matrix F_2 and f^d_{ij} the elements of the matrix F^d_1 with $i \in 1, \cdots, M$ and $j \in 1, \cdots, N$. f^d_{ij} is obtained by (4.2).

$$f^d_{ij} = f^2_{ij} - f^1_{ij} = \begin{pmatrix} f^2_{00} & \cdots \ldots & f^2_{oN} \\ \vdots & \ddots & \vdots \\ f^2_{M0} & \cdots & f^2_{MN} \end{pmatrix} - \begin{pmatrix} f^1_{00} & \cdots \ldots & f^1_{oN} \\ \vdots & \ddots & \vdots \\ f^1_{M0} & \cdots & f^1_{MN} \end{pmatrix} = \begin{pmatrix} f^d_{00} & \cdots \ldots & f^d_{oN} \\ \vdots & \ddots & \vdots \\ f^d_{M0} & \cdots & f^d_{MN} \end{pmatrix} \tag{4.2}$$

By generalizing it to all frames, the differential images will be obtained from (4.3).

$$F^d_k = F_k - F_{k-1}, \tag{4.3}$$

with k = 1, ... K, where K is the number of differential frames obtainable for each level of intensity and is related to the length of the recorded videos, the amount of FPS extracted from the video and the offset used.

In our method, in particular, we have adopted a function in Python language, called “abs_{diff}”, which makes up the difference between two images also reducing the noise that is obtained from a subtraction. In fact, a threshold is set at an arbitrary value between 0 and 255 (threshold 127 was used in this study); all BW pixels greater than 127 become 255; conversely, all BW pixels less than or equal to 127 become 0.

Once the differential images have been transformed into DCT matrices, a division is performed before inserting all the data within the neural network: 70% of these matrices are inserted in the training set and 30% in the test set. For each input, the probability percentage corresponding to each individual class will appear in the output.

The validation dataset was created as the dataset used in training and testing the neural network. In particular, the related videos are those that had not been included in the dataset for training the neural network.

The same procedures applied to the dataset used for training and testing the network are applied to the component frames of this dataset. The following chapter will analyze the performance of the network when this validation dataset is applied to the network input.

4.3.3 Proposed Method

Acquisition System Setup

The acquisition system is characterized by the following components, see Figure 4.15.

- Camera (a);
- Plastic shaker with a transparent cover (b);
- Tipping bucket rain gauge (c);
- Raspberry Pi (d);
- 4G dongle (e).

The webcam is connected via USB cable to the Raspberry Pi, where the processing phase of the collected data is performed. The operation of the labeling algorithm implemented within the Raspberry Pi, the connections between the

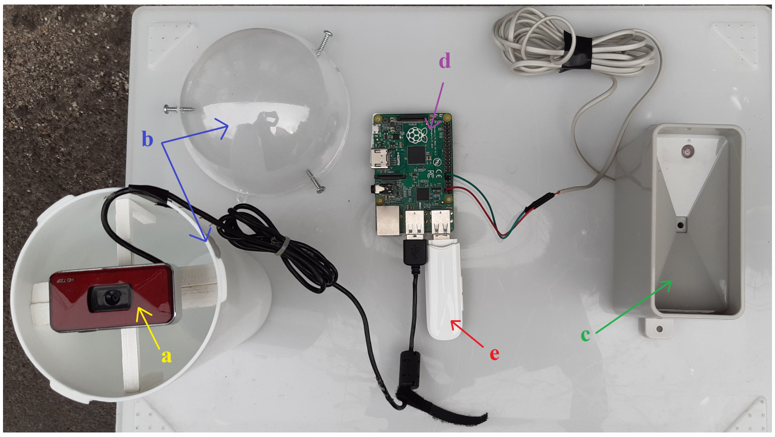


Figure 4.15: Hardware components of the video acquisition system.

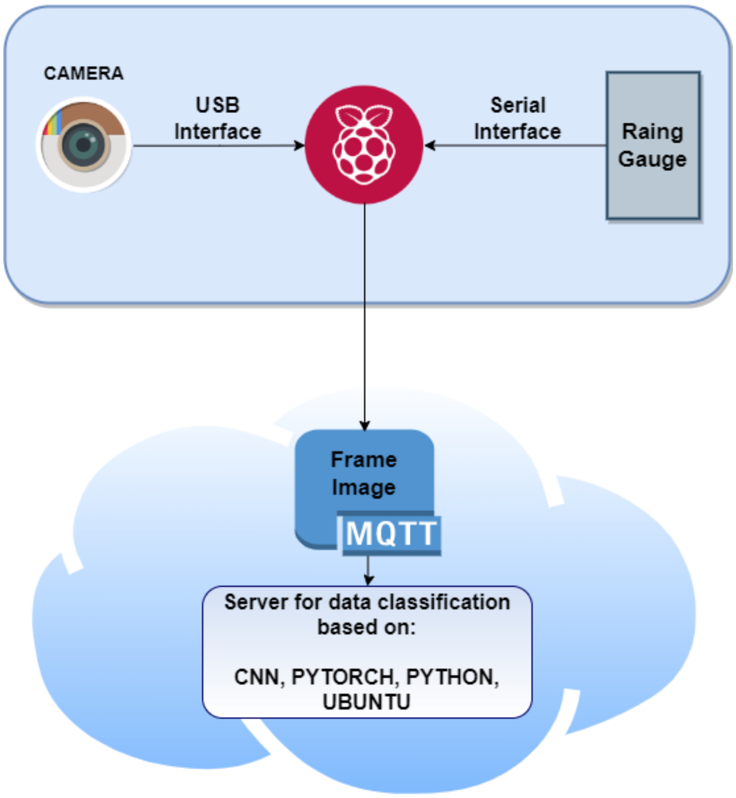

CAMERA
USB Interface
Serial Interface
Raing Gauge
Frame Image
MQTT
Server for data classification based on:
CNN, PYTORCH, PYTHON, UBUNTU

Figure 4.16: Conceptual scheme of the video acquisition system.

hardware devices in use and the categories of recording intensity are explained in detail in [113]. The 4G modem key allows real-time data transmission [129]. Figure 4.16 shows the conceptual scheme of the proposed system. In general, the processing unit implements the labeling algorithm that allows the generation of 30-second video sequences, divided by each class of rainfall intensity. The video sequences recorded and classified using the labeling algorithm are stored in the database and subsequently provided as input to the neural network for the training phase.

Analysis of Video Sequences

The present section deals with visual and statistical analysis of the obtained video sequences in order to further discriminate between different precipitation intensity levels. The video signal analysis tool, DiffImg, and the calculation tool, Matlab, are used to externally analyse the differential images extracted from the videos. In particular, DiffImg is a simple image comparison tool which takes two RGB images with the same size as the input. Some statistics are computed, and the positions at which pixels differ. With this tool, we are able to make a comparison between different images, highlighting the differences. By analyzing

Table 4.5: Standard Deviation for Each Precipitation Intensity

Rain Classification	Standard Deviation						AVG
nr	2.0	2.0	2.0	2.1	2.1	2.0	2.0
w	3.6	3.3	2.8	2.4	2.7	2.5	3.0
m	3.3	3.3	4.0	5.7	4.5	5.3	4.4
h	3.4	8.6	5.2	5.3	7.8	5.7	5.8
vh	6.2	5.1	4.1	8.4	6	5.4	6.3
s	4.0	8.3	9.6	10.0	8.2	9.3	8.1
c	17.3	7.4	10.7	15.4	8.5	11.4	11.9

the “differential images”, the authors can also view some interesting statistics, including:

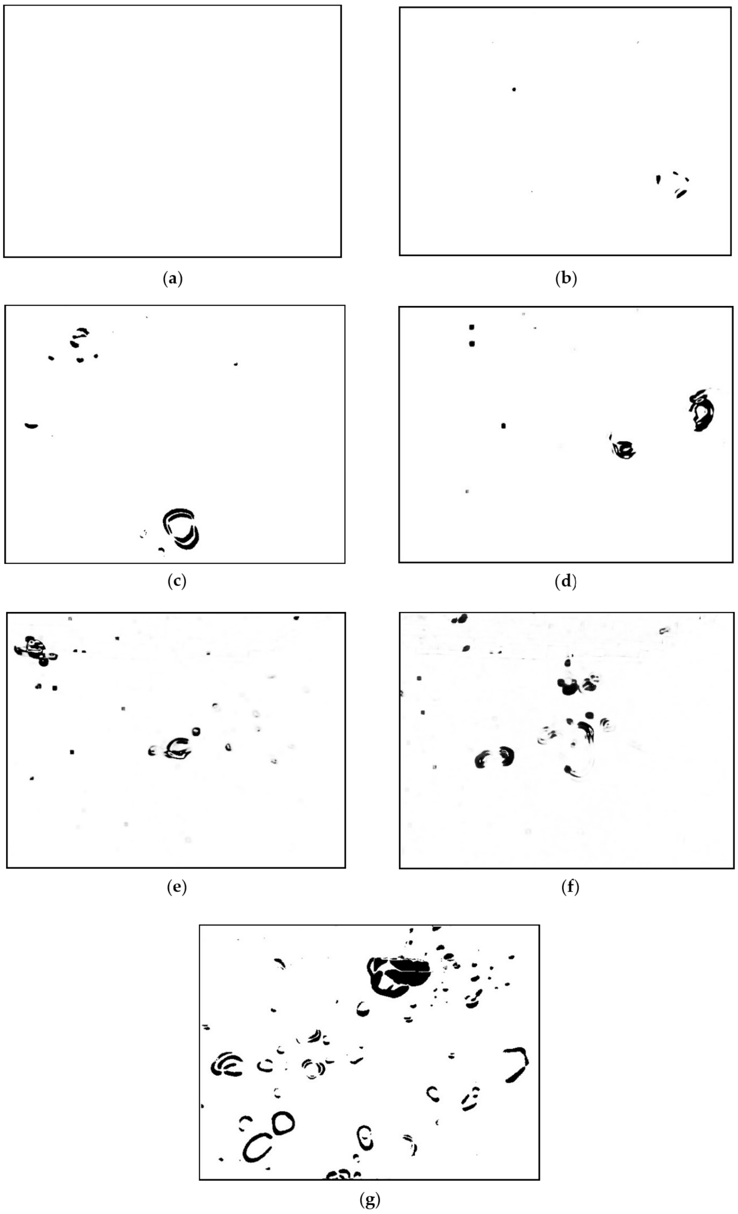
- Average error: indicates the difference between the two images in terms of RGB scale value;
- Standard deviation: indicates how much the second image varies compared to the first;
- Total of error pixels: indicates the total amount of different pixels between the first image and the second image (from which the “differential image” is obtained).

After a series of evaluations, it was observed that the most interesting feature to achieve our goal is the standard deviation. To better evaluate the values assumed by this parameter, it was decided to take into consideration 7 frames extracted from the videos, for each level of precipitation, with an interval equal to 1 second.

Subsequently, the value assumed by the standard deviation of the differential image obtained from the differences between consecutive frames was evaluated, so as to obtain a total of six “differential images”. By evaluating the trend of the parameter just mentioned, in particular the average of the standard deviation values, it was observed that this value increases with the increase in the rain level, as we can see in Table 4.5.

As can be seen from Table 4.5, there is a slight correlation between the standard deviation and the intensity of precipitation. This is immediately clear if you think that as the rate of rain increases, the number of raindrops and their size increases, so the difference between two frames belonging to higher levels will be more marked than the one between the images belonging to lower intensity levels.

To illustrate the concept discussed so far, a differential frame (taken from 9 “differential image”) for each rainfall level is represented in Figure 4.17. From



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 4.17: Examples of “Differential Image” for each level of intensity.

Figure 4.17 it is possible to note that there is a visible difference between the adjacent precipitation levels from the point of view of the raindrops represented in the frame. For example, it is clear that if in “No rain” class the image is white, in “Weak” the presence of a few raindrops is observed. The same concept may be applied to subsequent levels.

Despite some positive results obtained with the methodologies described above, some limits remain which do not allow the implementation of an autonomous system capable of achieving a reliable classification.

The different problems encountered during the experimental phase could be solved by creating a system based on machine learning by carrying out the training of a neural network through the use of more comprehensive and differentiated databases.

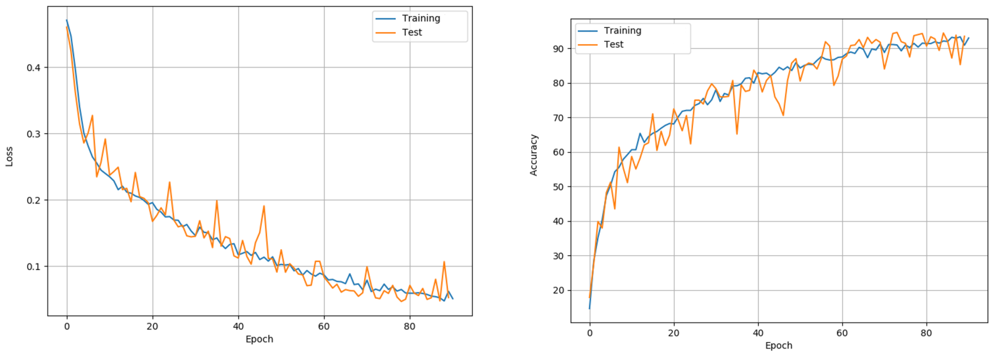
4.3.4 Performance Evaluation

This section will show the results of the training and validation of the convolutional neural network when the “differential images” are used as an input. The application of the DCT to 16 × 16 sub-blocks show an improvement in the classification performance. In fact, Figure 4.18a and Figure 4.18b respectively show the progress of training and test losses and training and test accuracy.

Figure 4.19 shows the confusion matrix. In this case, the percentage of accuracy of the average classification obtained by applying the DCT to 16 × 16 sub-blocks are approximately 49%, which can reach 75% if the adjacent miss-classifications are not considered.

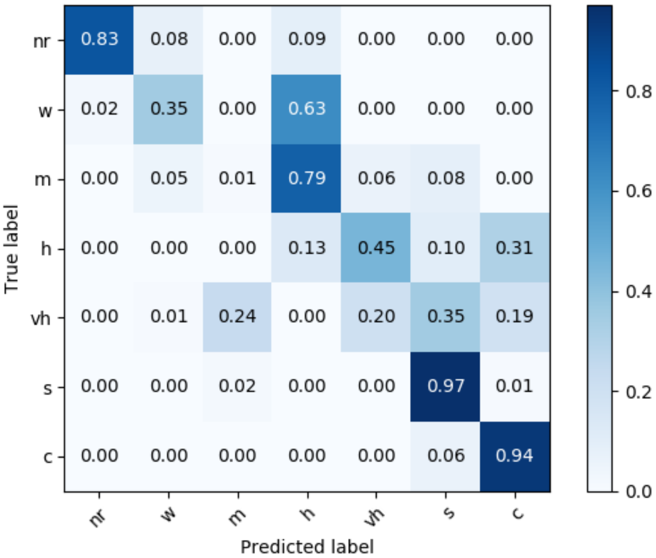
Once the performance has been defined, it is possible to compare our results with those present in the state of the art, in particular with the study proposed in [125]. The main differences from our study lie in the number of classes used, the methods of image pre-processing and the image capture mode. In fact, in [125] the Sobel algorithm is applied for the elimination of the edges; moreover, the images of the rain are captured in an orthogonal direction to the rain itself. From the performance point of view our study seems to deliver poorer performance, but it is important to note that our method is based on a classification using seven classes, that is, considering seven levels of intensity of precipitation which are fundamental in the context of monitoring the hydrogeological risks.

The use of seven classes to classify the intensity of precipitation and the high



(a) (b)

Figure 4.18: Training and test phase: (a) Loss function trend with Sub-block 16 trend. (b) Accuracy function trend with Sub-block 16 × 16.



	nr	w	m	h	vh	s	c
nr	0.83	0.08	0.00	0.09	0.00	0.00	0.00
w	0.02	0.35	0.00	0.63	0.00	0.00	0.00
m	0.00	0.05	0.01	0.79	0.06	0.08	0.00
h	0.00	0.00	0.00	0.13	0.45	0.10	0.31
vh	0.00	0.01	0.24	0.00	0.20	0.35	0.19
s	0.00	0.00	0.02	0.00	0.00	0.97	0.01
c	0.00	0.00	0.00	0.00	0.00	0.06	0.94

Figure 4.19: Confusion matrix with Sub-block 16 × 16.

temporal resolution of the level of intensity of precipitation allow for a more effective management of alerting mechanisms and therefore prevention and risk management in case of natural disasters-related hydrogeological risk.

4.4 Classification of Rainfall Estimation Using LTE Radio signal Parameter

Data regarding weather information are significantly increasing at a rapid pace. Identifying and obtaining valuable information from large amounts of weather data can be extremely beneficial in terms of agricultural development. Moreover, analytic of the weather data can help inform people in advance or alert them about possibly hazardous weather conditions (e.g. floods, extreme heat, droughts, and so on).

For this reason, an accurate estimate of rainfall levels is fundamental in smart city application scenarios.

The massive amount of data collected by low-cost sensors plus the recent data analysis technologies help us greatly improve the modern rainfall classification process. Big data analytics in cloud computing systems move from IoT to real-time control for smart cities.

The main existing rainfall level measurement methods employ tilt rain gauges, weather radars and satellites.

These traditional estimation techniques present a wide range of problems, for example:

- Tilt rain gauges tend to underestimate the amount of rainfall, particularly in snowfall and heavy rainfall events and they are also sensitive to the inclination of the receiver and different types of dirt that may clog the water collection point. Moreover, rain gauges only record local information, measuring the level of precipitation in the specific geographic location where the gauge is installed. Information requests for any other point must be obtained by interpolating the available data provided by nearby rain checks, with the consequence that this information may be influenced by a higher error;

- Weather radars have the advantage of being able to monitor a larger area, compared to the rain gauge, and to determine the real distribution of rainfall [103, 104], but they are very expensive;

- The satellite ensures greater spatial and temporal resolution, but the estimate itself is less accurate [105, 130].

For all these reasons, these systems are not very fast and accurate and are expensive to implement in smart cities. So, they cannot be used to estimate the intensity of rain in smart cities.

Aiming to implement rainfall estimate systems in smart cities, our idea is to use the already existing 4G/LTE network infrastructure. The innovative idea, proposed for the first time in this study, concerns the possibility to determine rainfall intensity based on the impact it has on the LTE radio channel parameters adopted for the cell selection mechanism. In particular, the study highlights the correlation between some parameters of the LTE system that measure the quality of the radio channel, i.e. the handover mechanism that selects the best base radio station for that particular UE. Compared to a previous study [112, 113], focused on the measurement of a single parameter related to the signal strength, in this study, the authors propose a new, larger set of parameters and use a wider class set of rain levels, including the “No rain” class and adding the “Shower” and “Cloudburst” classes. Therefore, the first part of the study involves analysing the impact of different rain level statistics on the main parameters adopted by the LTE radio-mobile system for the cell selection mechanism to hook to.

Finally, keeping in mind the recent spread of artificial intelligence and machine learning techniques applied in many contexts, the second part of the study is devoted to defining a pattern recognition technique based on the average and variance of parameters that characterize the quality of the LTE radio channel and an MLP neural network, leaving out the use of the latest and most advanced machine learning techniques for future work.

The studies is organized as follows: sub-section 4.4.1 briefly summarizes the main studies regarding the rainfall classification based on a radio link, followed by sub-section 4.4.3 that described the dataset created for this study; sub-section 4.4.4 described the proposed method to rainfall classification using LTE radio parameter and MLP network, including: rainfall classification based on radio signal quality parameters, testbed scenario and data analysis and results; sub-section 4.4.5 shows the results obtained using the MLP neural network; and, finally, there are the discussion and conclusions.

4.4.1 Related Work

Several past studies on this subject have only considered electromagnetic waves with frequencies greater than 10 GHz [131–133] since the impact of precipitation on the attenuation of electromagnetic waves, and therefore on the strength of the signal receiver, is best visible at high frequencies. A very interesting scenario, however, is represented by mobile radio systems that are widespread in the territory. Hence, in recent years, similar studies have been conducted considering frequencies used by cellular networks, therefore, less than 3 GHz, trying to analyze the effect of the impact of rain on the parameters that characterize the quality of the radio-mobile channel. In particular, in [134], the effect of rain on the $RSSI$, i.e. the intensity of the signal received by the user, was studied.

Network analysis on 2G links was performed in two different geographies for nine non-consecutive days. Measurements of the signal quality of each telephone terminal were transferred to a computer at the end of the day, along with measurements obtained by a rain gauge. The effect of rain on the intensity of the cellular signal was studied by analyzing variations in $RSSI$ values measured by the smartphone. Tests showed an $RSSI$ drop during rainfall in 8 out of 9 cases. In general, however, the decline in $RSSI$ did not lead to a clear and unambiguous distinction between the various levels of rainfall, since the decrease in power was insignificant. In [135], the measurements were taken for one year, using a transmitter/receiver system consisting of conventional antennas at the operating frequency of 2 GHz. The results showed reliable and accurate measurements for amounts of rain less than 1 mm for periods of 5 minutes. In our previous paper [107, 108], we proposed a study on the estimation of the rainfall level, based on the intensity of the received signal in LTE systems, evaluating the parameters of mean, variance and instant value of the $RSSI$ calculated in a sliding time window. These parameters were subsequently inserted into a Probabilistic Neural Network, which resulted in a satisfactory classification performance. The study deals with a first approach to using radio signal parameters for precipitation estimation, using frequencies from LTE technology (1.8 MHz/2.4 GHz). It is the first approach, as the distinction between rainfall levels (weak, moderate and strong) was not clearly evident and easily defined by the $RSSI$ values. Previous studies suggest that it is particularly difficult to classify rainfall levels only by taking into consideration signal strength received at frequencies below 3Ghz. For this reason, in this paper we focus on the study and analysis of additional radio parameters, other than the strength of the signal received, i.e. $RSSI$, defined by the LTE technology, which are able to provide the most accurate estimate of the rainfall level. In addition, we assess the effectiveness, in terms of classification, of all radio channel statistical parameters provided by the LTE technology, i.e. mean and standard deviation. Finally, the obtained data are fed as input to a MLP neural network [136] which differentiates between various levels of rainfall. The new rain gauge system, studied in this paper, offers great spatial resolution inasmuch as it is based on the impact of rain on radio routes between the UE and the base radio station. Base stations, in fact, have a fairly even and wide distribution in the territory and in cities. In smart cities of the future, characterized by intense use of the 5G radio route, this factor will be amplified as 5G base radio stations will have lower coverage radius and, therefore, much wider distribution in the territory than ever before.

This distribution allows to potentially convert each base radio station into a radio rain gauge. This technique comes with numerous advantages in terms of greater accuracy, speed and geographic accuracy, i.e. it will be possible to estimate the intensity of rain with high spatial precision.

At a functional level, even with the introduction of 5G technology, the proposed method will remain valid, as it will still be possible to use all the radio parameters to estimate the level of precipitation.

In perspective it is true that 5G uses higher frequencies offering a greater correlation between rainfall intensity levels and the $RSSI$ parameter, but it is also true that 5G provides operating modes even at frequencies below 10 GHz. For this reason, for a more correct and robust classification of rainfall levels, all radio channel monitoring parameters proposed in this study are required to be used also in 5G systems.

4.4.2 Cell Selection Criteria In LTE Technology

The smart city paradigm is a vision for future cities centred around the concept of connectivity. Indeed, connectivity is the core requirement for smart cities to exist, enabling tight integration among citizens, devices and service providers. However, it is also a means for interoperable access and interconnection among different services. Studies have been conducted on the use of the LTE infrastructure to implement different methodologies and scenarios that form smart cities [137–139].

This section describes the main features of the LTE technology with particular reference to the parameters that characterize the quality of the radio channel between the UE and the base station, as well as the handover mechanism.

In an LTE cellular network, when a UE moves between cells or can no longer have certain signal strength requirements from the cell it is connected up with, it must perform the selection/re-selection operation of a base station. For such an operation it is necessary to measure the strength and signal quality of the neighbouring cells. In LTE, the E-UTRAN Node B, also known as Evolved Node B (abbreviated as eNodeB or eNB), is the element in E-UTRA of LTE that is the evolution of the element Node B in UTRA of UMTS. In LTE, a UE measures the following two parameters of the reference signal, signalling them to the electronic node: RSRP and RSRQ.

From these two indexes, the eNB returns the Received Signal Strength Indicator (RSSI) parameter, which is the reference signal intensity indicator. The RSRP, which typically ranges between -44 dBm and -140 dBm, is a good measure of the power of a specific sector, excluding noise and interference from other sectors. When the UE is near an LTE station, the average RSRP values are around -75 dBm, and around -120 dBm when the UE is near the edge of the cellular coverage area [140].

After being turned on, the mobile device performs a low-level capture procedure to identify nearby LTE cells and find out how they are configured. The acquisition process is summarized in the following stages: The UE receives synchronization signals from all nearby cells. From the Primary Synchronization Signal (PSS), the UE reads the symbol timing and gathers information about the identity of the physical cell; from the Secondary Synchronization signal (SSS), the UE derives the frame timing, Physical Cell Identity (PCI), transmission mode (FDD or TDD), and duration of the cyclical prefix (normal or extended). At this point, the UE initiates the reception of cell-specific reference signals. These provide a reference of amplitude and phase for the channel estimation process, so they are essential for the following steps. The UE is then assigned the physical transmission channel and, subsequently, the UE reads the main block of information. The UE, after the first and second synchronization signals, receives the control format indicators.

Finally, the UE initiates the reception of the Physical Downlink Control Channel (PDCCH). This allows the UE to read the remaining blocks of system information (SIB), which are sent on the Physical Shared Downlink Channel (PDSCH). It captures all the remaining cell configuration data, such as the identities of the networks to which it belongs. The UE then initiates the reception of reference signals in the downlink channel [141]. Such signals are useful for the UE to:

- Provide an amplitude and a phase reference to be used in the estimation of channels;
- Measure the strength of the received signal according to the frequency;
- Calculate the channel quality indicators.

These procedures are carried out while the terminal is in the "IDLE" state, i.e. there is no active phone call or data transfer.

At this stage, the UE begins by performing the procedure of selecting the network and cell, which involves two main steps. Firstly, the UE selects a Public Land Mobile Network (PLMN) it will register with; secondly, it selects a cell that belongs to the selected network. Cell selection can be done in two ways. Usually, the UE has access to information stored on the potential frequencies and cells of the LTE service operator, starting from the last turning on or network selection procedure described above. If this information is not available, the device scans all supported LTE carrier frequencies and identifies the most powerful cell on each carrier on the selected network.

The selected cell is the one that meets several criteria, from release 9 onwards, standardized by 3GPP [142]. The most important criterion is show in (4.4):

$$S_{qual} > 0 \tag{4.4}$$

During the initial network selection, the UE calculates S_{qual} by (4.5):

$$S_{qual} = Q_{qualmeas} - Q_{qualmin} \tag{4.5}$$

In this equation, $Q_{qualmeas}$ [141] is the quality of the RSRQ measured, which indicates the signal-interference ratio plus the noise ratio of cell-specific reference signals. $Q_{qualmin}$ is the minimum value for the RSRQ, which the base station makes available in System Information Block 1 (SIB 1). This prevents a mobile phone from selecting a cell on a carrier frequency that is subject to high levels of interference.

For UEs in the “IDLE” state, cell re-selection management procedures have two main goals: maximizing UE battery life and minimizing signalling load on the network. From release 9 [143], a UE can also start taking measurements on neighbouring cells if the quality of the RSRQ falls below a threshold (4.6):

$$S_{qual} \leq S_{IntraSearchQ} \tag{4.6}$$

where, $S_{IntraSearchQ}$ is another threshold made public by the base station in SIB 3. S_{qual} depends on the RSRQ of the service cell and is calculated using (4.5). After finding and measuring the neighbouring cells, the UE calculates the classification scores of the service cell and one of its neighbours. The UE then moves on to the cell with the best score, provided that three conditions are met. Firstly, the UE must have been connected up to the service cell for at least a second. Secondly, the new cell must meet the criteria set out in “Cell Selection”. Finally, the new cell must be better classified than the service cell for a certain period between 0 and 7 s.

4.4.3 Rainfall LTE Radio Parameters Dataset

The database is made with *GMON* for each terminal and comprises data from LTE radio parameter measurements and CID. This data is sent at once as the tipping bucket rain gauge is activated. Tips generated by the tipping bucket rain gauge are sent to an IoT platform, using the publisher/subscriber protocol. The values taken from the tray are used to label the radio signal with different rain intensity. A labeling algorithm is applied to obtain the estimate in mm/h (described in detail in sub-section 4.1.3). Once the labelling algorithm is executed, the rainfall values, in mm/h, are compared and synchronized in time with the data in the CSV file created by *GMON*. This process enables data labelling in the CSV file. In fact, this allows obtaining files where the network parameters are related to each classification level. The database consists of five categories of precipitation intensity, defined in Table 4.6; for each category there is a CSV file containing network parameters: $RSRP$, $RSRQ$, $RSSI$ and SNR; date and time from when the data was recorded; CID and LAC. The current database was created by recording network parameters on different days and in different areas of the territory during the precipitation (in various rain intensity conditions,

Table 4.6: Rainfall classification using LTE radio parameter: Classification and rainfall intensity range

Classification	Acronym	Rainfall Intensity [mm/h]
No rain	nr	<0.5
Moderate	m	[0.5 ÷ 6]
Heavy	h	[6 ÷ 10]
Shower	s	[10 ÷ 30]
Cloudburst	c	>30

including the "No rain" case). This implies a certain robustness of the system, as the results obtained take into account the use of base stations in different locations.

As already mentioned, data collected by *GMON* are stored, every second, in a *CSV* file and this file is composed of records containing the data related to the radio parameters indicated above. In particular, the number of records for each level of rainfall is as follows:

- No rain: 124.210 records;
- Moderate rain: 378.584 records;
- Heavy rain: 18.667 records;
- Shower: 15.595 records;
- Cloudburst: 5.255 records

Initially, the first and second-order statistical parameters of the radio channel quality parameters described in sub-section 4.4.4 were calculated.

The statistical analysis was conducted by considering a 180-second sliding window with 15 second off-sets. The first order statistical parameters, represented by the averages of the values, are as follows:

- RXL_{AVG};
- $RSRP_{AVG}$;
- $RSRQ_{AVG}$;
- SNR_{AVG}.

The statistical parameters of the second order, represented by the variations, are as follows:

- RXL_{VAR};
- $RSRP_{VAR}$;
- $RSRQ_{VAR}$;
- SNR_{VAR}

The number of databases for each class is as follows:

- No rain: 686 records;
- Moderate rain: 2089 records;
- Heavy rain: 103 records;
- Shower: 86 records;
- Cloudburst: 29 records.

Due to the non-uniformity of the data in the dataset for each class, it was decided to reduce the number of samples in the "No rain", "Moderate rain", "Heavy rain" and "Shower" classes to 50 records each and to leave the "Cloudburst" class unchanged.

The three cell change parameters were also analysed: CID_{HOPS}, CID_{CSR} and $CID_{PERMANENCE}$. Subsequently, the degrees of linear separation were assessed using the *FDR*, which allows measuring the degree of linear separation that the given parameter has [144]. Finally, a MLP (described in Appendix E) was applied, providing the parameters used for statistical analysis as an input, and the 5 classes of rainfall levels as an output. After the training phase the neural network ranks the contribution made by each parameter considering that the best match is the one based on non-linear techniques. Finally, by analysing the confusion matrix, the accuracy of the system is determined.

4.4.4 Proposed Method

Rainfall Classification Based on Radio Signal Quality Parameters

The new method for estimating rainfall levels, proposed in this paper, is based on a nonlinear matching pattern recognition approach. As in previous studies carried out in audio biometrics [145, 146], once the set of parameters that characterizes the radio channel is extracted, they are analyzed on time windows by measuring the statistical parameters of the first and second order. The parameter set is sent to a nonlinear matching block based on MLP. In particular, the radio parameters used for analyzing and creating the dataset in input to the neural network were measured in terms of average and variance, via the app *GMON* for each level of rainfall. The following radio parameters were therefore obtained: $RSRP_{AVG}$, the average power of the $RSRP$ signal received from the UE;

- $RSRQ_{AVG}$, the average signal power quality received from the UE;
- RXL_{AVG}, the average instant power of the RXL signal received from the UE;
- SNR_{AVG}, the average signal-to-noise ratio received from the UE;
- $RSRP_{VAR}$ indicates variations in signal strength of the $RSRP$ signal received from the UE between two consecutive measurements;
- $RSRQ_{VAR}$ indicates variations in received signal strength quality from the UE between two consecutive measurements;
- RXL_{VAR} indicates variations in instant power of the RXL signal received from the UE between two consecutive measurements;
- SNR_{VAR} indicates variations in the signal-to-noise ratio received from the UE between two consecutive measurements.

In addition, with regard to the CID, i.e. the identification number of the LTE cell the terminal is connected to, several parameters have been calculated:

- The "reference CID", CID_{CSR}, the CID for which the terminal remains connected to for longer;
- The CID_{HOPS}, the number of connections to other cells the UE makes;
- The $CID_{PERMANENCE}$ that indicates how long the UE remains connected to the reference cell;
- The CID_{CSR} indicates the number of times the UE disconnects from the reference cell, and then returns to it.

These parameters are used for statistical analysis and as input to an MLP neural network.

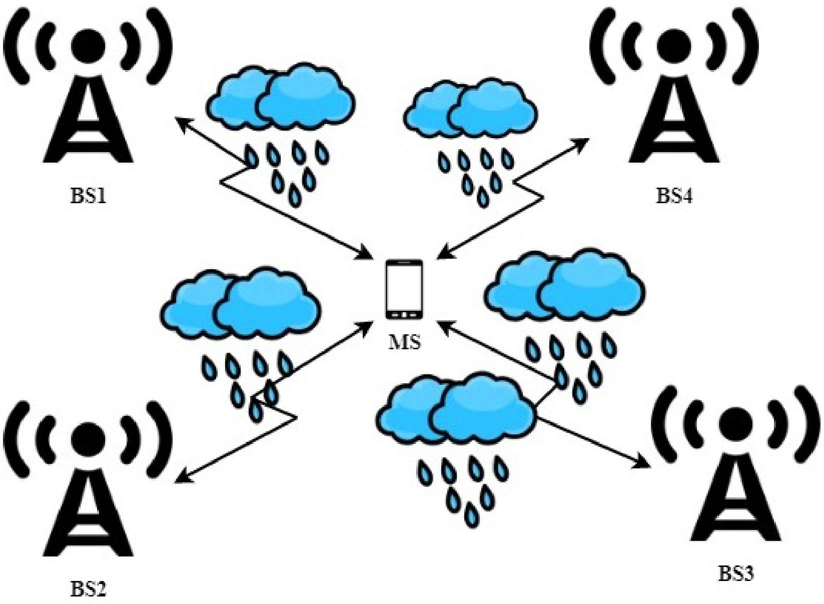


Figure 4.20: Rainfall classification using LTE radio parameter: Testbed scenario.

The Testbed Scenario

In order to assess the impact of rain on the main parameters that characterize cell selection phases, the scenario taken into account is that of an LTE UE in the “IDLE” state, equipped with an application (e.g. *GMON*) for the measurement of the parameters used by the cell selection mechanisms (Figure 4.20). Nearby, a classic tipping bucket rain gauge records rainfall levels in mm/h using a processing, labelling and synchronizing board with radio-mobile channel data measured by *GMON*. The UE, as described in the previous section, detects the power level of the base radio stations nearby and hooks to the station with the greatest power.

In cases where atmospheric precipitation occurs, in an “IDLE” state the UE may be affected by small changes in the radio signal. The UE, in fact, receiving a lower power signal from the base station to which it is connected, could pick up a higher power signal from another base station nearby. The impact of the rain can, thus, lead to greater frequency of the re-selection phases of the base station to which the device is connected. This data can have a major impact on the classification of rainfall levels as the higher the level of rainfall the higher the selection and re-selection procedures of one or more neighbouring cells may be. The parameter used to describe this procedure is the CID, which distinguishes, as already seen, the cell to which the UE is connected, the number of hops, i.e. jumps that it makes from one reference cell to another, and the timespan of continuous connection to a given reference cell; which is the one to which it is statistically connected to the longest. Following, in the next subsections, we will describe the testbed and the database used to train the MLP neural network.

This section will describe $RSRQ$, RXL, SNR, $RSRP$, and CID (LTE defines it as PCI) data collection procedure. The data employed for the creation of the radio database, on which the analyses and tests were carried out, were collected through an ad hoc implemented acquisition system consisting of: a tilting tub, a processing unit and a 4G smartphone SIM inside a shaker. A dedicated application, called *GMON*, is installed on the smartphone, able to export a full report for different network and signal parameters such as $RSSI$, $RSRP$ and $RSRQ$ (illustrated in sub-section 4.4.2) in CSV format.

As for the tipping bucket rain gauge, it acts as a rain indicator. It includes a rain-gathering funnel, two triangular tubs mounted on a fulcrum and an electronic switch.

The rain is channelled through the funnel to one of the trays. When the tub is full, it loses balance and flips over, emptying into the outer shell of the meter, while the other tub is lifted into place for later reading. The rain gauge is connected to the processing unit via an *RJ11* cable and is managed ad hoc through a software interface that can detect and count the "interruptions" generated by the rain gauge tray whenever a tilt occurs.

To process the proposed classification approach, a database was created by recording the collected data. The database includes measurements of the parameters described above in five different weather conditions: "No rain", "Moderate rain", "Heavy rain", "Shower" and "Cloudburst". The database framework is displayed in Table 4.6.

Data Analysis and Results

First, mean and variance of radio parameters are studied to determine if there is a link between the values obtained from the statistical analysis of these parameters and the rainfall levels defined in Table 4.6. The statistics of the individual parameter with the representation of mean, minimum and maximum, standard deviation and typical distribution of values around the mean were represented for each class of rainfall.

Figure 4.21, 4.22, and 4.23 show that there are no parameters clearly distinguishing between classes, although some classes have low-overlapping value distributions with other classes (e.g. $RSRP_{AVG}$ and RXL_{AVG} have "No Rain" class values that overlap with those of the other classes). The same goes for the $RSRQ_{AVG}$ parameter that ensures a good separation between the "Shower" and "Cloudburst"

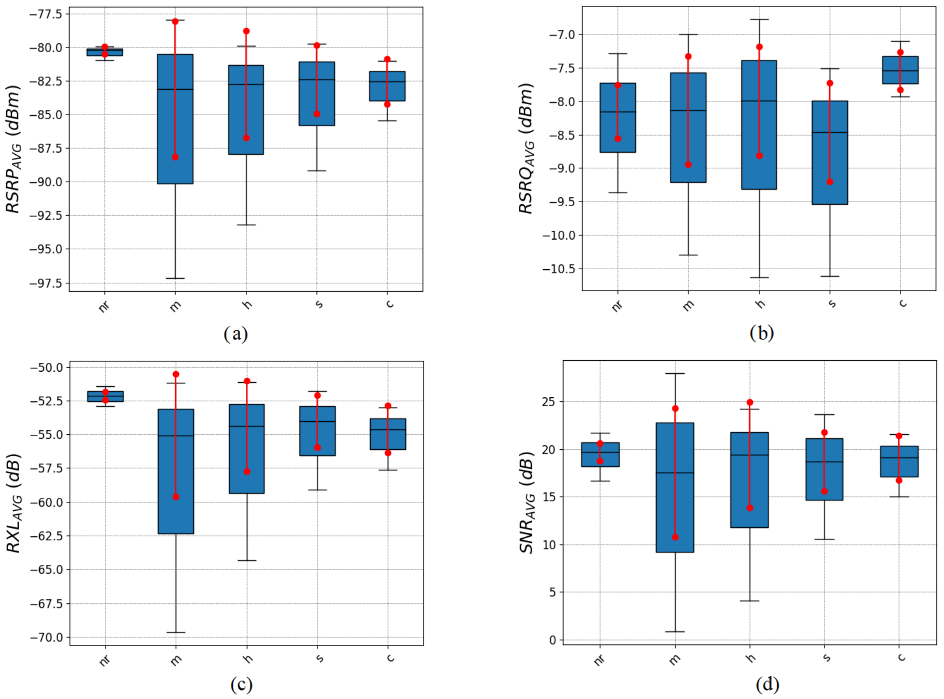


Figure 4.21: Representation of radio parameter averages statistics: (a) average $RSRP$, (b) average $RSRQ$, (c) average RXL, (d) average SNR.

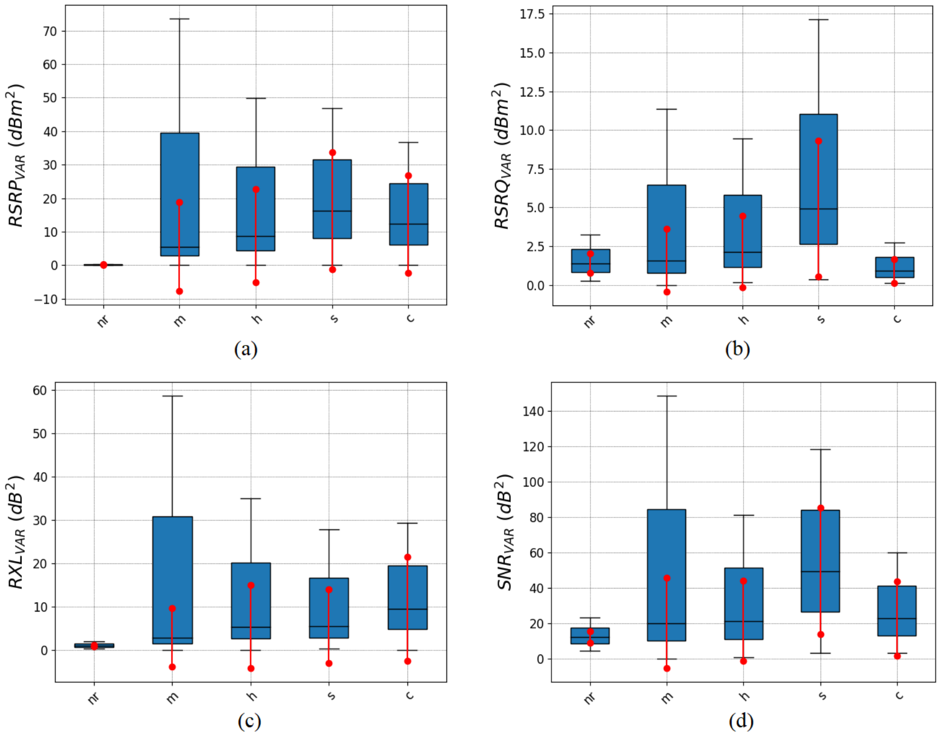


Figure 4.22: Representation of radio parameter variance statistics: (a) $RSRP$ variance, (b) $RSRQ$ variance, (c) RXL variance, (d) SNR variance.

classes. The same case applies to the variance of radio parameters (Figure 4.22).

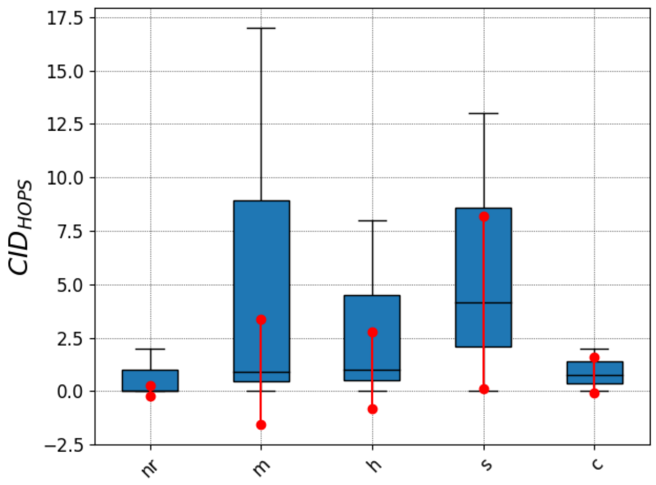


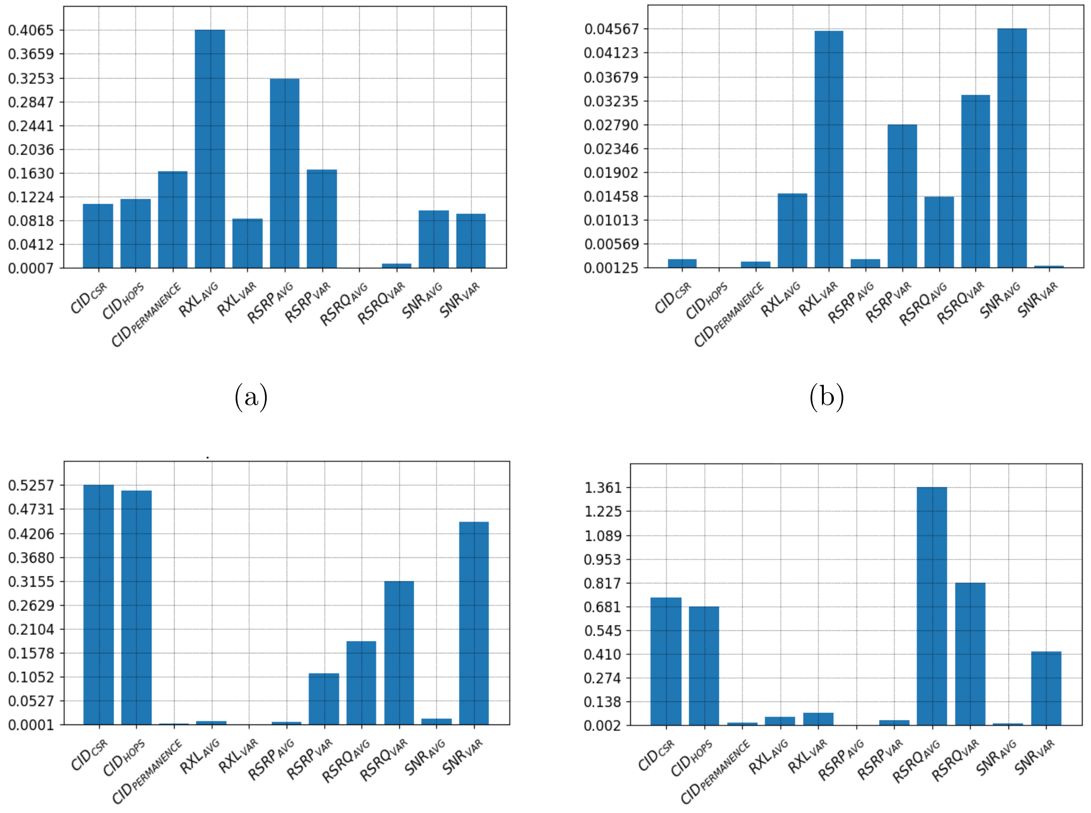
Figure 4.23: Representation of CID_{HOPS} statistics.

Figure 4.23 shows the average jump that is made from one cell to another at each level of rainfall. In this case, there is a clear distinction between "s" and "c" classes. In general, it is noted that the data appears to be jagged and does not give a clear distinction between precipitation levels according to linear analysis criteria. A simple first and second level statistical analysis is therefore not enough to establish clear classification of rainfall levels. For this reason, an analysis of the FDR applied to radio parameters, presents the results of classification techniques suitable for those cases with non-linear separation in order to define the parameter that helps obtain better discrimination.

FDR value between two adjacent classes, relative to each parameter described in the previous section, will be evaluated. As there are five levels of rainfall considered in this study, there will be five classes and, therefore, four FDR values, for each radio parameter considered, for the classes: $nr-m$, $m-h$, $h-s$, $s-c$. Figure 4.24 shows the total FDR values of all parameters for each pair of adjacent classes. The parameters that have the highest degree of linear separation are:

- the RXL_{AVG} and the $RSRP_{AVG}$ for classes nr to m (Figure 4.24a);
- The SNR_{AVG} and RXL_{VAR} for classes m through h (Figure 4.24b);
- The CID_{CSR} and CID_{HOPS} for classes h to s (Figure 4.24c);
- $RSRQ_{AVG}$ and $RSRQ_{VAR}$ for classes s through c (Figure 4.24d).

By analysing and calculating the degree of total linear separation (see Figure 4.25), i.e. applicable to all classes, we obtain the parameters with the highest linear separation index: the $RSRQ_{AVG}$, the CID_{CSR} and the CID_{HOPS}. Starting



(a) (b)

(c) (d)

Figure 4.24: Parameter FDR values between adjacent classes: (a) FDR between $nr - m$, (b) FDR between $m - h$, (c) FDR between $h - s$, (d) FDR between $s - c$.

from these values, it was decided to carry out the Principal Component Analysis (PCA) [36] of these parameters. PCA is a technique aimed at deriving a smaller set of "artificial" orthogonal variables starting from a set of correlated numerical variables. The reduced set of linear orthogonal projections (known as "principal components", "PC") is obtained by appropriately combining the original variables linearly.

Figure 4.26 outlines the results obtained by the PCA. As evidenced, individual parameters contribute differently to the separability of the individual classes. To find a solution to this problem, we implemented a MLP with multiple classes, where the above parameters are used as input, while the output will comprise the accuracy of the classification and the determination of parameters which are more important than others to neural network training.

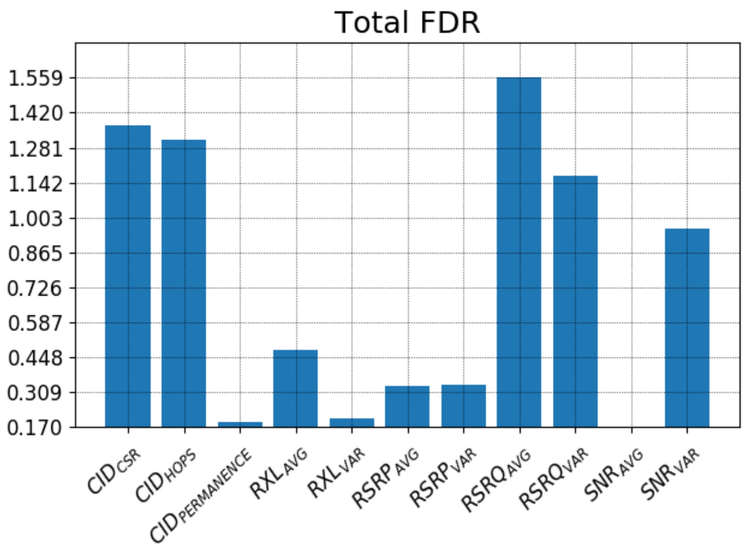


Figure 4.25: Total *FDR*.

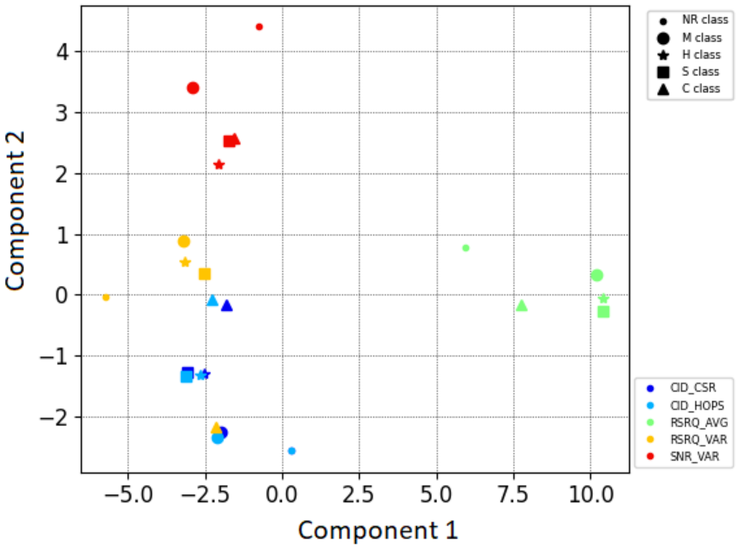


Figure 4.26: *PCA* analysis.

4.4.5 Performance Evaluation

Once the statistical analysis was completed and the *FDR* was calculated, as noticeable from Figure 4.26, the separation of classes is not linear. For this reason, the Perceptron neural network was applied to the dataset. The dataset used for training and testing the Perceptron network has been described in Appendix D.

In particular, from the *CSV* files, containing the recordings of radio parameters made by *GMON*, relating to each precipitation level, the statistical parameters were extracted within a window of 180 seconds (data is recorded every second) and a 15 second offset. The dataset is therefore divided as follows:

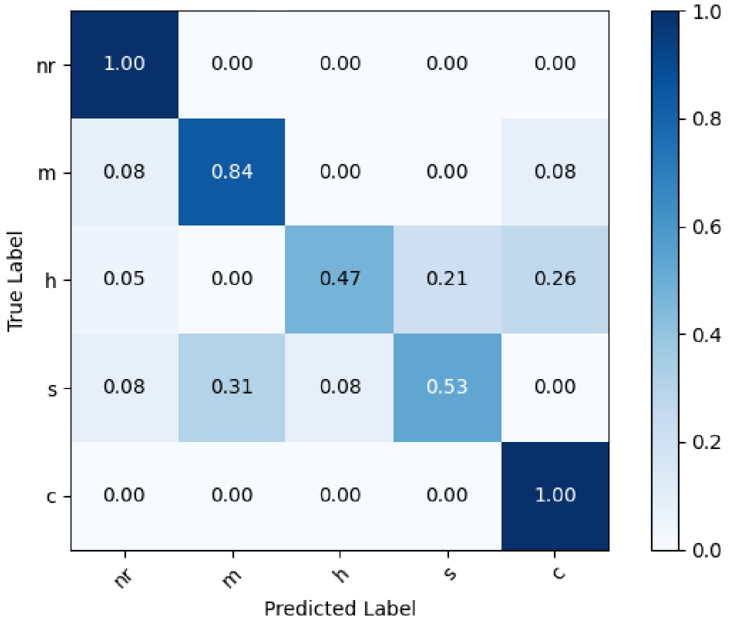


Figure 4.27: Rainfall classification using LTE radio parameter: Confusion matrix.

- Fifty data samples for the “No rain” class;
- Fifty data samples for the “Moderate” class;
- Fifty data samples for the “Heavy” class;
- Fifty data samples for the “Shower” class;
- Twenty-nine data samples for the “Cloudburst” class.

The training data makes up 70% of the dataset, while the remaining 30% of the dataset is used to the testing phase, once the neural network was trained. The result of this phase is shown in Figure 4.27 with the confusion matrix. The statistical classification functions (described in Appendix E) are applied to evaluate the proposed method, based on the results obtained from the confusion matrix.

The results obtained in terms of statistical parameters are shown in Table 4.7. Table 4.7 suggests good performance of Accuracy (96.0%), Sensitivity/Recall (96.0%), Fall – out (13.32%), precision (91.4%) and F_1Score (93.6%).

Once these accuracy metrics are implemented and calculated, it is possible to use an additional model validation tool, namely the one based on the ROC and AUC ROC graphs to validate the model.

A perfect classifier would be located in the upper left corner of the graph, with a true positive rate equal to 1 and a false positive rate equal to 0. Based on the ROC curve, we can therefore calculate the area under the curve, AUC, to characterize the performance of the classification model.

Table 4.7: The overall values of accuracy, TPR, FPR, PRE and $F_1 Score$ of rainfall classification using LTE radio parameter.

Class	TPR [%]	FPR [%]	PRE [%]	$F_1 Score$ [%]
nr	100	0	100	100
m	100	3.8	87	93
h	95	0	95	95
s	85	0	85	85
c	100	2.8	90	95
Total	96	13.32	91.4	93.6

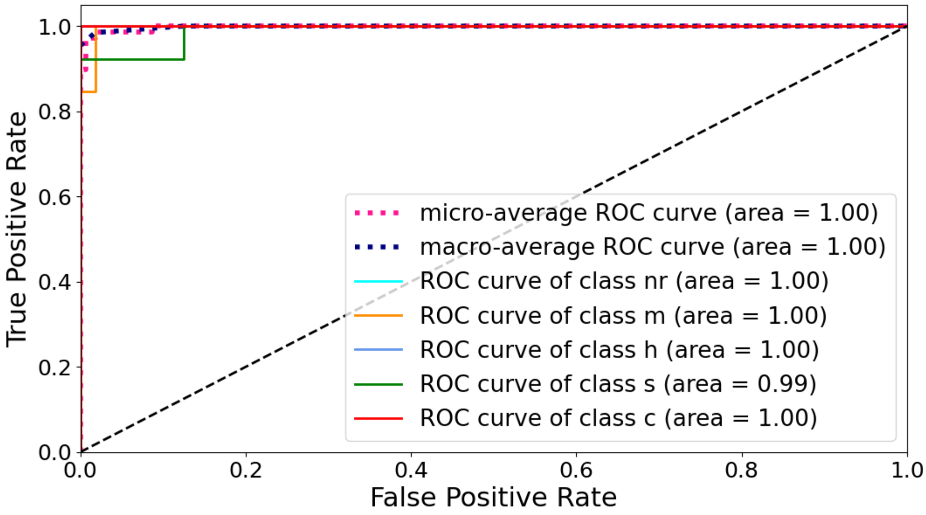


Figure 4.28: Rainfall classification using LTE radio parameter: ROC curve.

Applying this concept to our classification method, in Figure 4.28 we observe the resulting ROC curve which indicates that a certain degree of variance between the various parts and the average $ROC\ AUC$ lies between the perfect score (1.0) and the diagonal (0.5).

Research on the performance of a classifier such as $ROC\ AUC$ may provide additional information on its performance compared to unbalanced samples. In [147] Bradley demonstrated that $ROC\ AUC$ and accuracy metrics generally agree with each other.

The micro-average is calculated from the individual true positives, true negatives, false positives and false negatives of the system. In our case the micro-average of the precision score in a 5 (nr, m, h, s and c) class system can be calculated by (4.7):

$$PRE_{micro} = \frac{TP_{nr} + TP_m + TP_h + TP_s + TP_c}{TP_{nr} + \cdots + TP_c + FP_{nr} + \cdots + FP_c} \tag{4.7}$$

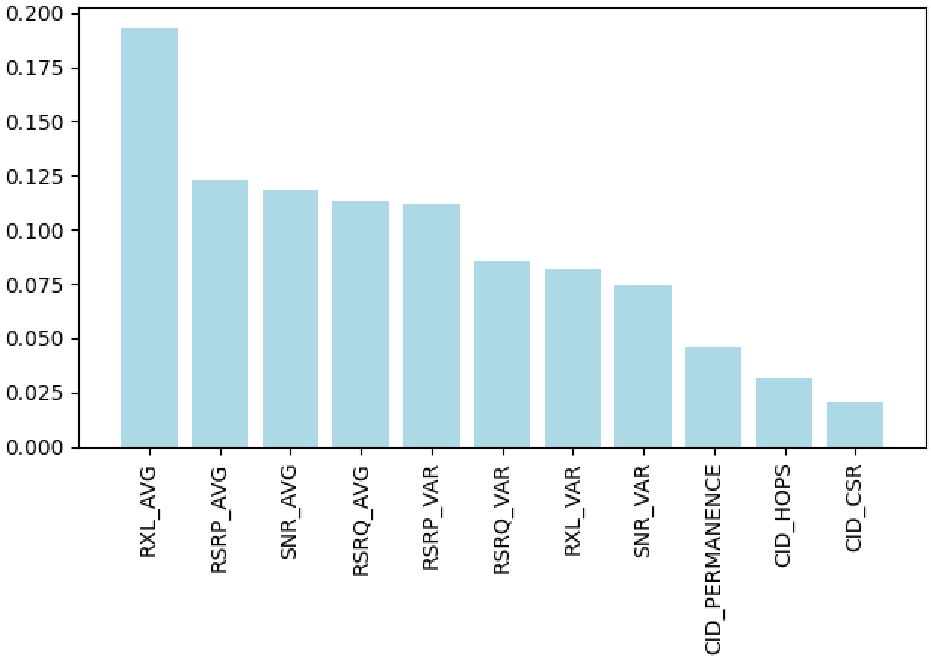


Figure 4.29: Rainfall classification using LTE radio parameter: Level of importance features.

The macro-average is simply calculated as the average scores of the different systems ((4.8)):

$$PRE_{macro} = \frac{PRE_{nr} + PRE_{m} + PRE_{h} + PRE_{s} + PRE_{c}}{5} \tag{4.8}$$

Micro-media is useful if we want to weigh each instance or forecast, while macro-media weighs all classes equally to evaluate the overall performance of a classifier compared to the labels given to the most frequent classes. The graph shows that the area under the ROC curve is very large. This means that our model has excellent performance. In general, tipping bucket rain gauge labelling does not allow establishing typical variations in rain intensity. By means of the causal forest technique, it is possible to gather useful information concerning the importance that the network assigns to various input parameters. The graph in Figure 4.29 shows that the five features in the order of importance for the neural network are: RXL_{AVG}, $RSRP_{AVG}$, SNR_{AVG}, $RSRQ_{AVG}$ and $RSRP_{VAR}$.

4.4.6 Discussion and Conclusion

In this section the obtained results are discussed and compared with recent studies in the state of the art.

Our five-class classifier (nr, m, h, s and c) has managed to obtain classification

Table 4.8: Rainfall classification using LTE radio parameter: Comparison of rainfall classification methods.

Ref.	Radio Technogies	Features	Methods	F_1 Score [%]	Accuracy [%]						
					No rain	Rain	Rain levels				
							w	m	h	s	c
[148]	GSM	RSL	SVM	n.d.	82.84	84.79	75.78	n.d.	88.26	n.d.	n.d.
[131]	Commercial Microwave link	Signal attenuation	KFD	n.d.	96.74	80.18	26	n.d.	n.d.	n.d.	n.d.
[108]	4G/LTE	RSL (value, mean and variance)	PNN	n.d.	100	n.d.	90	96.7	100	n.d.	n.d.
[116]	4G/LTE	RSRP, RSRQ, RXL, SNR (mean and variance), CID	Multiclass MLP	93.6	100	95	n.d.	100	95	85	100

accuracy of 96.0% and an $F_1 Score$ of 93.6%. Table 4.8 shows the comparison in terms of radio technology in use, features, classification methods, classification levels and achieved performance.

Based on research carried out on the current state of knowledge, the study referenced in [108] and the one described in [148] are the only studies dealing with the classification of precipitation levels by radio signals at frequencies below 10 GHz. In [148] GSM technology is used, and the RSL (Received Signal Strength) parameter is considered as a feature. As for the classifier, a 3-class (nr, w, h) SVM (Support Vector Machine) is used. Compared to the study [108], it can be seen that overall it has lower performance, since in [108] we consider 4 rainfall levels (nr, w, m, h).

In [131] commercial microwave link is used to analyze the radio signal attenuation caused by rain at frequencies about 20 Ghz. In this study the authors adopted the KFD (Kernel Fisher Discriminant) method for intensity classification into only 3 classes: dry (no rain), rain and sleet (i.e. melted snow/freezing rain which, for simplicity, has been considered as weak rain).

It should be emphasized that the comparison proposed in Table 4.8 presents a congruity of classes only for the two No rain/Rain columns, from which there is already a clear performance improvement in the proposed method compared to the existing solutions. As indicated, the classes of rain levels vary in number and therefore it is not possible to make a direct comparison between the various methods. It should also be noted that the datasets used are different.

However, comparing this study with the previous one [108], it can be said that in this study several classification levels are proposed, adding the highest intensity classes (Shower and Cloudburst) and using more parameters of 4G radio technology to strengthen the system in the previous study [108] which exploits only the mean and the variance of the RSL parameter. In fact, it is well known that the radio signal strength may undergo variations depending on the conditions of signal propagation and, therefore, on the surrounding environment. With regard to the matching technique, this study exploits a very simple neural network with low computational complexity, compared to the study proposed in [108], achieving, however, overall good performances.

At high frequencies the impact of rain on attenuation is very big, resulting in greater performance in terms of classification accuracy. In any case, for further performance optimization, it is always useful to add the set of parameters proposed in this study in addition to that of simple signal strength.

In conclusion the study proposes an innovative approach to rainfall classification for smart city applications. The main idea is based on the impact that rain has on a set of parameters that characterize the radio-mobile channel quality. The paper highlights the link between rainfall levels and the trend of parameters adopted for the cell selection phase in the LTE mobile network. In particular, the system requires only the extraction of the parameters that a UE measure and the subsequent comparison with a nonlinear matching system based on MLP networks. The performance is very good in terms of accuracy and spatial resolution. Taking into account the typical micro-variances of rainfall intensity, it is possible to consider an average accuracy of 96%. The new rain gauge exceeds the limits of the traditional ones, as it has no mechanical parts and requires no maintenance.

4.5 Radio Packet Error Compensation Using Radio Mobile Network Quality Parameters in a Rainfall Scenario

The last few years have been characterised by several applications devised with the scope of creating smarter environment and systems. The key enabler of these smart city applications has been the fusion of IoT or 5G based systems and AI.

In these scenarios, the data analysed via Deep Learning (DL) techniques and exchanged between the sensors and the network assume a primary role for the feasibility of smart systems.

The data passing through the 4G/5G communication channels must be supported by a robust link that withstands external disturbances in order to enable high QoS. New concerns are emerging in this field, such as bandwidth speed, reliability of data communication, robustness of data connections and throughput. Several studies have addressed these issues. In [149], the authors propose a classification system for online multimedia traffic based on a CNN network that will meet the demands of the networks of the future. Techniques of distributed user-to-multiple AP association methods are proposed in [150], where the objective is to maximize the long-term sum-rate subject to application QoS constraints, as well as to AP load constraints. In [151], a method of classifying data flows through the SDN is investigated in order to guarantee the QoS requirements of different services. A series of machine learning algorithms are combined and a data flow classification method called MACCA2-RF&RF is proposed in order to identify the data flow category and obtain the QoS requirements.

In fact, although the rapid and continuous consolidation of the demand for mobile Internet access together with the significant increase of mobile services provided by the incoming fifth generation networks (5G) have created the conditions for a significant expansion of mobile IP applications and services, radio network equipment is often unable to support real-time and time-critical traffic due to various disturbances affecting the wireless channel. There can be various causes for the disruption of the wireless medium, including the adverse effect of rain on radio channel quality [152, 153].

In this section, the rainfall level estimation algorithm, described in sub-section 4.4, is used to introduce a new Smart VPN bonding algorithm [154] that can occasionally compensate the PER based on the statistical parameters of the mobile radio signals and changes in the CID when the radio channel is affected by rain.

The key steps of the testbed scenario are summarized as follows:

1. Implementation of a video surveillance system equipped with a dual SIM modem in road safety contexts;

2. Through the 4G mobile network (LTE) the SIMs connect to the various eNBs from which the parameters of the mobile radio signal can be recorded. The recorded radio parameters are extracted, processed and fed as input to a CNN neural network that allows to estimate the rainfall intensity In particular, through the statistical parameters of the mobile signal and the number of “hops” between one mobile cell and another, it is possible to identify the level of rainfall with a certain accuracy;

3. Together with the radio parameters, also the Packet Error Rate values for each SIM were recorded, drawing the conclusion that the level of rainfall varies hand in hand with the Packet Error Rate. Furthermore, the variation of the latter is also linked to the variation of the radio parameters;

4. The VPN Bonding algorithm initially estimates the bandwidth for the two connections and assigns the weights. When the rain is detected through

the classification given by the neural network, it recalculates the weights according to the estimated Packet Error Rate value.

In general, this study proposes a method to solve the problem of radio channel fluctuations caused by the presence of rain, since the trend SNR, RSRP and CID negatively affect the network performance. In this regard, a scheme is proposed to detect rain and improve the radio link of critical equipment such as, for example, modems on highways for the road safety, city units, as well as video surveillance systems for forensic contexts and similar.

The work is structured as follows: in sub-section 4.5.1 a review is carried out on standard methods for calculating the PER using the parameters of CQI; a correlation between PER and statistics radio parameter (illustrated in sub-section 4.4) is presented in sub-section 4.5.2; in sub-section 4.5.3 the VPN Bonding Smart algorithm is described. In sub-section the testbed (sub-section 4.5.4) and the experimental results (sub-section 4.5.5) of the proposed solution are provided.

4.5.1 Standard Calculation of PER using the CQI

In this section, we will illustrate PER compensation techniques using Channel Quality Indicator (CQI) mechanisms found in the literature, through which the level of degradation of the radio channel is estimated. Finally, once the state of the art has been analysed, in the second part of the section we will illustrate the rationale behind the technique proposed in this paper.

In the literature, we find that as telecommunications technologies vary, so does the way the CQI is calculated. In UMTS systems, for example, the CQI is mainly related to three parameters: Signal-to-Noise Ratio (SNR) [155], Signal-to-Interference plus Noise Ratio (SINR) [156] and Signal to Interference Ratio (SIR) [157]. In the LTE standard, however, the calculation of the CQI depends on the SINR [158–160]. This parameter is given by the power of the signal and the noise received by the device plus the interference given by other radio channels. In addition, the CQI may contain information about the best part of the spectrum on witch to receive the data. This is used for frequency domainscheduling. Generally, a high CQI value indicates a high quality channel and vice versa.

Other studies propose the estimation of CQI, in LTE systems, by employing the SNR and RSSI together with deep learning techniques [161, 162].

Even in 5G systems, several studies have addressed this aspect using the SNR parameters [163] and SINR [164].

In general, therefore, by analysing traditional techniques from the literature, we can deduce a packet error rate dependency with CQI of this type (4.9):

$$PER = f\left[CQI\left(SNR, SINR, RSSI, SIR\right)\right] \tag{4.9}$$

In this paper, we propose a new method of PER compensation specifically designed for rainfall scenarios. In particular, as is well known in the literature, rainfall intensity does not have a very noticeable impact on radio signal strength when frequencies are below 10 GHz. In fact, studies carried out in [116, 165] have shown that, for a correct and accurate estimation of rainfall levels, it is advisable to analyse, for example, the variance of power levels and other parameters calculated by the UE that are related to the impact rain has on cell selection and handover and not just to channel quality.

The experimental analysis of this work also shows that, in rainfall scenarios, PER compensation methods based on traditional CQI criteria fail to guarantee an adequate level of quality of service in particular application scenarios. It is therefore necessary, in this case, to take advantage of new mechanisms for estimating the degradation of the channel quality based on the analysis of the level of rainfall and, therefore, on the actual impact this has on the signal quality, also through the use of non-standard CQI parameters such as the CID_{HOPS}. Taking into account the rainfall estimation approach presented in the paragraph 4.4.5, the idea is, therefore, to model, specifically during rainfall events, the packet loss through the relation (4.10).

$$\begin{aligned} PER^{rain} = g\,[CNN_{r_{est}}\,(&CID_{HOPS}, RSRP, SNR, \\ &RSRQ, RXL, CID_{PERMANENCE}, CID_{CSR})] \end{aligned} \tag{4.10}$$

A specific implementation of the model (4.10) is presented in the next subsection.

4.5.2 Correlation between packet error rate and statistics radio parameter

In this study, we empirically correlated the values of PER with a subset of the parameters used as input to the CNN network, namely: $RSRP$, SNR and CID_{HOPS}. The relationship derived is as follows:

$$PER_n^{rain} = \frac{1 - 10^{\frac{RSRP_n}{SNR_n} - 0.2 \cdot \log_{10}(CID_{HOPS_n} + 1)}}{10} \tag{4.11}$$

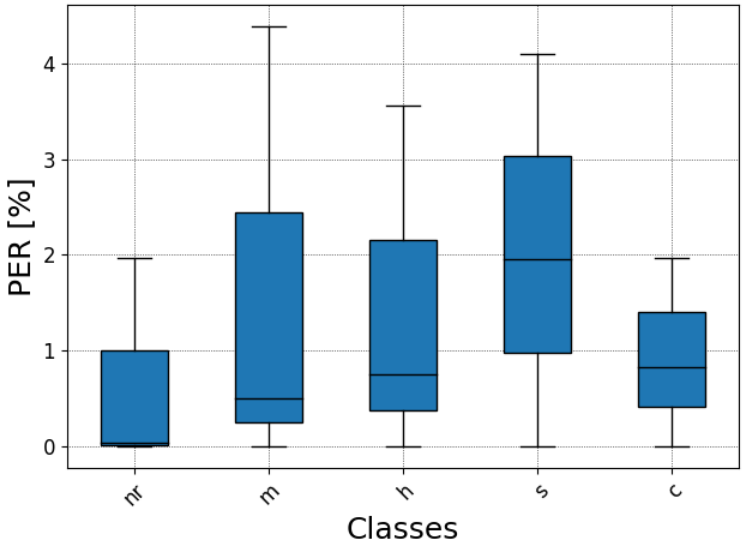


Figure 4.30: Percentage change in PER as precipitation level changes.

where $n = 1, ..., N$ is a number of connection. In our case $n = 2$ (Operator 1 and Operator 2)

The graph in Figure 4.30 shows the development of the PER as the intensity of precipitation change, according to 4.11.

In line with the data shown graphically in Figure 4.22, the graph in Figure 4.30 outlines that the PER increases as the level of precipitation increases (from no rain to shower) with a decrease in the cloudburst class.

Since PER is correlated to the $RSRP_{\text{VAR}}$, SNR_{VAR} and CID_{HOPS} values, it is consistent with the fact that in a cloudburst situation, since the channel is somewhat more stable (the rain is more evenly distributed and therefore there is only signal attenuation effect), PER also stabilises at a threshold that is higher than in the case of no rain. In the sub-section of experimental results, the effect of the VPN Bonding algorithm on PER obtained using two radio links of two different operators at varying rainfall levels is evaluated.

4.5.3 The Smart VPN Bonding Function

VPN bonding is usually employed in Ethernet switches but in the present study it is exploited to counteract the drawbacks related to wireless links of mobile access networks such as 3G, 4G and, incoming, 5G in rainfall conditions. VPN bond is responsible for balancing the data load among the available network interfaces enhancing the key performance indicators of the end-to-end path between the source and destination nodes. Our approach is based on the following steps:

i. Correlation of packet error rate with different rainfall levels through the CNN;

ii. Evaluation of weights to be assigned to each active network interface;

iii. configuration of the VPN bond by assigning the calculated weights to each active network interface.

As regards point (i), authors in [166] proposed a method for the fast analysis and evaluation of the KPIs in a mobile Internet access scenario such as Internet on board drone proving the suitability of the Self Loading of Periodic Streams (SLoPS) technique [167].

This technique involves periodically sending data across mobile interfaces and estimating the available bandwidth by evaluating the inter-arrival time between the transmitted packets. Although the algorithm has shown promising results in terms of accuracy and convergence time, it has two drawbacks: the waste of bandwidth due to the transmission of test packets and the complexity introduced by the use of a client-server application for which it is necessary to intervene both at the edge of the network and in the Operator Workstation in the Control Room. The technique proposed in this paper does not require any intervention from the Control Room and does not need the transmission of test data traffic to estimate the conditions of the wireless link. The estimation of rain is performed directly by the edge device which, based on what is detected, evaluates the packet error rate and defines the weights of the VPN bonding algorithm.

Once the KPIs of the individual connections have been estimated, the weights to be assigned to each connection are calculated using Algorithm 2. First of all, the weights are calculated on the basis of the service times (ST) guaranteed by each connection (ST_1 and ST_2) as can be observed in (4.12) and (4.13).

$$ST_1 = L_1 + \frac{PKT_{Size}}{BR_1} \tag{4.12}$$

$$ST_2 = L_2 + \frac{PKT_{Size}}{BR_2} \tag{4.13}$$

where, L_1 and L_2 are, respectively, the latency relative to connection 1 and 2; $\frac{PKT_{Size}}{BR_1}$ and $\frac{PKT_{Size}}{BR_2}$ are, respectively, the bandwidth relative to the connection 1 and 2. The weights, W_1 and W_2, are, therefore, calculated proportionally to the service rate (μ) offered by each mobile connection (μ_1 and μ_2) according to equations (4.14) and (4.15).

Algorithm 2 Dynamic VPN-Bonding algorithm.

```
    Input: $\mu_1, \mu_2, PER_1, PER_2$
    Output: $W_1$, $W_2$, $PER_{Bonding}$
 1: procedure
 2:     if $\mu_1 > \mu_2$ then
 3:         $W_2 \leftarrow 10$.
 4:         $W_1 \leftarrow round(\frac{\mu_1}{\mu_2}10)$.
 5:     else
 6:         $W_1 \leftarrow 10$.
 7:         $W_2 \leftarrow round(\frac{\mu_2}{\mu_1}10)$.
 8:     $PER_{Bonding}$ in Eq. 4.16
 9:     if $(PER_{Bonding} > PER_{th}) \wedge ((PER_1 < PER_{th}) \vee (PER_2 < PER_{th}))$ then
10:         do
11:             if $PER_1 \geq PER_2$ then
12:                 $W_1 - -$.
13:                 $W_2 + +$.
14:             else
15:                 $W_1 + +$.
16:                 $W_2 - -$.
17:             $PER_{Bonding}$ in Eq. 4.16
18:         while $(PER_{Bonding} \geq PER_{th})$
```

$$\mu_1 = \frac{1}{ST_1} \tag{4.14}$$

$$\mu_2 = \frac{1}{ST_2} \tag{4.15}$$

Once this step is completed, Algorithm 2 shows the pseudo-code of the dynamic VPN Bonding algorithm used in this study. The input of our algorithm are the service rate (μ_1 and μ_2) and the packet error rate (PER_1 and PER_2) parameters for each connection. The output of the algorithm are the weights (W_1 and W_2) to be assigned to each radio link of the aggregated connection. At the begin, the provisional weights of each connection, W_1 and W_2, are estimated proportionally to the values of the service rate, calculated in equations (4.14) and (4.15) accordingly to the procedure illustrated in lines from 2 to 7. Then, the value of PER of the VPN bonding connection, $PER_{Bonding}$, begins by correlating the estimated PER of each wireless link and the weight that has been assigned to each connection. In this way, the PER of the aggregated connection can be predicted by using 4.16):

$$PER_{Bonding} = \frac{W_1}{W_1 + W_2} PER_1 + \frac{W_2}{W_1 + W_2} PER_2 \tag{4.16}$$

where W_1 and W_2 are, respectively, the dynamically assigned weights for connection 1 and 2; PER_1 and PER_2 are, respectively, the Packet Loss Rates for connections 1 and 2 evaluated by exploiting (4.11).

From line 9 to 18 a condition is inserted; if respected, i.e. the PER of the aggregated connection is lower than a threshold value PER_{th} (which depends on the type of application/service), the weights are used to configure the bonding; otherwise the weight of the connection with the highest probability of error is lowered until the bonding value falls below the threshold value.

The use of dynamic VPN Bonding, compared to the static one presented in [168], allows performing the assignment of weights to each connection in a dynamic way. It was preferred considering that in the system under analysis a compensation of the packet error rate is provided based on the amount of rain detected (via the neural network and radio parameters), which can vary from one connection to another.

In [168], in fact, it was observed that the dynamic VPN Bonding technique offers superior performance; however, we are more interested in having a dynamic system that can adapt to the rainfall levels and the quality of detected radio parameters.

4.5.4 The Testbed Scenario

A typical use in the road sector was assumed for the testbed. Nowadays, motorways are equipped with a communication infrastructure based either on optical fibre or on modems/routers equipped with mobile radio connectivity. Our scenario is based on these premises and on the fact that mobile modems are often equipped with dual USIMs to guarantee redundancy of the connection. In addition, these devices integrate virtual private networking functions in order to guarantee the confidentiality of communications towards the operations room or the body managing the infrastructure or the police force. In this scenario, the introduction of a mechanism of rainfall estimation and Radio Packet Error Rate compensation based on VPN bonding appears extremely simple and technically feasible. More specifically, the scenario we envisaged concerns video surveillance and road monitoring by means of cameras connected to the operations room via a mobile radio network. The testbed was created with an edge node consisting of a Raspberry Pi 3 single board PC equipped with a camera and two 4G USB

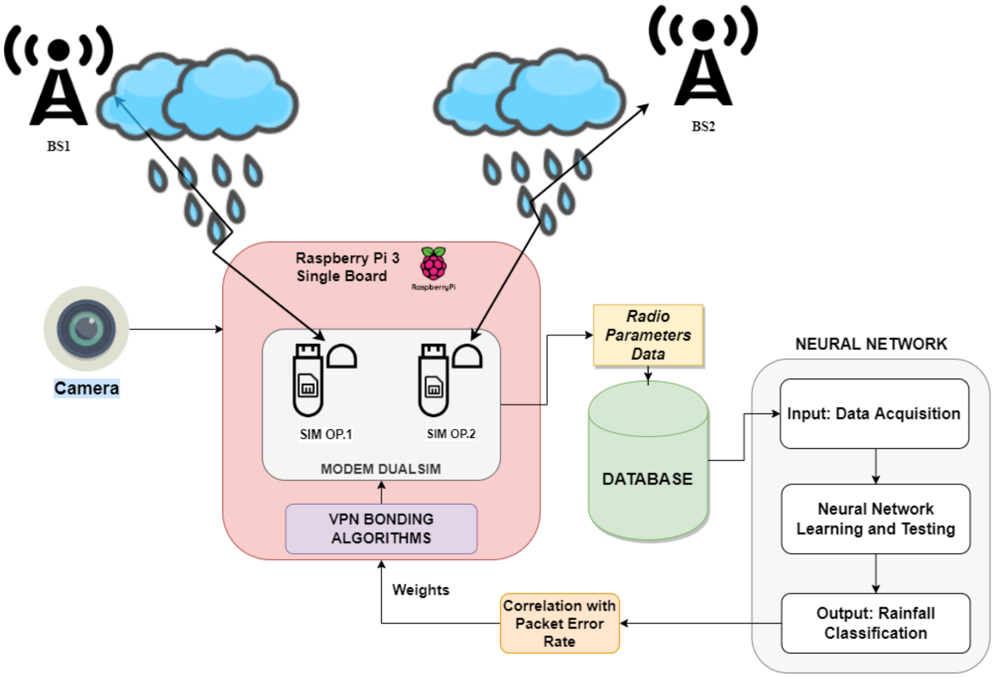


Figure 4.31: Testbed overview.

modems. The open-source software MLVPN was used for the application of VPN Bonding. This software enables the creation of end-to-end encrypted tunnels between the edge node and the remote Control Room and allows adapting the run-time of the bonding parameters on the basis of key performance indicators (latency, bandwidth, packet service rate and packet error rate correlated to the rainfall classification) and according to the weights evaluated by means of the proposed VPN-bonding algorithm, described in the previous Section.

The Smart UE was equipped with two SIMs belonging to different mobile operators, hereinafter referred to as Operator 1 (Op. 1) and Operator 2 (Op. 2), and placed at a distance of 740 metres from the BS of Op. 1 and 810 metres from the BS of Op. 2. The two BS are placed at a distance of about 1 Km. The diagram is shown in Figure 4.31.

In general, therefore, the estimates of rain levels made on each SIM could also be different because the two SIMs belong to two different operators and hook up to two different radio base stations, located at different distances and directions from the Smart UE.

4.5.5 Experimental Results

The aim of this study is to compensate for the packet loss rate when increased rainfall levels are detected through Smart VPN Bonding techniques. The objective is to guarantee the quality of the video stream even when the channel is affected by rain. The conducted tests compare the quality of the source video and the received video through the VMAF index [169].

The metric provides an overall performance indicator with values between 0 and 100 as a result of the comparison, where the higher the value returned, the higher the quality of the received video and, therefore, the end-user experience. To perform the test, data were collected on a rainy day, for the five precipitation intensities, in order to collect the performance indices in a video transmission using singularly Operator 1's access network, singularly Operator 2's access network and, finally, the intelligent VPN bonding algorithm.

The testbed was carried out by holding 5 recording sessions with a time duration of 10 minutes, for each intensity class. The data are shown in Table 4.9 and correspond to the average of the values over 10 minutes for each of the five sessions for each class.

From Table 4.9 it can be seen that the data for *PER Op. 1* and *PER Op. 2* are different due to the impact of rainfall on each connection relative to different operators. The video captured by the Raspberry cam (source video) and the videos received in the server were compared offline using the VMAF tool and calculating the performance index for 10-minute time windows.

The results are presented in Figure 4.32, which shows the comparison between the performance obtained using the individual mobile radio connections and the performance obtained using the smart VPN bonding technique, i.e., the aggrega-

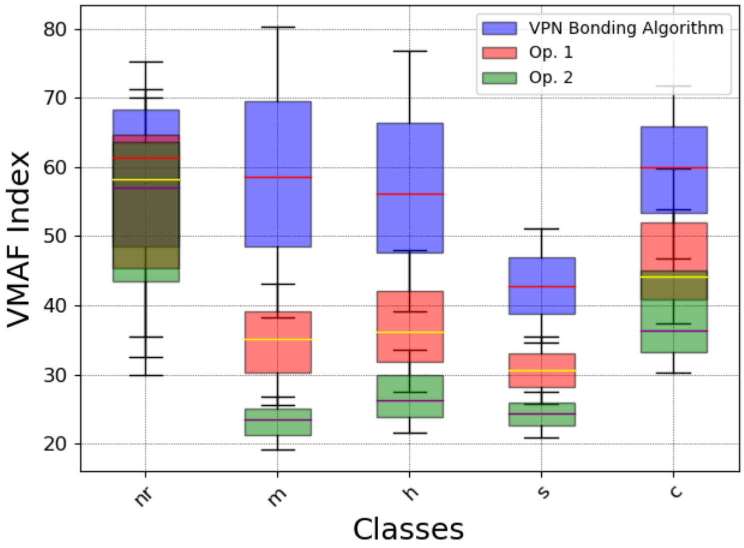


Figure 4.32: Comparison of VMAF index for single connections and Smart VPN bonding algorithm for different rainfall intensity levels.

Table 4.9: Comparison of PER and VMAF index for each single operator and with smart VPN bonding algorithm for different rainfall classes.

Classes	Test	VMAF with VPN Bonding	VMAF Op. 1	VMAF Op. 2	PER [%]	
					Op. 1	Op. 2
nr	1	71.29	75.29	70.10	0.00	0.00
	2	67.23	69.23	66.20	0.20	0.20
	3	59.80	63.80	59.80	0.50	0.50
	4	60.21	63.21	59.00	0.70	0.70
	5	32.54	35.54	30.00	1.00	1.00
m	1	80.37	43.06	24.40	0.20	1.50
	2	75.15	42.94	26.84	0.50	2.00
	3	47.74	29.84	20.89	1.00	2.50
	4	51.65	34.43	25.82	1.50	3.00
	5	38.25	25.50	19.12	2.00	4.00
h	1	71.71	41.37	26.20	0.40	1.50
	2	76.75	47.97	33.58	0.80	2.00
	3	50.79	33.86	25.39	1.20	2.40
	4	42.66	30.47	24.37	1.80	3.00
	5	39.20	27.44	21.56	2.00	3.50
s	1	51.15	35.41	27.54	1.00	1.80
	2	45.16	30.84	23.68	1.50	2.80
	3	47.94	34.24	27.40	1.80	3.00
	4	35.45	25.78	20.95	2.50	4.00
	5	34.58	26.41	22.32	2.90	4.20
c	1	71.83	59.85	53.87	0.40	0.50
	2	65.26	41.95	30.30	0.80	1.80
	3	65.25	44.27	33.79	1.00	1.90
	4	50.63	37.50	30.94	1.30	2.00
	5	46.74	37.39	32.71	1.50	2.00

tion of the two radio connections. The results indicate that the proposed approach allows to compensate for the PER level as the rainfall level increases, resulting in an improvement of the radio channel quality. Moreover, as expected, during "no rain" scenario the PER performance, when exploiting the single operators or the VPN Bonding algorithm, are comparable.

In general, as seen in the graphs, the intermediate rainfall levels (m, h and s) have a similar impact on the performance of the radio channels and it is actually possible to further simplify the rainfall level classification and estimation technique to only two classes nr, m/h/s/c.

Ultimately, the main factor we want to highlight in our study is that the Smart VPN bonding system is activated when rain is detected and thus the PER begins to degrade. From the graph, in fact, when we are in the "no rain" case there is no point in activating VPN Bonding. Viceversa, it is automatically activated when we are in the other classes, which, as already mentioned, can be incorporated into a single “no rain” class.

4.5.6 Discussion and Conclusion

The approach we propose in this paper is based on the use of a CNN for rainfall estimation and an intelligent VPN bonding technique for compensating radio packet errors (PER) when rain is detected. The objective of this study is to guarantee the quality of service when the channel is heavily degraded by the effect of rain. This is achieved by utilising new mechanisms for estimating channel quality degradation based on the analysis of the level of rainfall and its actual impact on signal quality, using, for example, non-standard CQI parameters such as CID_{HOPS}. Taking into account the rainfall estimation approach presented in the 4.4.5 section, the idea is to model the packet loss through the empirical relation in (4.11). The reasons for introducing this new technique are many and will be explained below:

- Several techniques exist in the literature for calculating packet error ratios directly from channel quality indicators and compensating mechanisms, but they are all at the base station level. The proposed mechanism allows an UE terminal, equipped with two or more wireless connections, to assess the PER, correlating it to the estimated rainfall level, and compensate for it by activating the adaptive bonding mechanism on the available wireless links. The proposed device will therefore be able to connect to existing networks without the need to make changes to the existing infrastructure;

- The rainfall estimation algorithm, once the rainfall condition is detected, automatically and dynamically activates the QoS compensation technique through VPN bonding. Exploiting this technique avoids wasting bandwidth and energy because it is not necessary to periodically send control packets to evaluate the performance of individual wireless connections: the standard VPN Bonding technique is based, in fact, on sending trains of data packets (SLoPS technique) over the network in order to evaluate the performance of wireless links. Furthermore, the adaptation of the weights in the proposed Smart VPN Bonding technique occurs dynamically in a time (15 seconds) typically halved compared to traditional techniques;

- The system and its proposed prototype lend themselves to a dual use: the first as a meteorological station capable of estimating rainfall levels and activating all the various alerting mechanisms when the radio parameters indicate that rain is increasing (decreasing quality); the second as a Smart UE, suitable for road monitoring, security and video surveillance applications, capable of compensating for the increase in PER, experienced by the radio transceivers in rainy situations, through the activation and adaptation of the VPN bonding mechanism.

Chapter 5

An ML-based automatic analysis of cardiac pathologies for advanced telemedicine services

The increase in life expectancy and the consequent increase in the healthcare demands of the population, with a growing proportion of elderly people and chronic diseases, have led to a consequent evolution of innovative care models based on data analysis techniques, sensors and highly advanced diagnostic tools.

Lately, the increasing use of machine learning and deep learning algorithms has led to the emergence of new techniques for automatic disease monitoring and recognition in different biomedical area [170, 171]. In particular, many studies focus on the recognition of heart disease [57–63].

In sub-section 5.1, as described in [57, 172], the technique of recognizing different heart diseases through ECG signals and convolutional neural networks is illustrated. In particular, two tests are carried out: the first test is about using only the MIH-BIT Arrithmya dataset as input to the neural network; the second test is about merging the MIH-BIT Arrithmya and Atrial Fibrillation datasets into a single database. For both scenarios, the neural network was trained and tested, providing the results on the classification accuracy of heart disease. In sub-section 5.2, as described in [60, 173], the technique of recognizing different heart diseases through PCG signals and convolutional neural networks is illustrated. Again, two tests are performed: the first is related to the use of a public dataset consisting of 5 classes of diseases. The network is trained with clean PCG signals and tested with clean PCG signals, noisy (addition of "office noise") and real-time signals. The second test relates to the combining of two public datasets into a single dataset consisting of 7 disease classes. In this scenario, the network was trained and tested with clean and noisy PCG signals (“office noise” and “bubble noise”) where noise was added to the clean signals with different SNR.

5.1 Automatic Heart Pathologies Recognition Using ECG Signal

For many years, doctors have been aware that cardiovascular diseases constitute a class of diseases considered to be one of the main causes of mortality [56]. Cardiovascular diseases occur in the form of MI. Myocardial infarction, commonly referred to as “heart attack”, stands for the failure of heart muscles to contract for a fairly long period of time. Using appropriate treatment within an hour of the start of the heart attack the mortality risk of the person who suffers from a heart attack in progress can be reduced. When a heart condition occurs, the first diagnostic check consists of an ECG, which therefore is the main diagnostic tool for CVD. The electrocardiograph detects the electrical activity of the heart during the test time, which is then represented on a graphic diagram that reflects cyclical electrophysiological events in the cardiac muscle [174]. By conducting careful analysis of the ECG trace, doctors can diagnose a probable myocardial infarction. It is important, however, to underline that the sensitivity and specificity of manual detection of acute myocardial infarction are 91% and 51%, respectively [175]. Developing a computer-aided system to automatically detect MI would help the cardiologists make better decisions. Hence, lately various studies have been conducted on automatic MI detection. Sub-section 5.1.1 describe the electrical activity of the heart during its operation. An overview of the state of the art on cardiac disease recognition techniques using ECG signals and neural networks is given in sub-section 5.1.2. In the sub-sections to follow, two major studies [57, 172] on cardiac disease recognition using one or more ECG datasets and CNN networks are described. The first study, proposes a low-complexity solution for automatic heart disease recognition based on the direct application of a CNN-based classification network to EGC signals, thus bypassing any possible heart disease ECG signals from the time domain to other domains (e.g. frequency domain as MFCC, Wavelet, etc.). This paper evaluates the performance of a classifier in the following three classes: “Normal”, “Atrial Premature Beat” and “Premature Ventricular Contraction”. The second one, extend previous work [57] by increasing the number of cardiac diseases to be classified. In particular, two public datasets are used: MIT-BIH Arrhythmia and the MIT-BIH Atrial Fibrillation. Again, the CNN neural network model, described in Appendix C, and to the ECG signals, no pre-processing step is performed. These studies were carried out during the PhD period.

5.1.1 ECG Signal

From a graphic or numerical point of view, ECG represents the electrical activity of the heart during its operation. The most important elements of an ECG waveform, which repeats for each cardiac cycle, are shown in Figure 5.1.

ECG is carried out to provide information about different heart diseases that a person can suffer from [176], in order to guarantee effective therapy. According to international conventions, the specific points that are identified in the trace of an electrocardiogram are labeled with the letters P, Q, R, S, T, and, in particular, are the following:

- P wave: the first wave that occurs in the ECG cycle, a small deflection that represents Atrial depolarization or most commonly called “Atrial Contraction”;
- T wave: represents the depolarization of ventricles or most commonly called “Ventricular Relaxation”;
- Q, R and S waves: together, these waves form the so-called QRS complex.

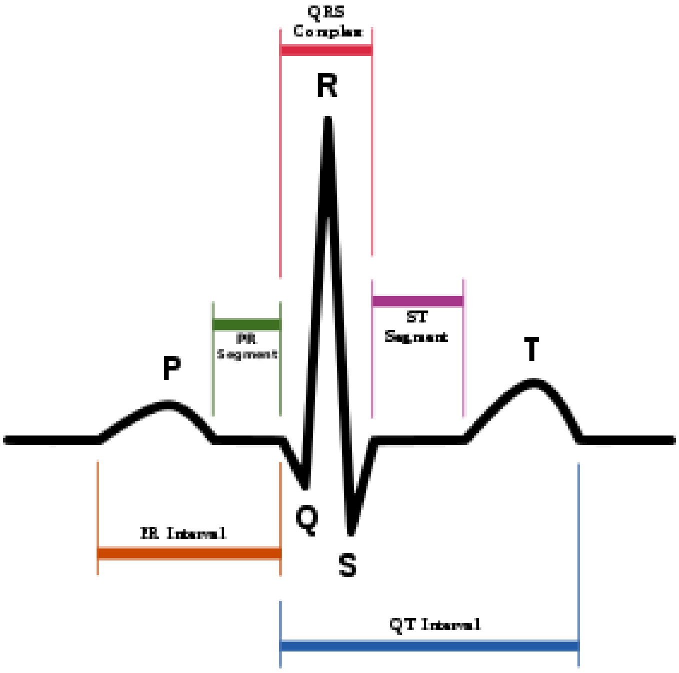


Figure 5.1: A typical ECG waveform and its characteristic patterns (P and T waves, PR and ST segments, PR and QT intervals, as well as the QRS complex).

The QRS complex represents the contraction of the ventricles or, technically speaking, the depolarization complex of the ventricles. In particular, the Q wave represents the depolarization of the interventricular septum, the R wave reflects the depolarization of the main mass of the ventricles, and the S wave is the final depolarization of the ventricles at the base of the heart.

Taken together, the P, Q, R, S and T waves make up the so-called $PQRST$ complex. Cardiologists denote the interval between two $PQRST$ complexes by the term "$R - Rinterval$", which corresponds to a cardiac cycle.

Other parameters, which have been extensively used to make medical diagnoses using the ECG trace, are:

- PR interval or PQ interval: the PR interval is a stretch formed by the P wave and the PR segment (rectilinear stretch) that begins with the P wave, that is, during the first deflection, and ends at the QRS complex. This interval indicates the time that the depolarization wave takes propagating from the atrial sinus node along the part of the electrical conduction system of the heart present on the myocardium;
- ST segment, i.e. the time between the end of QRS complex and the start of the T wave;
- QT interval, i.e. the time between the beginning of the QRS complex and the end of the T wave, which is the electrocardiographic manifestation of ventricular depolarization and repolarization [177].

5.1.2 Related Works

CNN have been utilized in arrhythmia detection, coronary artery disease detection, beats classification [68, 178, 179]. A deep belief network has been used to classify signal quality in ECG [180]. Some researchers have implemented an 11 layers CNN to detect MI [181]. The authors have demonstrated the use of a shallow CNN only focusing on inferior MI. This network benefits from the use of varying filter sizes in the same convolution layer which allows it to learn features from signal regions of varying length. In [182] the authors propose a classification system of cardiovascular diseases using the MLP network and the CNN network. In particular, they compare the results obtained by both models, using the same data set but different classes. There are two classes used in the MLP network: "Arrhythmias" and "Normal", while those used for $4-$ layer CNN are nine classes. ECG data used for the training / validation and test dataset was downloaded from two public dataset. This study shows low performance both using the MLP network and the CNN network, i.e. 88.7% and 83.5%., respectively.

There are many other studies that deal with the classification of heart disease via the ECG signal using deep learning algorithms based on CNN.

In [183] the authors use the RR interval to represent temporal characteristics, focusing on the symbolic signal metric as the input. This contributes to a more comprehensive study of the model and ensures 96.4% accuracy in the diagnosis of arrhythmia among patients, using the MIT-BIH Arrhythmia dataset and the Multi-Perspective Convolutional Neural Networks (MPCNN).

In [184] the authors develop a deep learning-based approach for multi-label ECG classification called Multi ECGNet, which can effectively identify patients with multiple heart diseases at the same time. A network model used is based on the integration of ResNet, Xception and SE module. Experimental results show that the performance can achieve an average $micro - F_1$ score of 86.3% in classifying 55 types of arrhythmia. The dataset is retrieved from the 2019 Tianchi Hefei High-Tech Cup ECG Human-Machine Intelligence Competition, which contains 32,142 cases, 24,106 of which are used as the training set and 8,036 - as the test set.

In [185] the authors adopt a procedure consisting of a learning phase in which classification accuracy is improved through a major feature extraction protocol. A Genetic Algorithm (GA) process is used to aggregate the best combination of feature extraction and classification. The execution of the proposed strategy involves feeding the ECG signal into the Deep Neural Network (DNN) model in which the deep features of each patient are extracted. Subsequently, these features are entered into a GA that determines their optimal combination. Several classifiers, including k-NN, SVM and multilayer perceptron (MLP), are used to classify the features, and, finally, the detection of cardiovascular diseases is performed. The results show that the proposed model achieves average accuracy values of 98% in arrhythmia detection on the MIT-BIH Arrhythmia Dataset.

In [186] a multi-model system is proposed consisting of two different deep learning bagging models. The first model is based on CNN and LSTM architectures and takes raw ECG beats as input; the second model is based on the combination of classical features, i.e. RR and HOS intervals, and the LSTM model. ECG data is pre-processed (RR and HOS segment extraction is performed) before feeding it to the network. The input of this model includes 5 consecutive heartbeats, in which 2 previous and 2 subsequent heartbeats accompany the current heartbeat. Performance evaluation of the proposed method is performed using 6 metrics:

Accuracy [%], F_1*Score* [%], *Specificity* [%], *Sensitivity* [%], *Positive Predictive Value* [%] and *Cohen's Kappa* . The dataset used is the MIT-BIH Arrhythmia with average accuracy of 95.81%.

In [187] the authors develop a 16-layer CNN-LSTM model to classify CAD, MI and CHF signals. The lead II ECG signals used in this study are acquired from healthy subjects and patients from four databases. The signals of 92 normal subjects, 7 CAD patients, 148 patients with myocardial infarction and 15 patients with CHF are obtained from PTB Diagnostic ECG Database and Fantasia Databases, St. Petersburg Institute of Cardiological Technics 12 lead Arrhythmia Database, PTB Diagnostic ECG Database and BIDMC Congestive Heart Failure Database, respectively. The classification *Accuracy*, *Specificity*, *Sensitivity* and *Positive Predictive Value* are 98.51%, 97.89%, 99.30%, 97.33%, respectively, and they are obtained with the proposed deep learning model.

In [188] the authors base their research on 15 different classes from the MIT-BIH arrhythmia dataset. They then propose a new data augmentation technique using Generative Adversarial Networks (GANs) to restore the balance of the dataset. Two deep learning approaches are used: an end-to-end approach and a two-stage hierarchical approach based on deep cCNNs to eliminate manual engineering features by combining feature extraction, feature reduction, and classification into a single learning method. With the end-to end approach, an overall accuracy of 98.30% and precision of 90% are achieved. The hierarchical approach achieved 98% overall accuracy and 93.95% precision.

5.1.3 Database

MIT-BIH Arrhythmia Dataset

One of the most commonly used databases in the literature is the MIT-BIH Arrhythmia Dataset [189, 190]. It consists of 48 two-channel ambulatory ECG recordings, each lasting 30 minutes, associated with different clinical conditions (e.g. ventricular and supraventricular arrhythmia, ventricular tachyarrhythmia, atrial fibrillation, etc.) and recordings of regular heartbeats. The database contains ECG recordings of 47 subjects: 25 males aged 32 to 89, and 22 females aged 23 to 89. 23 recordings were randomly selected from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Beth Israel Hospital in Boston, the remaining 25 recordings were selected from the same set to include less common but clinically significant types of arrhythmia that would not be well represented in a small random sample. The recordings were digitized at 360 samples per second per channel with a resolution of 11 bits over a range of $\pm 10mV$.

The dataset consists of 3 macro-disease classes: Normal Beat, Atrial Fibrillation Beat and Premature Ventricular Beat.

These 3 macro-disease classes could split in: Normal sinus rhythm (N), Paced rhythm (P), Atrial fibrillation (AFIB), Ventricular bigeminy (B) e Sinus bradycardia (SBR).

MIT-BIH Atrial Fibrillation Dataset

The MIT-BIH Atrial Fibrillation Database [191] consists of 25 ECG recordings of subjects with atrial fibrillation, many of which are of the paroxysmal type. 23 out of 25 recordings include two ECG signals (in .dat files); recordings 00735 and 03665 are represented only by the rhythm annotation (.atr) and unverified beat (.qrs) files. Each recording is 10 hours long, contains 2 ECG signals each sampled at 250 samples per second with 12-bit resolution in a range of ±10 millivolts. The original analogue signal, with a bandwidth of approximately 0.1Hz to 40 Hz, is also retrieved from Beth Israel Hospital. The .atr annotations (rhythm annotations) were produced manually while the .qrsc (beat annotations) were generated via an automated identification system.

The files contain rhythm annotations related to 5 classes of pathology: AFIB (atrial fibrillation), AFL (atrial flutter), J (AV junctional rhythm), and N (used to indicate all other rhythms).

Combined Dataset

In this section we will describe how the two datasets were combined in order to create a single dataset that can be used for training and testing the neural network. The two datasets have different characteristics, and the only common feature is the AFIB (atrial fibrillation) class.

The first difference lies in the sampling rate (f_s): the sequences in the MIT-BIH Arrhythmia Dataset have the f_s of 360 samples per second, whereas the ECG sequences in the MIT-BIH Atrial Fibrillation Dataset have the f_s of 250 samples per second. To combine the data, the original sequences were re-sampled at a frequency of $f_s = 1kHz$.

Regarding disease classes, the only class the two datasets have in common is the *AFIB*. In order to combine the two datasets uniformly, the authors have decided to remove the *N* class related to *MIT-BIH Atrial Fibrillation* (a class that includes

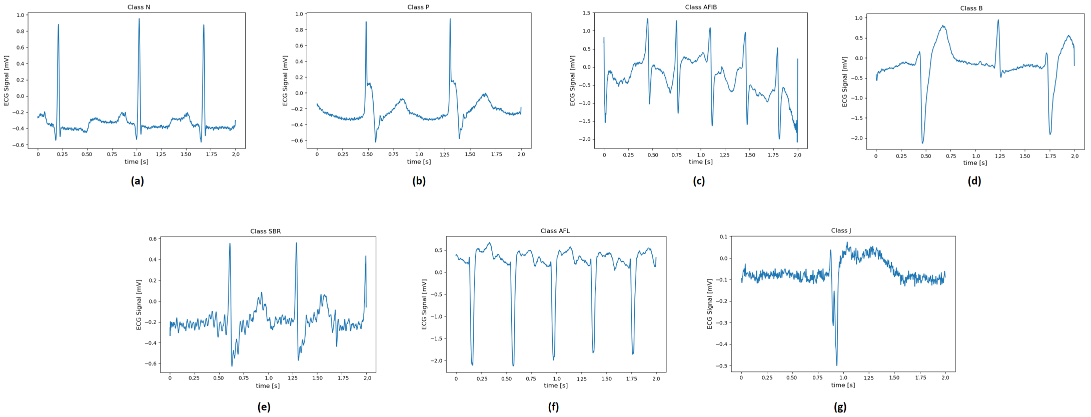


Figure 5.2: ECG signal for each type of diseases: (a) Normal Sinus Rhythm; (b) Paced Rhythm; (c) Atrial Fibrillation; (d) Ventricular bigemy; (e) Sinus bradycardia; (f) Atrial flutter; (g) Junctional rhythm.

unspecified rhythms) and the *AFIB* class related to *MIT-BIH Arrhythmia* (as it was already present in the other, more comprehensive dataset).

Based on these choices, the classes resulting from the combination of the two datasets are as follows:

- *N*: Normal sinus rhythm;
- *P*: Paced rhythm;
- *AFIB*: Atrial Fibrillation
- *B*: Ventricular bigemy;
- *SBR*: Sinus bradycardia;
- *AFL*: Atrial flutter;
- *J*: Junctional rhythm.

Figure 5.2 shows an example of two-second ECG recordings for each type of classified cardiac pathology.

5.1.4 Proposed Method

Method using only MIT-BIH Arrhythmia Dataset

Once the two public datasets were integrated and the classes of pathologies to be recognized were defined it possible train and test the neural network (described in Appendix C). NN input consists of 30-second segments where every second of

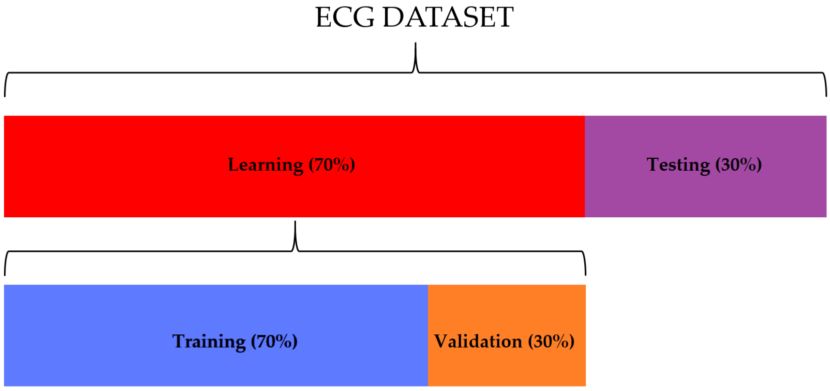


Figure 5.3: The distribution of ECG segments used for learning (70%) and testing (30%). 30 % of the learning dataset was used for the validation of the network.

ECG recording is equivalent to 360 samples, for a total of 10800 samples. So, the dataset presents the following classes:

- "Normal" class, containing 1421 ECG segments;
- "Premature Ventricular Contraction" class, containing 335 ECG segments;
- "Atrial Premature Beat" class, containing 133 ECG segments;

This dataset was subsequently divided into two different datasets, see Figure 5.3.

- Training/Validation set, consisting of 995 segments for the "Normal" class, 234 segments for the "Premature Ventricular Contraction" class and 93 segments for the "Atrial Premature Beat" class. 70% of this set was used for the training and the other 30% was used for the testing;
- Testing set, consisting of 426 segments for the "Normal" class, 101 segments for the "Premature Ventricular Contraction" class and 40 segments for the "Atrial Premature Beat" class.

At first, the network was trained by entering the data relating to the "Training set" as input, then it was validated using the "Validation set", in order to evaluate the performance of the neural network (the percentage of loss and accuracy). Finally, the "Testing set" was applied to validate and verify, through the accuracy estimate, the robustness of the neural network to data external to the training / validation set.

Method using MIT-BIH Arrhythmia and Atrial Fibrillation Dataset

Once the two public datasets are combined and the classes of pathologies to be recognized are defined, each individual ECG recording (of fairly long duration) is split up into chunks of 2-second segments for each class. By doing so, it is possible to extend the amount of recordings constituting each class.

As a result of the resampling process, the neural network input consists of 2-second segments where every second of an ECG recording is equivalent to 2000 samples.

The dataset presents the following classes:

- N class, containing 31570 ECG segments;
- P class, containing 3608 ECG segments;
- B class, containing 1804 ECG segments;
- SBR class, containing 902 ECG segments;
- AFIB class, containing 150157 ECG segments;
- AFL class, containing 2932 ECG segments;
- J class, containing 164 ECG segments.

This dataset was subsequently divided into two different datasets:

- Training/Validation set, consisting of 22099 segments for the N class, 2525 segments for the P class, 1262 segments for the B class, 631 segments for the SBR class, 105109 for AFIB class, 2052 for AFL class and 114 for J class. 70% of this set was used for the training and the other 30% was used for the testing;
- Testing set, consisting of 9471 segments for the N class, 1082 segments for the *P* class, 541 segments for the B class, 270 segments for the SBR class, 45047 for AFIB class, 879 for AFL class and 49 for J class.

At first, the network was trained by entering the data relating to the "Training set" as input, then it was validated using the "Validation set", in order to evaluate the performance of the neural network (the percentage of loss and accuracy). Finally, the "Testing set" was applied to validate and verify, through the accuracy estimate, the robustness of the neural network to data external to the training/validation set.

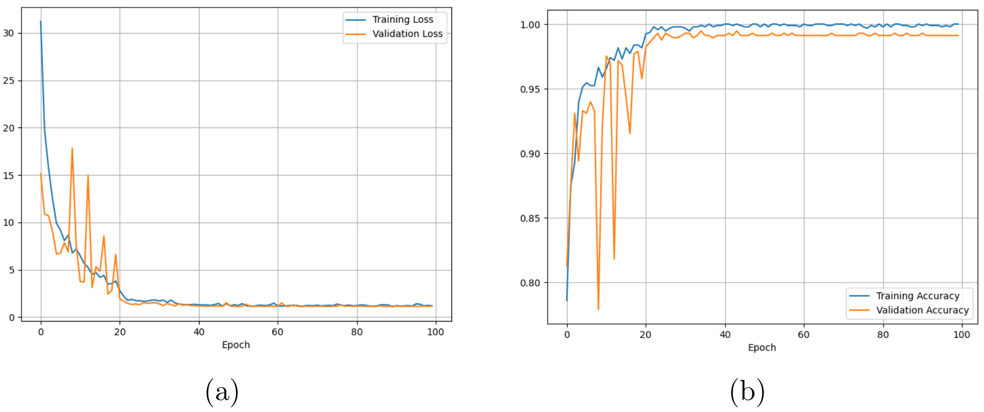
5.1.5 Experimental Results

Result using only MIT-BIH Arrhythmia Dataset

In this sub-section, the results of training and validation of the neural network are presented and discussed. Figures 5.4a and Figure 5.4b represent the progress of the training and validation loss and the progress of the training and validation accuracy, respectively. As the graphs show, after 100 epochs training and validation losses stabilize at a value close to zero (Figure 5.4a), while training and validation accuracy stabilizes at 100%. Such data are very encouraging, as it is understood that there is a good percentage of accuracy in the classification of the three classes described above.

In order to evaluate the performance of the CNN network with ECG sequences external to the training dataset, the accuracy obtained with the "Testing set" is assessed. Figure 5.5 shows the relative confusion matrix. The matrix highlights an average classification accuracy level of 98.33%. The results obtained in terms of the statistical parameters described in Appendix F, are shown in Table 5.1.

K-fold [192] cross validation was used in this study, which involves randomly dividing the training dataset into k parts without reintegration: the $K-1$ parts are used for training the model and a part is used for testing. This procedure is repeated k times so as to obtain k models and performance estimates. Subsequently, the average performance of the models is calculated on the basis of the different independent subdivisions to obtain an estimate of the performance that is less sensitive to the partitioning of the training data. Since k-fold cross validation is a resampling without reintegration technique, the advantage of this approach is that each sample point will be part of the training and test datasets



(a) (b)

Figure 5.4: Training and validation phase: (a) Loss function trend. (b) Accuracy function trend.

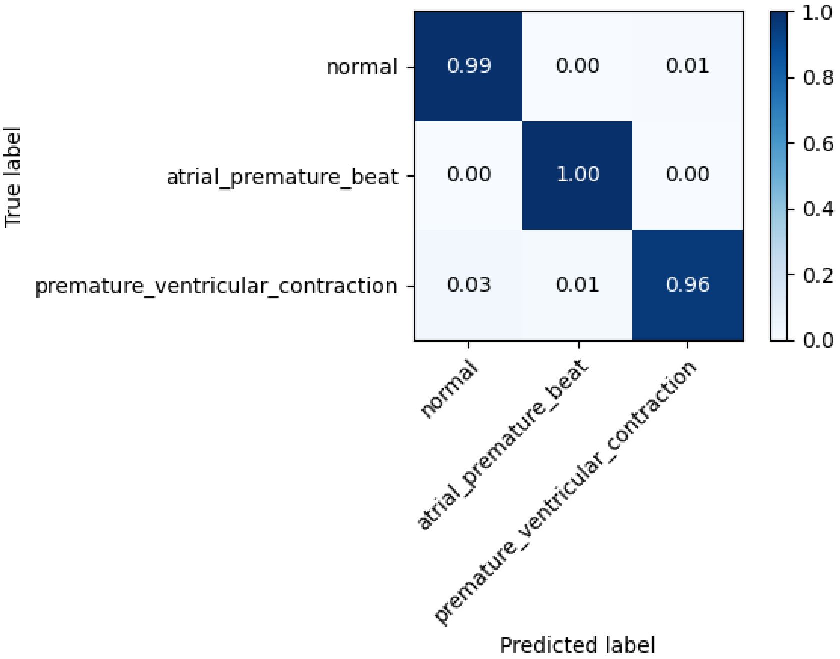


Figure 5.5: Confusion matrix for “Testing set”.

only once, which provides a lower variance estimate of the template performance. For this study, the training dataset is divided into ten parts, $K = 10$, and during the ten iterations, nine parts are used for training and one part is used as a test set for model evaluation. In addition, the estimated performance (for example, the accuracy of the classification) of each part is then used to calculate the average estimated performance of the model. Figure 5.6 depicts the concept of the k-fold cross validation technique. The average accuracy and standard deviation for the model used in this study is $96.8 \pm 1.2\%$.

Table 5.1: The table reports the overall values of accuracy TPR, TNR, TPR, TDR and $F_1 Score$.

α	Class	TPR [%]	TNR [%]	FPR [%]	FDR [%]	$F_1 Score$ [%]
1	Normal	99.0	97.1	2.9	1	98.0
2	Atrial Premature Beat	100	99.0	1.0	0	99.5
3	Premature Ventricular Contraction	96.0	98.96	1.04	4	97.5
Mean Accuracy	98.33 %	98.33	98.35	1.65	1.66	98.33

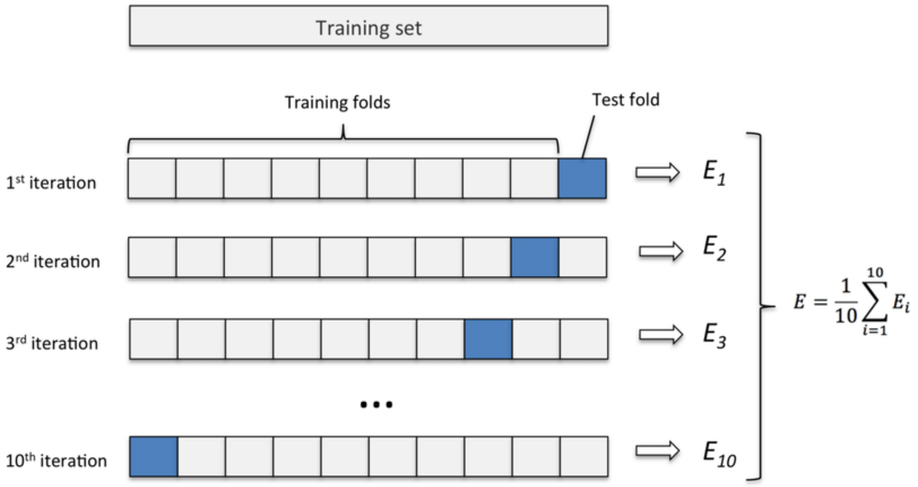


Figure 5.6: K-fold cross-validation method with subdivision of the training set into $k = 10$ parts.

Result using MIT-BIH Arrhythmia and Atrial Fibrillation Dataset

Figure 5.7a and Figure 5.7b show the trend of loss function and accuracy related to the training of the neural network, respectively. In both graphs, the blue curve refers to the training set, while the orange curve refers to the validation set. It can be observed that along with the increase in the number of epochs, i.e. the number of complete training cycles performed, there is a reduction in loss function and an increase in accuracy, which stabilize, respectively, after about 20 epochs (loss function) and 40 epochs (accuracy).

The testing phase of the network yields excellent results. The accuracy turns out to be quite high, about 93.3% considering 2-second chunks.

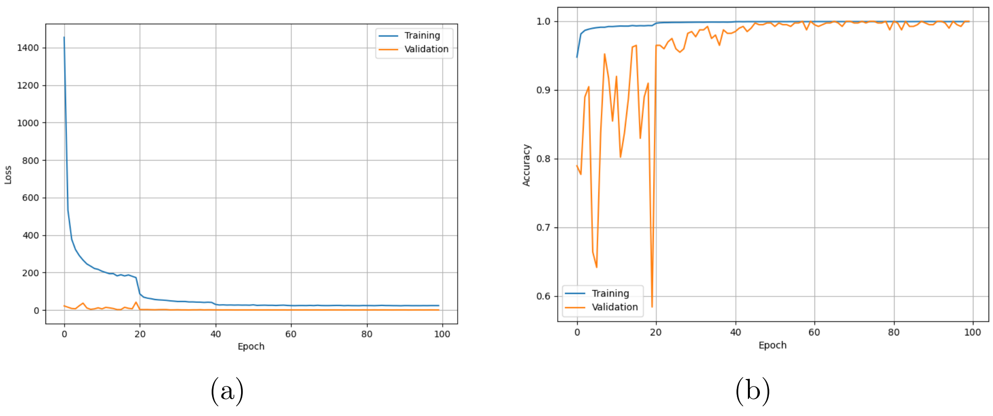


Figure 5.7: Training and validation phase: (a) Loss function trend. (b) Accuracy function trend.

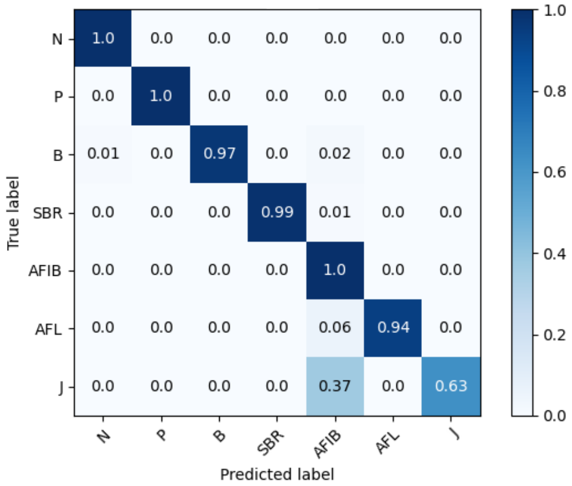


Figure 5.8: Testing phase: Confusion matrix.

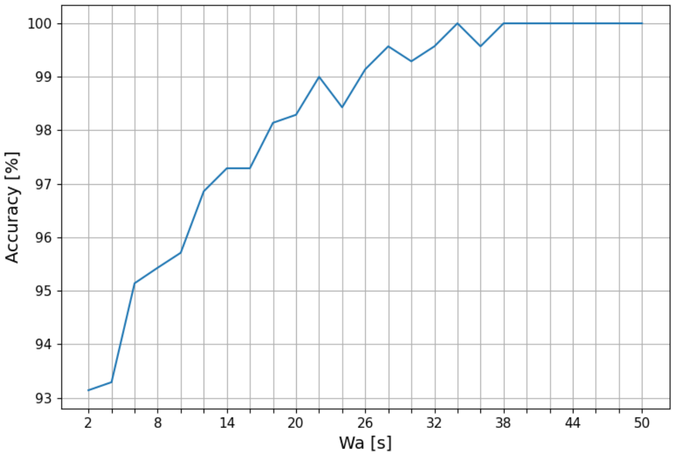


Figure 5.9: Testing phase: Accuracy as the analysis window changes.

The confusion matrix of the test set (Figure 5.8), denotes very good discrimination ability of the network among the various classes except for the J class. Most probably, this is due to the lack of data present in the class.

By applying the recurrence filter in post-processing, a final accuracy of about 100 percent is achieved after about 38 seconds (Figure 5.9). The use of this filter solves the problem of the J class.

5.1.6 Discussion and Conclusion

Method with only MIT-BIH Arrhythmia Dataset

Table 5.2 shows a comparison between our method and other methods in terms of feature extraction (FE), model used, system's accuracy, and the statistical classification accuracy.

Hereinafter the differences between this work and the state of the art will be discussed. In [193, 194] the authors use the extraction of decision tree (DT) and R-Peak (RP) as features and do not apply convolutional neural networks (CNN), but rather the Discrete Wavelet Transformation (DWT) and the Feed-Forward Neural Network (FFNN). The authors claim an average accuracy of 96.56% and 87.66% respectively, while in our study the average accuracy is equal to 98.1%. This result is higher than the study proposed in [193, 194].

Compared to the approaches proposed in [181, 182, 193–196], our method has higher classification performances. As far as the studies proposed in [197, 198] are concerned, it is evident that they have quite comparable performances, but they use more hidden layers than our study, with a consequent increase in computation costs. In addition, they do a preprocessing of data using wavelet transformation, which implies an additional computational cost. Also from the point of view of the structure of the neural network, in [197], in particular, 5 layers (two convolution layers, two down sampling layers and one full connection layer) plus the output layer formed by Softmax are used for classification, however, we use another structure (previously described), which is more robust to the "vanishing gradients" phenomenon.

In addition, to ensure that the model is correct, we apply the K-fold technique (previously described) for cross validation, obtaining an average accuracy of 96.8% and standard deviation of ±1.2%.

Usually, the processing unit implements the automatic disease classification algorithm described above, showing the result of the diagnosis on the display.

This paper proposed an automated heart disease recognition technique based on recent and innovative CNN networks. The proposed technique had high accuracy and had low complexity of implementation. This approach harnessed the potential of deep learning to capture the typical characteristics of given heart disease in the ECG signal domain.

Using the "Testing set", the proposed method yielded the following results: 98.33% mean accuracy; 98.33% sensitivity; 98.35% specificity; 1.65% false positive ratio; 1.66% false negative ratio; 98.33% $F_1 Score$.

Method with MIT-BIH Arrhythmia and Atrial Fibrillation Dataset

The results obtained show that by applying the post-processing filter 100% accuracy performance can be obtained on all classes after about 38 seconds. This result exceeds even the performance of the state of the art, in particular, the research using the MIT-BIH Arrhythmia dataset [183, 185–187].

Table 5.2: ECG Classification - Comparison between the proposed method and those previously studied.

Method	FE	Model	A [%]	TPR [%]	TNR [%]	FPR [%]	FDR [%]
[193]	DT	DWT	96.56	90.87	98.45	9.13	1.55
[194]	RP	FFNN	87.66	94.04	76.21	5.96	23.79
[181]	RP	11– Layer CNN	95.22	95,49	94.19	-	-
[195]	-	PNN	96.53	93.1	100	-	-
[182]$^{*,+}$	-	MLP 5 – Layer CNN	88.7/ 83.5	-	-	-	-
[196]	-	4 – Layer CNN	92.7	-	-	-	-
[197]	Wavelet transform	6 – Layer CNN	97.5	-	-	-	-
[198]	Wavelet transform	10 – Layer CNN	99.8	99.5	-	-	-
[57]	-	5 – Layer CNN	98.33	98.33	98.35	1.65	1.66

* Different dataset for training/validation and testing

+ Use more cardiovascular diseases classes

Moreover, this study has as an added value of not being limited to the use of only one dataset, but two datasets with similar characteristics are incorporated and aggregated together in order to form a unique set of data with multiple classes to be recognized. In addition, ECG data were fed to the network without performing any kind of pre-processing.

5.2 Automatic Heart Pathologies Recognition Using PCG Signal

Sub-section 5.2.1 describes the acoustic signal trend of the heart during its operation. An overview of the state of the art on cardiac disease recognition techniques using PCG signals and neural networks is given in sub-section 5.2.2. In sub-sections to follow, two major studies [60, 173] on cardiac disease recognition using one or more PCG datasets and CNN networks are described. These studies were carried out during the PhD period. In particular, the first study propose an automated low-complexity solution for heart disease recognition based on the direct application of a CNN classification network to raw data of a PCG signals (5 classes), without the need for domain transformation from time to frequency (e.g. MFCC, Wavelet, etc.). Further, this first approach, highlights the performance's dependence on office noise (only in testing phase). The second study is based on a classification algorithm that allows classifying the PCG signal in 7 classes, characterized by the presence of ambient noise (e.g. hospitals, etc.). The database used in this study is a combination of two open access datasets present in the literature. This study further extends the research proposed in [60] by using only the dataset splitting mode so as to obtain recordings totally unknown to the network in the testing dataset. The novelty it offers, in addition to the inclusion of raw PCG data directly as input to the CNN neural network, is a simulation of ambient noise (office and babble noise) within the clean PCG sequences.

5.2.1 Heart Sound Features

Sounds from the cardiac system are generated by the opening and closing of heart valves [56]. In fact, during the closure event, the simultaneous vibration of the valve, the adjacent heart structures and the surrounding blood rapidly propagates through the chest with an acoustic wave. With the help of a stethoscope we are able to hear the heart tones, while the phonocardiographs (and electrocardiographs) allow for their graphic representation.

The PCG is a graph that represents the acoustic signal produced by the heart system over time. Figure 5.10 shows some examples of normal and abnormal PCG sequences. They are not always distinguishable by the human ear, but they are always detectable by phonocardiogram.

Acoustically, the tones that characterize a normal heartbeat are defined as *S1* and *S2*. The first is generated by the closure of the atrioventricular (AV) valves and is characterized as a quiet, prolonged tone, lasting approximately 0.15 seconds. The second is determined by the closure of the semilunar valves and is characterized by a higher tone and duration of 0.12 seconds. Between *S2* and *S1* there is a long pause, representative of diastole, in which two other tones, *S3* and *S4*, can be perceived. The third tone is dull, weak and short, generated by the vibration due to the rapid filling of the ventricles at the beginning of diastole; it is often audible when the ventricle fails to empty completely with a consequent increase in the ventricular end-diastolic volume. The fourth tone is audible during atrial systole due to the contraction of the atria and the vibration of the ventricular walls given by the blood received; *S4* is often associated with swirling blood flow. The *S1* and *S2* components are in the range 20 - 200 Hz, while *S3* and *S4* in the frequency range between 15 - 65 Hz.

The *S3* and *S4* tones [56],[64] are abnormal sounds, which include heart murmurs, adventitious sounds, and galloping rhythms. Murmurs are intra-cardiac events observed in the frequency range between 20 - 600 Hz [56] and can be broadly classified according to their intensity over time (i.e. crescendo, decrescendo

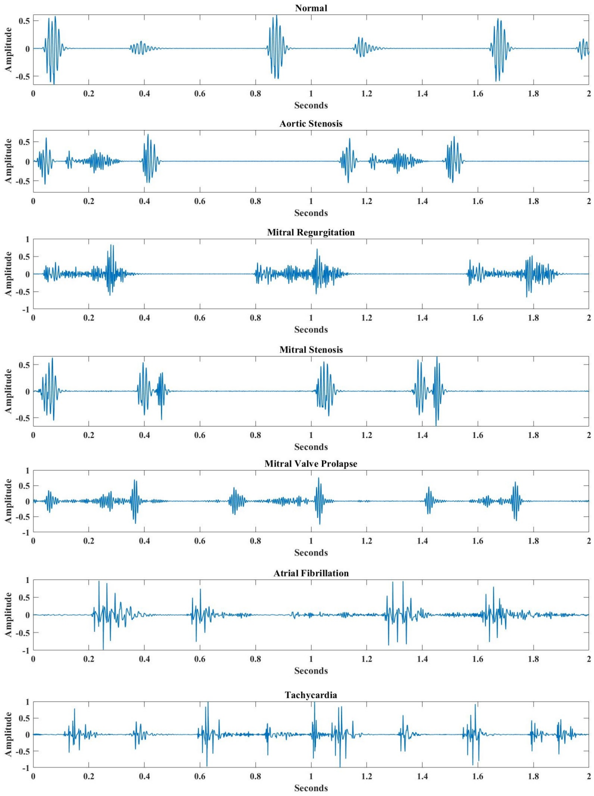


Figure 5.10: Acoustic representation of the normal and abnormal PCG signal.

or crescendo-decrescendo) and position in the cardiac cycle (early, medium or late or continuous.

5.2.2 Related Works

This section will provide an overview of the studies conducted over the past 10 years by multiple researchers to create cardiovascular disease detection and classification systems using PCG heart sound.

Many authors have proposed classification systems with two or more classes for the detection of heart disease using methods based on Multiclass Composite Classifier [66], Support Vector Machine (SVM), Deep Neural Network (DNN), K-Nearest Keighbors algorithm (KNN), Naïve Byes [199–203] and Feedforward Neural Network (FNN) and Convolutional Neural Network (CNN) [60],[204–209].

An open access dataset containing 1000 PCG sequences, approximately two seconds long, is proposed by the authors in [199]. The dataset is divided into a class containing normal beats and four different classes of cardiac pathologies (Mitral Regurgitation, Mitral Stenosis, Aortic Regurgitation and Mitral Valve Prolapse), each containing 200 PCG recordings. The method proposed by the authors for the recognition of heart sound in the 5 different classes consists in the application of three different algorithms for feature extraction: MFCC; DWT; MFCC and DWT. The performance obtained by applying the extracted parameters from the three different extraction methods to the three different machine learning algorithms for classification varies from 80.2% to 97.9%. In [66] the authors use Chirplet Transformation (CT) on the PCG signal to conduct an analysis based on the Time-Frequency (TF) domain. The database referred to in the study is the one proposed in [199], where only the "Mitral Valve Prolapse" class is neglected. PCG sequences are split into sub-sequences of approximately 0.7 seconds duration to which CT is, subsequently, applied. The matrix containing the TF parameters obtained from CT are used for the classification of cardiac pathologies (Mitral Regurgitation, Mitral Stenosis, Aortic Regurgitation) or normal beats. Entropy and local energy extraction functions are applied to the TF matrices, which are obtained from CT on each 0.7-second PCG segment. The data obtained for each matrix are entered as input to the Multiclass Composite Classifier network. With this approach, the authors have achieved classification performance between 96.2% and 99.6% with a low complex classifier.

Some authors conduct studies on PCG signal classification into only two classes, normal and abnormal, using different deep learning algorithms (SVM, KNN, and Naïve Byes) and non-open access datasets. In [200] twelve statistical parameters are used to extract features of PCG signals, the dataset is randomly split to create the raining and testing dataset. The accuracy of the results obtained by applying the four classifiers range from 94.44% to 97.78%, with an equal error rate between 5 and 2.22. In [201], PCG sequences are divided into sub-sequences of duration between 10 and 25 milliseconds. For each sub-sequence, the extraction of signal features is performed by time and frequency domain analysis and MFCC parameters. The obtained parameters are given as input to the two different neural network models. The accuracy, depending on the trained and tested neural network model, varies between 96% and 100%.

Further studies [202–205] apply convolutional neural networks to classify PCG signals into "normal" and "abnormal" class and use different feature extraction techniques. The PCG signals used in the papers are extracted from the same public database, PhysioNet/CinC Challenge 2016. The achieved performance is quite significant and ranges between 63% and 97.05%.

In [206], the authors propose a 5-class classification system using convolutional neural networks and the dataset from [199]. In particular, data augmentation technique is applied to expand the dataset from 1000 to 2000 PCG recordings and they do not use any kind of feature extraction technique. Classification is performed on two-second PCG segments and the dataset is randomly split into training and testing. The achieved accuracy performance is 98.6%.

The studies illustrated so far use different databases, both public and private, but they all apply the partitioning into training and testing datasets in a random manner. This very often leads to different recordings that could belong to the same patient in both datasets. This situation could lead to a false estimate of the percentage of accuracy obtained at the time of classification, since the heartbeat is related to the person. The testing dataset must, instead, have segments totally unknown to the neural network in order to correctly estimate the robustness and accuracy of the classification system.

5.2.3 Proposed Method

This section will describe the proposed method for the identification and classification of cardiac diseases affected by ambient noise. In particular, this section will outline the pre-processing, matching and decision techniques for processing and classifying the data.

During this phase, the PCG input signal is analyzed according to a raw waveform block analysis lasting $W_{in} = 2$ seconds which, once re-sampled at 22,050 Hz and normalized in power to -20 dBFS, are fed to the CNN neural network (Appendix C). The reference power value chosen for normalization is set by calculating the average power value of all the training sequences for each class and computing a global average of all the classes.

The matching phase is based on the convolutional neural network described in section V and suitably trained with the training database. Every W_{in} seconds the CNN network directly receives a raw data vector that contains the amplitudes of the PCG signal over time. The CNN decides which of the N_C classes C_i the input vector belongs to.

In order to make the method more robust and increase accuracy, even in the presence of any ambient noises picked up by the electronic stethoscope, in the decision block, a filter called "recurrence filter" allows to improve performance by analyzing N_o successive decisions of the CNN network, with the purpose of returning class $C_{out}(K)$ as final output at k-th iteration. More specifically, the filter acts on a circular vector $\vec{C_K}$, which always contains the latest N_o decisions, after a transition phase lasting $W_{in}(N_o - 1)$ seconds.

The most common among the classes in buffer $\vec{C_K}$ is selected according to the criterion in 5.1:

$$C_{out}(K) = ArgMax[Hist(\vec{C_K}, N_C)] \tag{5.1}$$

where the *Hist* function creates a histogram bar chart of the elements in the vector $\vec{C_K}$ sorted into N_C equally spaced bins and the *ArgMax* function returns the C_i index of the recurrence maximum value along the vector $\vec{C_K}$. When in $\vec{C_K}$ two or more classes have the same maximum number of repetitions, $C_{out}(K)$ is the last C_i class provided by the matching block. The method then makes the first decision on the type of heart condition after $K = N_o$ iterations, when the circular buffer fills up, i.e. after a time equal to the analysis window lasting $W_A = W_{in}N_o$ seconds.

Using only G-Y Son et al. Public Dataset

The public database [199] used in this paper contains $N_c = 5$ different classes C_i of PCG recordings, each of $W_{in} = 2$ seconds in duration and sampled at a frequency of 8000 Hz, with the following numbers: Normal (N): 200 recordings; Aortic stenosis (AS): 200 recordings; Mitral regurgitation (MR): 184 recordings; Mitral stenosis (MS): 186 recordings; Mitral valve prolapses (MVP): 181 recordings. Each class was divided into a learning database for 70% and a testing database for the remaining 30% following two different criteria: (a) separating in sequential order a first group equal to 70% of the records from a second group equal to the remaining 30%. In this case, testing not only contains different recordings from learning, but the subjects are, also, different between testing and learning; (b) obtaining the testing database by choosing randomly, for each class, the sequences from the complete database of that class until reaching a number equal to 30%. This method does not offer any guarantee that in testing there are no registrations belonging to the same subject, although with other registrations, even in learning the same issue persists. The method (a), although it offers lower performance, as we will see from the results, seems to us the most correct one given the real functioning of an automatic cardiac pathology identification system occurs with subjects who, obviously, did not take part in the design of the system and took the test for the first time.

In order to verify the robustness against environmental noises, an additional type (a) testing databases were created digitally adding ambient noise sequences typical of office environments [209], normalized to -40 and -35 decibels relative to full scale (dBFS) [210], thus obtaining respectively a SNR of 20 dB and 15 dB.

The learning set is in turn divided into a 70% training set and a 30% validation set. The training set is used to train the neural network, while the validation set is used to validate it, calculating the accuracy and loss as the epochs increase.

G-Y Son et al. Public and Physionet Challenge 2016 - Combined Dataset

The database of PCG recordings used for convolutional neural network training and testing is given by the union of two databases publicly available and widely used in the literature: (a) the dataset presented by Y. G-Y Son et al. in [199] and (b) Physionet Challenge 2016 [207, 208]. In particular, the first dataset was considered in its entirety and includes the Normal (N) class and the following 4 cardiac pathologies: Aortic Stenosis (AS), Mitral Regurgitation (MR), Mitral Stenosis (MS) and Miltral Valve Prolapse (MVP) to which the Atrial Fibrillation (AF) and Tachycardia (T) classes appropriately extracted from the second dataset were added. To standardize the characteristics of the two datasets, all sequences were resampled at 22050 Hz. Moreover, taking into account the opportunity to have normalized sequences at the system input, the PCG signal has been normalized to a fixed power value. This power value is set by calculating the average power value of all the sequences for each class and taking a global average of all the classes. This value is expressed in decibels relative to full scale (dBFS) and in the present study it is -20 dBFS.

In order to strengthen the proposed classification system, “office” type ambient noise [209] was digitally added to the non-noisy PCG sequences, normalized in power to -30, -40, -50 dBFS in order to have a SNR of 10, 15, 20 and 30 dB, respectively. Once the noisy PCG sequences with the various SNR values were obtained, a further normalization to -20 dBFS in power was done. In order to evaluate the effect of different ambient noise, an equivalent PCG database was also made by adding “babble” noise [211] with SNRs of 25 and 30 dB. A total of 6 databases were obtained one of which was named “clean”, i.e. without the addition of noise, 3 with “office” noise and two with “babble” noise in the background.

The databases have been subdivided in a part equal to 70% for the Learning (L-DB) and a remaining part equal to 30% for the Testing (T-DB), making so that in the two datasets there were no sequences recorded by the same subject, so as to obtain more realistic performances, when the network is tested with a testing dataset completely unknown to it.

The Learning database without noise was denoted L_C (Clean Learning Dataset), while, by adding to this database examples of the Learning database with “office” noise in the breakdown 70% Clean, 10% SNR 10 dB, 10% SNR 20 dB, 10% SNR 30 dB a second Learning database with both clean and noisy sequences named L_N (Noisy Learning DataBase) was obtained.

Table 5.3: PCG dataset structure.

Class	L_C	L_N	DB-T
Normal	140	182	60
Aortic Stenosis	140	182	60
Mitral Regurgitation	124	163	60
Mitral Stenosis	127	166	59
Mitral Valve Prolapse	131	173	50
Atrial Fibrillation	56	74	28
Tachycardia	56	74	28

Finally, by subdividing the PCG sequences into several sub-sequences of Win duration equal to two seconds, the numerical values for each class of pathology were obtained (Table 5.3). Table 5.3 shows the number of 2-second PCG samples related to each class in the different datasets.

5.2.4 Experimental Results

Results Using only G-Y Son et al. Public Dataset

In this section, the results of training and subsequent testing of the neural network are presented and discussed.

In order to evaluate the performance of the CNN network with PCG sequences external to the training dataset, the accuracy obtained with the "Testing set" is assessed. Figure 5.11 shows the relative confusion matrix, in the case of the testing set obtained according to criterion (a). Considering that the testing database contains sequences of people totally unknown to the network, the CNN network classifies 2-second long PCG recordings, guaranteeing an average classification accuracy level of 89.6%. The performance obtained with a testing set generated according to criterion (b), provides a better accuracy of about 7 percentage points, but does not properly represent the real application case in which testing dataset certainly contains recordings belonging to people other than learning dataset.

By applying the recurrence filter it is possible to obtain 100% accuracy in both types of testing set (a) and (b), as illustrated in Figure 5.12. When testing according to criterion (a), the recurrence filter in the post-processing phase guarantees 100% performance, with an analysis window of at least 34 seconds (blue line), compared to 6 seconds (red line) when using a testing set (b). The presence of PCG recordings, also different for the subjects, leads to an initial reduction in performance of about 7 percentage points, a difference that with the increase of the analysis window gradually becomes zero after 34 seconds, a time abso-

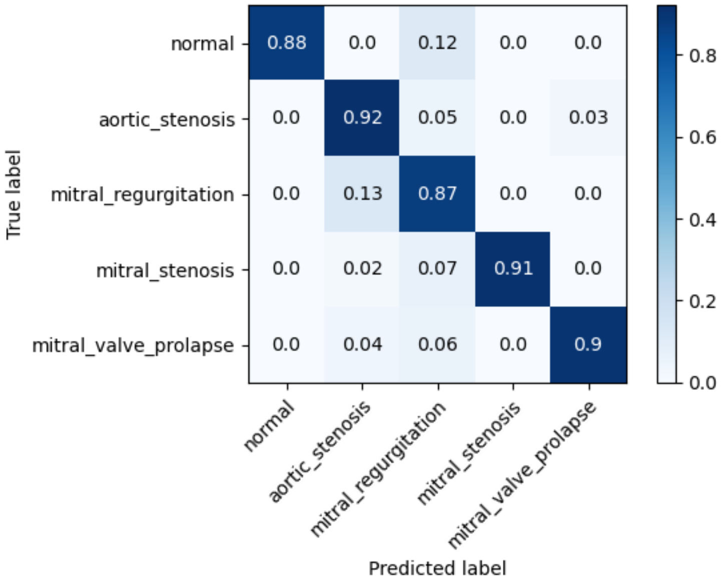


Figure 5.11: Confusion matrix - G-Y Son et al. public dataset.

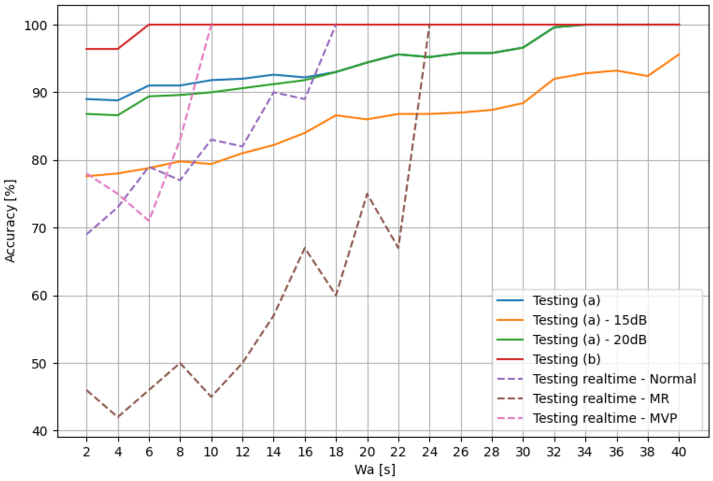


Figure 5.12: Accuracy as the analysis window changes - G-Y Son et al. public dataset.

lutely compatible with the application context. Finally, as regards the robustness against environmental noise, more specifically office noise [209], Figure 5.12 shows that, for $SNR > 20dB$, the impact of noise on accuracy is negligible (green line), while for $SNR = 15dB$ there is a reduction of 11%, which, however, gradually decreases as the observation window increases (orange line). In general, it is possible to increase the robustness to ambient noise if the auscultation time of the cardiac signal is increased.

In order to verify the behavior of the system with real-time testing, Normal, Mitral Regurgitation and Mitral Valve Prolapse PCG sequences (Figure 5.13)

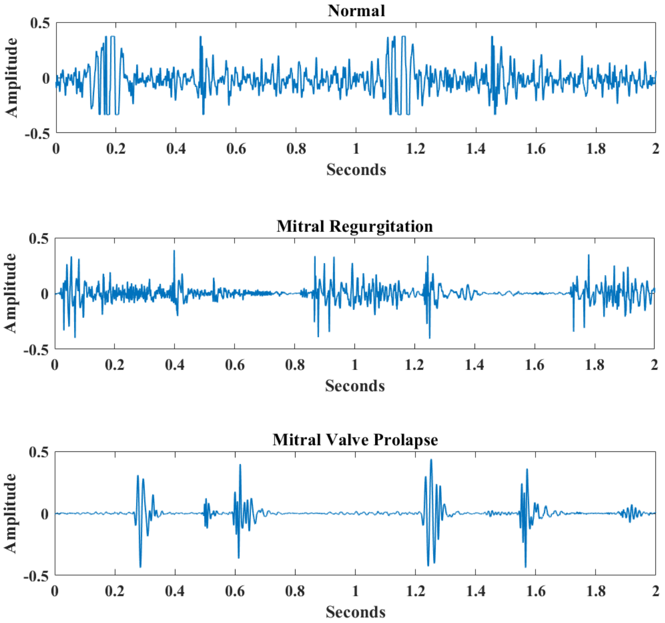


Figure 5.13: Normal and abnormal real-time heart sounds.

were recorded with a Thinklabs Rhythm Digital Electronic Stethoscope model in a hospital setting. The results in terms of accuracy are shown in Figure 5.12 (dashed lines). The accuracy is low with reduced analysis window values, it reaches 100% for analysis window values between 10 and 24 seconds, i.e. values compatible with the application context.

Results with G-Y Son et al. Public and Physionet Challenge 2016 - Combined Dataset

This section will present the results obtained in the testing phase of the proposed method. The first part of the results concerns the results obtained with the CNN neural network training based on the L_C dataset, while the second part concerns the results obtained with the mixed L_N dataset. In both cases, the testing datasets used for the performance evaluations are represented by the "clean" case (T-C), "office" with SNR=15 dB (T_O-N SNR 15dB), and "babble" with SNR=25dB (T_B-N SNR 25dB) and with SNR=30 dB (T_B-N SNR 30dB).

Figure 5.14 and Figure 5.15 represent, respectively, the confusion matrix in the case of testing "clean" sequences fed to the CNN trained with the L_C dataset and with the L_N dataset. It can be observed from the comparison of the two figures that the degree of misclassification is very similar and this is due to the right compromise imposed on the L_N between clean and noisy training sequences,

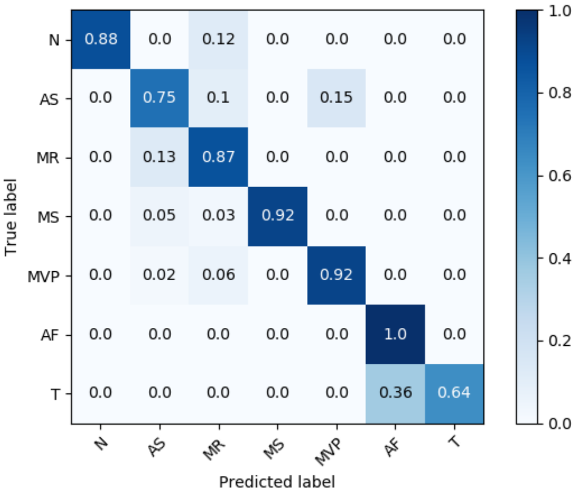


Figure 5.14: Confusion matrix with L_C and T-C - G-Y Son et al. and Physionet challenge 2016 dataset.

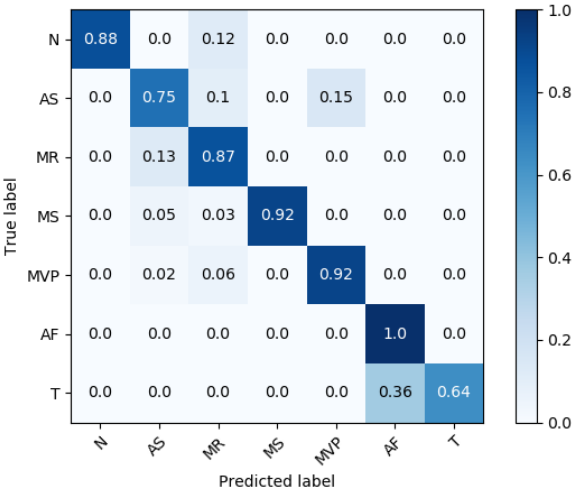


Figure 5.15: Confusion matrix with L_N and T-C - G-Y Son et al. and Physionet challenge 2016 dataset.

obtained respectively with the percentages of 70% (clean) and 30% (noisy).

The graphs in Figure 5.16 show the accuracy as the W_A analysis window varies. Specifically, the solid curves are pertinent to the clean learning dataset (L_C), while the dashed curves are pertinent to the second noisy learning dataset (L_N).

In general, it must be noted that for both trainings the level of accuracy ranges between 80% and 100%, depending on the width of the analysis window.

The tests carried out on the network trained with the mixed learning database (L_N) result in better performance than the tests obtained with the training given by the clean dataset (L_C). In fact, it was observed that apart from the curve relative to testing "clean" sequences for which we have however comparable

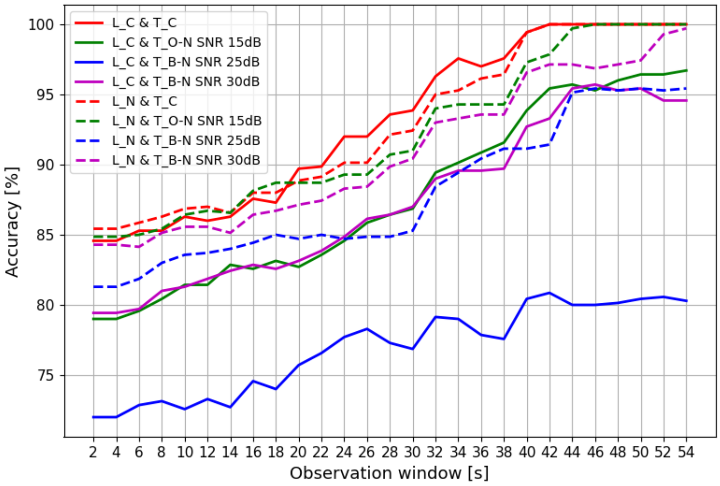


Figure 5.16: Accuracy as W_A analysis window changes - G-Y Son et al. and Physionet challenge 2016 dataset.

performances, all the dashed curves show superior values than the respective continuous curves. Training with examples of noisy PCG sequences allows ensuring greater robustness to ambient noise obtaining an average increase in accuracy of 5%.

From the graph we can see that in order to obtain 100% accuracy we need a W_a analysis window of about 42 seconds in the clean case (dashed red line) and 46 seconds in the case of "office" noise up to 15 dB SNR (dashed green line). The system is not only robust to the same type of noise used for the learning phase of the CNN network but, as can be seen from the figure (blue and purple lines), it is also robust to other types of noise such as "babble" noise that could be present during an automatic heart disease recognition test through the use of an electronic stethoscope.

Table 5.4: Accuracy, TPR, TNR, FPR and $F_1 Score$ comparison values as the Learning/Testing dataset and W_A change.

DB-L	Evaluation's Metrics [%]	DB-T			
		T_C	T_O-N SNR 15dB	T_B-N SNR 25dB	T_B-N SNR 30dB
L_C	A	[84.6, 100.0]	[79.0, 97.4]	[72.0, 80.3]	[79.4, 94.6]
	TPR	[86.8, 100.0]	[83.3, 97.4]	[72.0, 80.3]	[81.5, 94.6]
	TNR	[86.7, 100.0]	[84.0, 97.1]	[74.0, 81.0]	[82.2, 94.0]
	FPR	[0, 13.3]	[2.9, 16.0]	[19.0, 26.0]	[6.0, 17.8]
	F_1 Score	[83.8, 100.0]	[78.0, 97.4]	[68.5, 80.3]	[79.2, 94.6]
L_N	A	[85.4, 100.0]	[84.9, 100.0]	[81.3, 95.4]	[84.3, 99.7]
	TPR	[86.5, 100.0]	[87.1, 100.0]	[81.3, 95.4]	[84.3, 99.7]
	TNR	[83.5, 100.0]	[86.8, 100.0]	[81.6, 94.8]	[84.1, 99.7]
	FPR	[0, 16.6]	[0, 13.2]	[5.2, 18.4]	[0.3, 15.9]
	F_1 Score	[85.9, 100.0]	[85.2, 100.0]	[80.7, 95.4]	[84.3, 99.7]

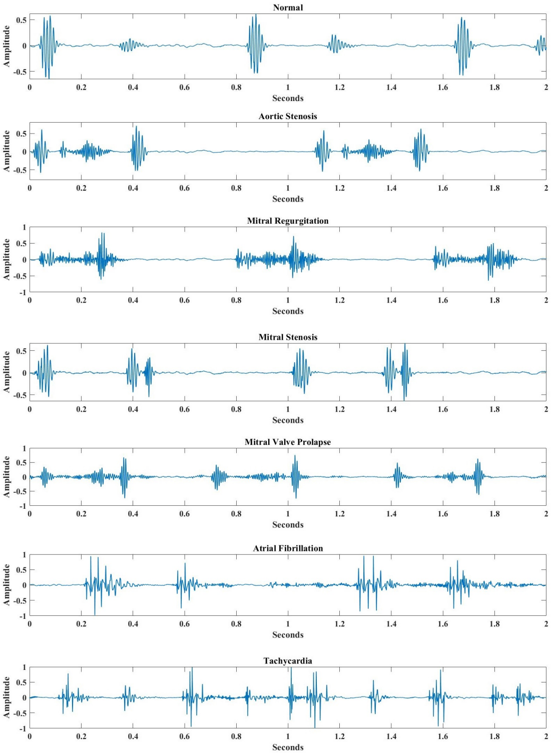


Figure 5.17: Acoustic representation of PCG signals in the presence of office noise with SNR equal to 15 dB.

In order to evaluate the performance of the system trained with the two datasets, Table 5.4 introduces the metrics of evaluation of the confusion matrix related to each of the tests performed starting from an analysis window of 2 seconds up to 54 seconds (details in Appendix F).

In general, it is possible to obtain 100% accuracy in the presence of ambient noise by increasing the analysis window to values that generally do not exceed 60 seconds, i.e. time compatible with an application context such as telemedicine. Figure 5.17 outlines a graphical feedback in the case of office noise with SNR equal to 15 dB.

5.2.5 Discussion and Conclusion

Using only G-Y Son et al. Public Dataset

Table 5.5 shows the comparisons with the two recent scientific publications [66, 199] based on the same database. All the performance indices and the relative performances in terms of accuracy, $F_1 Score$, sensitivity and specificity [199], clearly highlight the superior performance of the proposed method despite using raw data in input to the CNN network which, thereby, succeeds in perfectly grasping

Table 5.5: Method and performance comparison PCG 5 classes.

Authors	Method	N_C	Analysis Window [s]	Feature Extraction	Testing Dataset	Accuracy [%]	Sensitivity [%]	Specificity [%]	$F_1 Score$ [%]
[199]	SVM DNN KNN	5	2	MFCC DWT MFCC + DWT	(b) Different records (20 %)	[80.2, 97.9]	[82.0, 98.2]	[93.5, 99.4]	[86.0, 99.7]
[66]	Multiclass composite classifier	4	0.7	Chirplet Transf Time-frequency analysis	(b) Different records (30%)	[96.2, 99.6]	[91.7, 100]	[97.8, 100]	[94.9, 100]
[60]	Raw Data CNN-based classifier	5	[2, 6]	-	(b) Different records (30%)	[96.6, 100]	[96.8, 100]	[96.8, 100]	[96.9, 100]
			[2, 34]		(a) Different subjects/ records (30%)	[89.6, 100]	[90.8, 100]	[89.6, 100]	[89.9, 100]

Table 5.6: Method and performance comparison PCG 7 classes.

Authors	Method	Nc	Analysis Window [s]	Feature Extraction	Testing Dataset	Noisy Rec	Accuracy [%]
[199]	SVM DNN KNN	5	2	MFCC DWT MFCC + DWT	(b) Different records (20 %)	No	[80.2, 97.9]
[66]	Multiclass composite classifier	4	0.7	Chirplet Transf Time-frequency analysis	(b) Different records (30%)	No	[96.2, 99.6]
[208]	CNN	5	2	-	G-Y Son et al. DB + data Augmentation	No	99
[60]	Raw Data CNN-based classifier	5	[2, 6]	-	Different records (30%)	No	[96.6, 100]
			[2, 34]		Different subjects/ records (30%)		[89.6, 100]
[173]	Raw Data CNN-based Classifier	7	[2, 42]	-	Different subjects/ records (30%)	No	[84, 100]
			[2, 42]		Different subject/ records (30%)	Yes	[72, 85]

the differences between the characteristics of the 5 classes taken into consideration in the time domain. This study proposes an automated technique for cardiac disease diagnosis based on the PCG audio signal obtained from an electronic stethoscope. The technique does not use domain transforms, thus keeping computational complexity to a minimum by directly processing the raw data recorded by the stethoscope. Thanks to the intrinsic ability of CNN networks to capture the differences in the signals in the various classes and to the recurrence filter proposed, the new heart sound disease recognition method allows to achieve 100% accuracy in less than 34 seconds, even in the presence of ambient office noise. That is compatible with the application contexts to which the proposed method is directed.

G-Y Son et al. Public and Physionet Challenge 2016 - Combined Dataset

Table 5.6 compares recent scientific publications [66],[199], from the perspective of methods and results, that are based on the same database.

Our method stands out from its predecessors by three factors:

1. Increased number of classes of heart disease for classification;
2. Use of testing PCG signals, totally unknown to the neural network. So, PCG signals in the testing dataset are related to different people and recordings;
3. Addition of two different types of ambient noise (office noise and babble), with different SNR, in both the learning and testing dataset sequences. This makes the system much more robust to interference.

From the point of view of classification performance, the proposed method offers excellent results regardless of the type of learning and testing dataset used.

Chapter 6

Conclusion

For the development of more sustainable and resilient cities, emergencies should be treated as of the main elements of the urban dynamics, potentially affecting multiple systems in a city environment. Actually, with most world population living in urban areas, with increasingly presence of large and mega cities, the negative impacts of emergencies and disasters have been more significant in the last decades. This scenario has fostered the adoption of different emergency-prone approaches in multiple contexts, but the challenges imposed by overcrowded cities will still demand more efficient solutions.

In this thesis, three main issues were addressed in relation to environmental and health risk management in smart cities.

The first topic focuses on the possibility of using drone-femtocell systems to locate devices under rubble in post-earthquake scenarios. Assuming that the localisation of each individual device leads to the rescue of a human life, the studies presented aim to devise algorithms with very low localisation error and very high drone energy efficiency.

Considering studies conducted on this topic, described in chapter 3, our algorithms implemented for each phase (mobile terminal classification and localisation), in typical conditions of a real application scenario, achieved accuracies greater than 60% with an average error in estimating the position of about 1 meter. On the other hand, with regard to the study for the optimisation of flight time and energy of drones through GT, it can be concluded that through the proposed method there is a reduction of the overall flight time, and therefore of energy consumption, which, on average, ranges from 25% to 66%. These values depend on the technology adopted and/or the spatial resolution to be obtained in the localization process.

This performance entails a further increase of about 10%, bearing in mind that application scenarios are characterized by non isotropic propagation of radio frequency signals. The studies conducted are innovative in their field, as in the state of the art few researches deal with this topic; mainly because the scenario is very complex. For this reason, the tests conducted and the results obtained are very promising.

The second topic involves the detection and classification of rainfall levels using different types of signals (audio, video, radio) and deep learning techniques. This approach predicts and manages hydrogeological risk situations (landslides, floods and inundations) in cities in time, and thus also in terms of road safety and much more. Three different approaches for the classification of rainfall levels using audio, video and radio signals, respectively, were addressed in the chapter 4.

Regarding the classification of rainfall by audio signal, the proposed study presents a technique that allows classifying the different levels of rainfall intensity accurately and quickly. An analysis of the different statistical and spectral characteristics of the acoustics produced by the rain at various rainfall levels is presented. In particular, the system is very simple, being based on a plastic shaker, a microphone, and a low-cost/low-power signal processing unit. The performance is very good in terms of accuracy and ability to adapt to sudden changes in precipitation intensity. It should be borne in mind that, especially in low rainfall, the peak rain indicator used to label the database has a low temporal resolution. Thus, taking into account the typical micro-variances of rainfall intensity, it is possible to consider an average accuracy of 93%, assuming that overall system performance does not include mis-classification between adjacent classes. In general, the proposed solution is adequate for a precipitation level monitoring service, with its major advantage being that it is an entirely electronic system that can be easily integrated on existing platforms and systems.

Regarding the classification of rainfall by video signal, the proposed study is an alternative to traditional rain gauges which is based on the extraction of differential frames from video sequences. Rainfall levels are classified by applying a convolutional neural network. Performance is very good in terms of precision and ability to adapt to sudden changes in rainfall intensity. The percentage of accuracy of the average classification obtained by applying the DCT for to 16×16 sub blocks is approximately 49%, which can reach 75% if the adjacent mis-classifications are not considered. The audio and video rain gauge exceeds the limits of traditional ones, as it requires no mechanical parts, specific and periodic maintenance.

Regarding the classification of rainfall by radio signal, the proposed study introduces an innovative approach to rainfall classification for smart city applications.

The main idea is based on the impact that rain has on a set of parameters that characterize the radio-mobile channel quality. The approach highlights the link between rainfall levels and the trend of parameters adopted for the cell selection phase in the LTE mobile network. In particular, the system requires only the extraction of the parameters that a UE measures and the subsequent comparison with a non-linear matching system based on MLP networks. The performance is very good in terms of accuracy and spatial resolution. Taking into account the typical micro-variances of rainfall intensity, it is possible to consider an average accuracy of 96%. Also in this case, the rain gauge exceeds the limits of the traditional ones, as it has no mechanical parts and requires no maintenance. Compared to the state of the art, the study described in this thesis exploits radio frequencies typical of telecommunication networks (1.8 MHz and 2.6 GHz) to assess the impact of rain on radio parameters and thus to classify it.

For all studies, ad hoc databases were constructed. These contain recordings of audio/video/radio signals taken on different days and at different locations. The third topic concerns healthcare management within smart cities, in particular, the possibility of defining innovative algorithms for the detection and classification (in very short times and with high precision) of heart disease using ECG and PCG signals in conjunction with deep learning techniques.

Public datasets were used in the studies regarding the classification of heart disease using ECG signals. The classification results obtained are highly accurate, approximately 98-100% by applying a post-processing filter. The innovations in these works, compared to the state of the art, are many: excellent recognition and classification of more than three disease classes; use of RAW signals as input to the neural network without the use of signal pre-processing systems; excellent performance in short times (2-38 seconds). Also in this case, for the classification of heart disease using the PCG signal, public datasets were used. In this thesis, two kinds of studies are conducted for the classification of heart diseases and the average results obtained are very high, approximately 84-100% using clean PCG sound and 72-85% using PCG sound with ambiental noise (office and babble noises). In both cases the maximum performance is obtained by applying a post-processing filter. In these studies RAW signals were used as input to the neural network, without the use of signal pre-processing systems and in order to obtain the maximum classification accuracy, it is necessary to wait a maximum of 34 seconds in clean PCG signal cases and 42 seconds in noises PCG signal cases. These timelines fit perfectly into the contexts of Telemedicine and health monitoring in smart cities.

List of journal publications

1. R. Avanzato, F. Beritelli and C. Rametta, “Enhancing Perceptual Experience of Video Quality in Drone Communications by Using VPN Bonding,” In IEEE Embedded Systems Letters, 2022, doi: 10.1109/LES.2022.3182466.

2. R. Avanzato and F. Beritelli, “Heart Sound Multiclass Analysis Based on Raw Data and Convolutional Neural Network,” in IEEE Sensors Letters, vol. 4, no. 12, pp. 1-4, Dec. 2020, Art no. 7004104, doi: 10.1109/LSENS.2020.3039366.

3. R. Avanzato, and F. Beritelli, “A CNN-based Differential Image Processing Approach for Rainfall Classification,” In Advances in Science Technology and Engineering Systems Journal, vol. 5, no. 4, pp. 438-444, 9 August 2020, doi: http://dx.doi.org/10.25046/aj050452.

4. R. Avanzato and F. Beritelli, “Hydrogeological Risk Management in Smart Cities: A New Approach to Rainfall Classification Based on LTE Cell Selection Parameters,” in IEEE Access, vol. 8, pp. 137161-137173, 2020, doi: 10.1109/ACCESS.2020.3011375.

5. R. Avanzato, F. Beritelli, A. Raspanti, and M. Russo, “Assessment of Multimodal Rainfall Classification Systems Based on an Audio/Video Dataset,” In International Journal on Advanced Science, Engineering and Information Technology, vol. 10, no. 12, pp. 1163-1168, 2020. DOI: 10.18517/ijaseit.10.3.12130.

6. R. Avanzato, and F. Beritelli, “Automatic ECG Diagnosis Using Convolutional Neural Network,” In MDPI Electronics – Open Access Journal, vol. 9, no. 6, 8 June 2020. doi: https://doi.org/10.3390/electronics9060951.

7. R. Avanzato, and F. Beritelli, An Innovative Acoustic Rain Gauge Based on Convolutional Neural Networks,” In MDPI Information – Open Access

Journal, vol. 11, no. 4, 28 March 2020. doi: https://doi.org/10.3390/info11040183.

8. R. Avanzato, F. Beritelli, F. Di Franco, and M. Russo, "SMILE: Smart Monitoring IoT Learning Ecosystems," In International Journal on Advanced Science, Engineering and Information Technology, vol. 10, no. 1, pp. 413-419, 2020. doi: 10.18517/ijaseit.10.1.11144.

9. R. Avanzato and F. Beritelli, "A Smart UAV-Femtocell Data Sensing System for Post-Earthquake Localization of People," in IEEE Access, vol. 8, pp. 30262-30270, 2020, doi: 10.1109/ACCESS.2020.2972699.

10. R. Avanzato, F. Beritelli, and M. Vaccaro, "An Innovative Technique for Identification of Missing Persons in Natural Disaster based on Drone-Femtocell Systems," in Special Issue UAV-Based Applications in the Internet of Things (IoT), In MDPI Sensors – Open Access Journal, vol. 19, no. 20, 4547, 19 October 2019. doi: https://doi.org/10.3390/s19204547.

List of conferences publications

1. M. Russo, V. F. Puglisi, R. Avanzato and F. Beritelli, "A CNN-based Audio Sensor for Rainfall Estimation: Implementation on Embedded Board," 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021, pp. 911-915, doi: `10.1109/IDAACS53288.2021.9660891`.

2. M. Russo, R. Avanzato and F. Beritelli, "PER-COVID (PEople pRoximity based on Certified and coOperative VIDeo-intelligence): A Software Tool for Physical Distancing and PPE Monitoring," 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021, pp. 881-885, doi: `10.1109/IDAACS53288.2021.9660949`.

3. R. Avanzato, F. Beritelli, G. Capizzi, C. Napoli and C. Rametta, "Soil Moisture Estimation based on the RSSI of RFID Modules," 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021, pp. 647-652, doi: `10.1109/IDAACS53288.2021.9661031`.

4. R. Avanzato, F. Beritelli, F. Raciti and E. Spataro, "Optimization of UAV-Femtocell Systems Positioning via Game Theory to Geolocate Mobile Terminals in a Post-Earthquake Scenario," 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021, pp. 785-790, doi: `10.1109/IDAACS53288.2021.9660873`.

5. R. Avanzato, F. Beritelli, C. Rametta and M. Russo, "Exploiting VPN Bonding for Time Critical Video Transmission on Board Drone," 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021, pp. 808-812, doi: `10.1109/IDAACS53288.2021.9660387`.

6. M. Russo, G. Mirulla, R. Avanzato and F. Beritelli, “Car Social Network: Contact a Driver Through the License Plate,” 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2021, pp. 924-927, doi: 10.1109/IDAACS53288.2021.9660976.

7. F. Mazzola et al., “Taormina: A LoRa-based localization scheme for smart road scenarios,” 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2021, pp. 1-6, doi: 10.1109/ICECCME52200.2021.9591156.

8. R. Avanzato, F. Beritelli and G. Nicotra, "Flight time optimization in people identification by multidrone-femtocell systems," In Proc. on CEUR Workshop Proceedings, Symposium for Young Scientists in Technology, Engineering and Mathematics, Online, 20 May 2020.

9. S. Russo, S. I. Illari, R. Avanzato and C. Napoli, "Reducing the psychological burden of isolated oncological patients by means of decision trees," In Proc. on CEUR Workshop Proceedings, Symposium for Young Scientists in Technology, Engineering and Mathematics, Online, 20 May 2020.

10. R. Avanzato, F. Beritelli, M. Russo, M. Vaccaro, "YOLOv3-based mask and face recognition algorithm for individual protection applications," In Proc. on CEUR Workshop Proceedings, Symposium for Young Scientists in Technology, Engineering and Mathematics, Online, 20 May 2020.

11. R. Avanzato and G. Nicotra, "An efficient HDL IP-core Generator for OFDM modulators," In Proc. on CEUR Workshop Proceedings, Symposium for Young Scientists in Technology, Engineering and Mathematics, Online, 20 May 2020.

12. S. I. Illari, S. Russo, R. Avanzato and C. Napoli, "A cloud-oriented architecture for the remote assessment and follow-up of hospitalized patients," In Proc. on CEUR Workshop Proceedings, Symposium for Young Scientists in Technology, Engineering and Mathematics, Online, 20 May 2020.

List of accepted and submitted paper

1. R. Avanzato and F. Beritelli, “Heart disease recognition based on extended ECG sequence database and deep learning techniques,” Accepted for publication to 2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS).

2. R. Avanzato, F. Beritelli, and V.F. Puglisi, “Dairy Cow Behavior Recognition Using Computer Vision Techniques and CNN Networks,” Accepted for publication to 2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS).

3. R. Avanzato, F. Beritelli, F. Raciti, and E. Spataro, “Energy Management Optimization of UAV-Femtocell Geolocalization Systems based on Game Theory,” Accepted for publication to 2022 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS).

4. R. Avanzato, F. Beritelli, and C. Rametta, “An DL-based Approach for Packet Error Compensation Using Radio Mobile Network Quality Parameters in a Rainfall Scenario,” Accepted for publication to Computer Networks, Elsevier, Available at SSRN: `https://ssrn.com/abstract=4088414`.

5. R. Avanzato, F. Beritelli, G. Capizzi, C. Napoli, and R. Scherer, “Robust Recognition of Cardiac Pathologies Based on PCG Analysis and Deep Learning,” submitted to Pattern Recognition, Elsevier.

6. R. Avanzato, F. Beritelli, F. Raciti, and E. Spataro, “Game Theory-based Energy Efficiency Optimization of UAV-Femtocell Geolocalization Systems,” submitted to IEEE Transactions on Aerospace and Electronic Systems.

References

[1] Ayca Kirimtat et al. “Future trends and current state of smart city concepts: A survey”. In: *IEEE access* 8 (2020), pp. 86448–86467.

[2] Ruben Sánchez-Corcuera et al. “Smart cities survey: Technologies, application domains and challenges for the cities of the future”. In: *International Journal of Distributed Sensor Networks* 15.6 (2019), p. 1550147719853984.

[3] Ammar Gharaibeh et al. “Smart cities: A survey on data management, security, and enabling technologies”. In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2456–2501.

[4] Daniel G Costa and Felipe P de Oliveira. “A prioritization approach for optimization of multiple concurrent sensing applications in smart cities”. In: *Future Generation Computer Systems* 108 (2020), pp. 228–243.

[5] Harish Kumar et al. “Moving towards smart cities: Solutions that lead to the Smart City Transformation Framework”. In: *Technological forecasting and social change* 153 (2020), p. 119281.

[6] Mohd Abdul Ahad et al. “Enabling technologies and sustainable smart cities”. In: *Sustainable cities and society* 61 (2020), p. 102301.

[7] Maycon Leone Maciel Peixoto et al. “A traffic data clustering framework based on fog computing for VANETs”. In: *Vehicular Communications* 31 (2021), p. 100370.

[8] Safa Ben Atitallah et al. “Leveraging Deep Learning and IoT big data analytics to support the smart cities development: Review and future directions”. In: *Computer Science Review* 38 (2020), p. 100303.

[9] Constantine E Kontokosta and Awais Malik. “The Resilience to Emergencies and Disasters Index: Applying big data to benchmark and validate neighborhood resilience capacity”. In: *Sustainable cities and society* 36 (2018), pp. 272–285.

[10] Mark Tebeau. *Eating smoke: Fire in urban America, 1800–1950.* JHU Press, 2012.

[11] Daniel G Costa et al. “A Survey of Emergencies Management Systems in Smart Cities”. In: *IEEE Access* (2022).

[12] Samir Rana, Lisa Gopal, and Neha Gupta. “Smart City Concepts, Features and the Role of Internet of Things: A Review”. In: *2021 International Conference on Computational Performance Evaluation (ComPE)*. 2021, pp. 585–591. DOI: 10.1109/ComPE53109.2021.9752040.

[13] Kaya Kuru and Darren Ansell. “TCitySmartF: A comprehensive systematic framework for transforming cities into smart cities”. In: *IEEE Access* 8 (2020), pp. 18615–18644.

[14] Ebru Tekin Bilbil. “The operationalizing aspects of smart cities: The case of Turkey’s smart strategies”. In: *Journal of the Knowledge Economy* 8.3 (2017), pp. 1032–1048.

[15] Vahid Javidroozi, Hanifa Shah, and Gerald Feldman. “Urban computing and smart cities: Towards changing city processes by applying enterprise systems integration practices”. In: *IEEE Access* 7 (2019), pp. 108023–108034.

[16] Daniel G Costa et al. “On the use of cameras for the detection of critical events in sensors-based emergency alerting systems”. In: *Journal of Sensor and Actuator Networks* 9.4 (2020), p. 46.

[17] Izabela Rojek and Jan Studzinski. “Detection and localization of water leaks in water nets supported by an ICT system with artificial intelligence methods as a way forward for smart cities”. In: *Sustainability* 11.2 (2019), p. 518.

[18] Hafiz Suliman Munawar et al. “Disruptive technologies as a solution for disaster risk management: A review”. In: *Science of the total environment* 806 (2022), p. 151351.

[19] Paolo Gasparini, Angela Di Ruocco, and Raffaella Russo. “Natural hazards impacting on future cities”. In: *Resilience and Sustainability in Relation to Natural Disasters: A Challenge for Future Cities*. Springer, 2014, pp. 67–76.

[20] Timothy Sim, Dongming Wang, and Ziqiang Han. “Assessing the disaster resilience of megacities: The case of Hong Kong”. In: *Sustainability* 10.4 (2018), p. 1137.

[21] Zakria Qadir et al. “Addressing disasters in smart cities through UAVs path planning and 5G communications: A systematic review”. In: *Computer Communications* 168 (2021), pp. 114–135.

[22] Donghyun Son et al. “Effects of partial infrastructure on indoor positioning for emergency rescue evacuation support system”. In: *Proceedings of the First CoNEXT Workshop on ICT Tools for Emergency Networks and DisastEr Relief*. 2017, pp. 13–17.

[23] Dimitri Tassetto, Eriza Hafid Fazli, and Markus Werner. “A novel hybrid algorithm for passive localization of victims in emergency situations”. In: *International Journal of Satellite Communications and Networking* 29.5 (2011), pp. 461–478.

[24] Stefan Zorn et al. “A novel technique for mobile phone localization for search and rescue applications”. In: *2010 International Conference on Indoor Positioning and Indoor Navigation*. 2010, pp. 1–4. DOI: 10.1109/IPIN.2010.5647107.

[25] Zhe Zhang et al. “Finding Disaster Victims: A Sensory System for Robot-Assisted 3D Mapping of Urban Search and Rescue Environments”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. 2007, pp. 3889–3894. DOI: 10.1109/ROBOT.2007.364075.

[26] Viska Mutiawani, Cut Thifal Nazila, and Kurnia Saputra. “WLAN Based Indoor Localization System for Evacuation of Victims in a Building”. In: *2020 International Conference on Electrical Engineering and Informatics (ICELTICs)*. 2020, pp. 1–6. DOI: 10.1109/ICELTICs50595.2020.9315388.

[27] Yanxiao Feng et al. “Bimil: automatic generation of bim-based indoor localization user interface for emergency response”. In: *International Conference on Human-Computer Interaction*. Springer. 2020, pp. 184–192.

[28] Shayan Nikoohemat et al. “Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management”. In: *Automation in construction* 113 (2020), p. 103109.

[29] Marco Carli, Stefano Panzieri, and Federica Pascucci. “A joint routing and localization algorithm for emergency scenario”. In: *Ad Hoc Networks* 13 (2014), pp. 19–33.

[30] Rui Zhang, Fabian Hoflinger, and Leonhard Reindl. "Inertial sensor based indoor localization and monitoring system for emergency responders". In: *IEEE Sensors Journal* 13.2 (2012), pp. 838–848.

[31] Taslim Arefin Khan et al. "Escalating post-disaster rescue missions through ad-hoc victim localization exploiting Wi-Fi networks". In: *Heliyon* 8.5 (2022), e09314.

[32] Claudio Ettore Casetti et al. "Data Connectivity and Smart Group Formation in Wi-Fi Direct Multi-Group Networks". In: *IEEE Transactions on Network and Service Management* 15.1 (2018), pp. 245–259. DOI: `10.1109/TNSM.2017.2766124`.

[33] Gianluca Aloi et al. "The SENSE-ME platform: Infrastructure-less smartphone connectivity and decentralized sensing for emergency management". In: *Pervasive and Mobile Computing* 42 (2017), pp. 187–208.

[34] Zongqing Lu, Guohong Cao, and Thomas La Porta. "Teamphone: Networking smartphones for disaster recovery". In: *IEEE Transactions on Mobile Computing* 16.12 (2017), pp. 3554–3567.

[35] Luca Sciullo, Angelo Trotta, and Marco Di Felice. "Design and performance evaluation of a LoRa-based mobile emergency management system (LOCATE)". In: *Ad Hoc Networks* 96 (2020), p. 101993.

[36] Luca Fibbi et al. "Correction of a 1 km daily rainfall dataset for modelling forest ecosystem processes in Italy". In: *Meteorological Applications* 23.2 (2016), pp. 294–303.

[37] Md Abiar Rahman et al. "Impacts of temperature and rainfall variation on rice productivity in major ecosystems of Bangladesh". In: *Agriculture & Food Security* 6.1 (2017), pp. 1–11.

[38] Yixin Zhou et al. "Urban rain flood ecosystem design planning and feasibility study for the enrichment of smart cities". In: *Sustainability* 13.9 (2021), p. 5205.

[39] Congzheng Han et al. "Rainfall monitoring based on next-generation millimeter-wave backhaul technologies in a dense urban environment". In: *Remote Sensing* 12.6 (2020), p. 1045.

[40] Xi Shen et al. "3-D Tomographic Reconstruction of Rain Field Using Microwave Signals From LEO Satellites: Principle and Simulation Results". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.8 (2019), pp. 5434–5446. DOI: `10.1109/TGRS.2019.2899391`.

[41] Hagit Messer. "Capitalizing on cellular technology—Opportunities and challenges for near ground weather monitoring". In: *Environments* 5.7 (2018), p. 73.

[42] Roy Janco, Jonatan Ostrometzky, and Hagit Messer. "Rain Estimation from Smart City's E-band Links". In: *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*. IEEE. 2022, pp. 1–5.

[43] Feyisa Debo Diba et al. "Wireless telecommunication links for rainfall monitoring: deep learning approach and experimental results". In: *IEEE Access* 9 (2021), pp. 66769–66780.

[44] Hansong Guo et al. "Chaac: Real-Time and Fine-Grained Rain Detection and Measurement Using Smartphones". In: *IEEE Internet of Things Journal* 6.1 (2019), pp. 997–1009. DOI: 10.1109/JIOT.2018.2866690.

[45] Edgar Marko Trono et al. "Rainfall monitoring using acoustic sensors". In: *TENCON 2012 IEEE Region 10 Conference*. IEEE. 2012, pp. 1–6.

[46] Meriem Ferroudj et al. "Detection of rain in acoustic recordings of the environment". In: *Pacific Rim International Conference on Artificial Intelligence*. Springer. 2014, pp. 104–116.

[47] Xing Wang et al. "Rainfall observation using surveillance audio". In: *Applied Acoustics* 186 (2022), p. 108478.

[48] Mazin Alshamrani. "IoT and artificial intelligence implementations for remote healthcare monitoring systems: A survey". In: *Journal of King Saud University-Computer and Information Sciences* (2021).

[49] Mohammad Ghamari et al. "A survey on wireless body area networks for ehealthcare systems in residential environments". In: *Sensors* 16.6 (2016), p. 831.

[50] Gonçalo Marques et al. "Internet of things architectures, technologies, applications, challenges, and future directions for enhanced living environments and healthcare systems: a review". In: *Electronics* 8.10 (2019), p. 1081.

[51] Carlos Pereira et al. "Open IoT architecture for continuous patient monitoring in emergency wards". In: *Electronics* 8.10 (2019), p. 1074.

[52] Ghulam Muhammad, M. Shamim Hossain, and Neeraj Kumar. "EEG-Based Pathology Detection for Home Health Monitoring". In: *IEEE Journal on Selected Areas in Communications* 39.2 (2021), pp. 603–610. DOI: 10.1109/JSAC.2020.3020654.

[53] Weilin Zang et al. "CMDP-based intelligent transmission for wireless body area network in remote health monitoring". In: *Neural computing and applications* 32.3 (2020), pp. 829–837.

[54] M Shamim Hossain et al. "Applying deep learning for epilepsy seizure detection and brain mapping visualization". In: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15.1s (2019), pp. 1–17.

[55] Young Moon Chae et al. "Patient satisfaction with telemedicine in home health services for the elderly". In: *International journal of medical informatics* 61.2-3 (2001), pp. 167–173.

[56] Emelia J Benjamin et al. "Heart disease and stroke statistics—2017 update: a report from the American Heart Association". In: *circulation* 135.10 (2017), e146–e603.

[57] Roberta Avanzato and Francesco Beritelli. "Automatic ECG diagnosis using convolutional neural network". In: *Electronics* 9.6 (2020), p. 951.

[58] Jingshan Huang et al. "ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network". In: *IEEE Access* 7 (2019), pp. 92871–92880.

[59] Xue Xu, Sohyun Jeong, and Jianqiang Li. "Interpretation of electrocardiogram (ECG) rhythm by combined CNN and BiLSTM". In: *IEEE Access* 8 (2020), pp. 125380–125388.

[60] Roberta Avanzato and Francesco Beritelli. "Heart sound multiclass analysis based on raw data and convolutional neural network". In: *IEEE Sensors Letters* 4.12 (2020), pp. 1–4.

[61] Buscema Paolo Massimo et al. "Computer Aided Diagnosis for atrial fibrillation based on new artificial adaptive systems". In: *Computer Methods and Programs in Biomedicine* 191 (2020), p. 105401. ISSN: 0169-2607.

[62] Hongpo Zhang et al. "TP-CNN: A Detection Method for atrial fibrillation based on transposed projection signals with compressed sensed ECG". In: *Computer Methods and Programs in Biomedicine* 210 (2021), p. 106358.

[63] Aizatul Shafiqah Mohd Faizal et al. "A review of risk prediction models in cardiovascular disease: conventional approach vs. artificial intelligent approach". In: *Computer Methods and Programs in Biomedicine* 207 (2021), p. 106190. ISSN: 0169-2607.

[64] Amit Krishna Dwivedi, Syed Anas Imtiaz, and Esther Rodriguez-Villegas. "Algorithms for automatic analysis and classification of heart sounds–a systematic review". In: *IEEE Access* 7 (2018), pp. 8316–8345.

[65] Francesco Beritelli et al. "Automatic heart activity diagnosis based on Gram polynomials and probabilistic neural networks". In: *Biomedical engineering letters* 8.1 (2018), pp. 77–85.

[66] Samit Kumar Ghosh et al. "Automated detection of heart valve disorders from the PCG signal using time-frequency magnitude and phase features". In: *IEEE Sensors Letters* 3.12 (2019), pp. 1–4.

[67] Gharehbaghi Arash, Ask Per, and Babic Ankica. "A pattern recognition framework for detecting dynamic changes on cyclic time series". In: *Pattern Recognition* 48.3 (2015), pp. 696–708. ISSN: 0031-3203.

[68] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. "Real-time patient-specific ECG classification by 1-D convolutional neural networks". In: *IEEE Transactions on Biomedical Engineering* 63.3 (2015), pp. 664–675.

[69] Udit Satija, Barathram Ramkumar, and M Sabarimalai Manikandan. "Automated ECG noise detection and classification system for unsupervised healthcare monitoring". In: *IEEE Journal of biomedical and health informatics* 22.3 (2017), pp. 722–732.

[70] Chunbo Luo et al. "A UAV-cloud system for disaster sensing applications". In: *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE. 2015, pp. 1–5.

[71] Milan Erdelj, Michał Król, and Enrico Natalizio. "Wireless sensor networks and multi-UAV systems for natural disaster management". In: *Computer Networks* 124 (2017), pp. 72–86.

[72] Zhongli Liu et al. "HAWK: An unmanned mini-helicopter-based aerial wireless kit for localization". In: *IEEE Transactions on Mobile Computing* 13.2 (2012), pp. 287–298.

[73] Farshad Koohifar, Abhaykumar Kumbhar, and Ismail Guvenc. "Receding horizon multi-UAV cooperative tracking of moving RF source". In: *IEEE Communications Letters* 21.6 (2016), pp. 1433–1436.

[74] Margot Deruyck et al. "Emergency ad-hoc networks by using drone mounted base stations for a disaster scenario". In: *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2016, pp. 1–7.

[75] Margot Deruyck et al. "Designing UAV-aided emergency networks for large-scale disaster scenarios". In: *EURASIP Journal on Wireless Communications and Networking* 2018.1 (2018), pp. 1–12.

[76] Hui Liu et al. "Survey of wireless indoor positioning techniques and systems". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080.

[77] Hui Liu et al. "Survey of Wireless Indoor Positioning Techniques and Systems". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007), pp. 1067–1080. DOI: 10.1109/TSMCC.2007.905750.

[78] S. M. Mehdi Dehghan, Saeid Haidari, and Hadi Mordi. "Toward aerial simultaneous target localization and obstacle estimation using RSSI observations". In: *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*. 2015, pp. 517–522. DOI: 10.1109/ICRoM.2015.7367837.

[79] Tatyana Golubeva et al. "A Study on the Wi-Fi Radio Signal Attenuation in Various Construction Materials (Obstacles)". In: *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2018, pp. 718–723. DOI: 10.1109/ICUFN.2018.8436785.

[80] Jiao He et al. "Radio frequency propagation characteristics in disaster scenarios". In: *Proceedings of 2014 3rd Asia-Pacific Conference on Antennas and Propagation*. 2014, pp. 818–821. DOI: 10.1109/APCAP.2014.6992624.

[81] Sai-Qiong Zhou et al. "Investigation of the attenuation characteristics of radio signal in ruins". In: *2015 IEEE International Wireless Symposium (IWS 2015)*. 2015, pp. 1–4. DOI: 10.1109/IEEE-IWS.2015.7164543.

[82] Yong Zeng, Rui Zhang, and Teng Joon Lim. "Wireless communications with unmanned aerial vehicles: opportunities and challenges". In: *IEEE Communications Magazine* 54.5 (2016), pp. 36–42. DOI: 10.1109/MCOM.2016.7470933.

[83] Bin Li, Zesong Fei, and Yan Zhang. “UAV Communications for 5G and Beyond: Recent Advances and Future Trends”. In: *IEEE Internet of Things Journal* 6.2 (2019), pp. 2241–2263. DOI: 10.1109/JIOT.2018.2887086.

[84] Margot Deruyck et al. “Emergency ad-hoc networks by using drone mounted base stations for a disaster scenario”. In: *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2016, pp. 1–7. DOI: 10.1109/WiMOB.2016.7763173.

[85] Karina Gomez et al. “Capacity evaluation of Aerial LTE base-stations for public safety communications”. In: *2015 European Conference on Networks and Communications (EuCNC)*. 2015, pp. 133–138. DOI: 10.1109/EuCNC.2015.7194055.

[86] Jiangbin Lyu et al. “Placement Optimization of UAV-Mounted Mobile Base Stations”. In: *IEEE Communications Letters* 21.3 (2017), pp. 604–607. DOI: 10.1109/LCOMM.2016.2633248.

[87] Elham Kalantari, Halim Yanikomeroglu, and Abbas Yongacoglu. “On the Number and 3D Placement of Drone Base Stations in Wireless Cellular Networks”. In: *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. 2016, pp. 1–6. DOI: 10.1109/VTCFall.2016.7881122.

[88] Akram Al-Hourani, Sithamparanathan Kandeepan, and Simon Lardner. “Optimal LAP Altitude for Maximum Coverage”. In: *IEEE Wireless Communications Letters* 3.6 (2014), pp. 569–572. DOI: 10.1109/LWC.2014.2342736.

[89] Mohamed Alzenad et al. “3-D Placement of an Unmanned Aerial Vehicle Base Station (UAV-BS) for Energy-Efficient Maximal Coverage”. In: *IEEE Wireless Communications Letters* 6.4 (2017), pp. 434–437. DOI: 10.1109/LWC.2017.2700840.

[90] Mohamed Alzenad, Amr El-Keyi, and Halim Yanikomeroglu. “3-D Placement of an Unmanned Aerial Vehicle Base Station for Maximum Coverage of Users With Different QoS Requirements”. In: *IEEE Wireless Communications Letters* 7.1 (2018), pp. 38–41. DOI: 10.1109/LWC.2017.2752161.

[91] *The Basics of Signal Attenuation.* URL: https://www.dataloggerinc.com/wp-content/uploads/2016/11/16_Basics_of_signal_attenuation.pdf (visited on 12/03/2019).

[92] *Technical sheet of MATRIX SERIES 200 V2.* URL: https://www.dji.com/it/matrice-200-series-v2/%20info#specs (visited on 11/28/2019).

[93] Roberta Avanzato and Francesco Beritelli. "An Innovative Technique for Identification of Missing Persons in Natural Disaster Based on Drone-Femtocell Systems". In: *Sensors* 19.20 (2019). URL: https://www.mdpi.com/1424-8220/19/20/4547.

[94] Roberta Avanzato and Francesco Beritelli. "A Smart UAV-Femtocell Data Sensing System for Post-Earthquake Localization of People". In: *IEEE Access* 8 (2020), pp. 30262–30270. DOI: 10.1109/ACCESS.2020.2972699.

[95] Dingcheng Yang et al. "Energy Tradeoff in Ground-to-UAV Communication via Trajectory Design". In: *IEEE Transactions on Vehicular Technology* 67.7 (2018), pp. 6721–6726. DOI: 10.1109/TVT.2018.2816244.

[96] Fahui Wu et al. "Energy Consumption and Completion Time Tradeoff in Rotary-Wing UAV Enabled WPCN". In: *IEEE Access* 7 (2019), pp. 79617–79635. DOI: 10.1109/ACCESS.2019.2922651.

[97] Yiwen Nie et al. "Energy-efficient UAV trajectory design for backscatter communication: A deep reinforcement learning approach". In: *China Communications* 17.10 (2020), pp. 129–141. DOI: 10.23919/JCC.2020.10.009.

[98] Xiaoyan Hu et al. "UAV-Assisted Relaying and Edge Computing: Scheduling and Trajectory Optimization". In: *IEEE Transactions on Wireless Communications* 18.10 (2019), pp. 4738–4752. DOI: 10.1109/TWC.2019.2928539.

[99] Giacomo Bacci, Luca Sanguinetti, and Marco Luise. "Understanding Game Theory via Wireless Power Control [Lecture Notes]". In: *IEEE Signal Processing Magazine* 32.4 (2015), pp. 132–137. DOI: 10.1109/MSP.2015.2406338.

[100] Philip B. Charlesworth. "A Non-cooperative Game to Coordinate the Coverage of Two Communications UAVs". In: *MILCOM 2013 - 2013 IEEE Military Communications Conference*. 2013, pp. 668–673. DOI: 10.1109/MILCOM.2013.119.

[101] Roberta Avanzato et al. "Optimization of UAV-Femtocell Systems Positioning via Game Theory to Geolocate Mobile Terminals in a Post-Earthquake Scenario". In: *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Vol. 2. 2021, pp. 785–790. DOI: 10.1109/IDAACS53288.2021.9660873.

[102] Roberta Avanzato et al. “Game Theory-based Energy Efficiency Optimization of UAV-Femtocell Geolocalization Systems”. Manuscript submitted for publication. 2022.

[103] Jorge M Trabal and David J McLaughlin. “Rainfall estimation and rain gauge comparison for x-band polarimetric CASA radars”. In: *2007 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2007, pp. 2726–2729.

[104] Dieter Nagel. “Detection of rain areas with airborne radar”. In: *2017 18th International Radar Symposium (IRS)*. IEEE. 2017, pp. 1–7.

[105] Anoop Kumar Shukla, CSP Ojha, and Rahul D Garg. “Comparative study of trmm satellite predicted rainfall data with rain gauge data over himalayan basin”. In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2018, pp. 9347–9350.

[106] Shih-Hau Fang and Yu-Hsuan Sylvia Yang. “The impact of weather condition on radio-based distance estimation: A case study in GSM networks with mobile measurements”. In: *IEEE Transactions on Vehicular Technology* 65.8 (2015), pp. 6444–6453.

[107] Francesco Beritelli et al. “A neural network pattern recognition approach to automatic rainfall classification by using signal strength in LTE/4G networks”. In: *International Joint Conference on Rough Sets*. Springer. 2017, pp. 505–512.

[108] Francesco Beritelli et al. “Rainfall Estimation Based on the Intensity of the Received Signal in a LTE/4G Mobile Terminal by Using a Probabilistic Neural Network”. In: *IEEE Access* 6 (2018), pp. 30865–30873. DOI: 10.1109/ACCESS.2018.2839699.

[109] Aishwarya Himanshu Manek and Parikshit Kishor Singh. “Comparative study of neural network architectures for rainfall prediction”. In: *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*. IEEE. 2016, pp. 171–174.

[110] Apoorv Gupta et al. “Urban waterlogging detection and severity prediction using artificial neural networks”. In: *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. IEEE. 2017, pp. 42–49.

[111] K. L. Keung et al. “Smart City Application and Analysis: Real-time Urban Drainage Monitoring by IoT Sensors: A Case Study of Hong Kong”. In: *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. 2018, pp. 521–525. DOI: 10.1109/IEEM.2018.8607303.

[112] Roberta Avanzato et al. “A convolutional neural networks approach to audio classification for rainfall estimation”. In: *2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Vol. 1. IEEE. 2019, pp. 285–289.

[113] Roberta Avanzato and Francesco Beritelli. “An innovative acoustic rain gauge based on convolutional neural networks”. In: *Information* 11.4 (2020), p. 183.

[114] Michele Russo et al. “A CNN-based Audio Sensor for Rainfall Estimation: Implementation on Embedded Board”. In: *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Vol. 2. IEEE. 2021, pp. 911–915.

[115] Roberta Avanzato and Francesco Beritelli. “A cnn-based differential image processing approach for rainfall classification”. In: *Advances in Science, Technology and Engineering Systems Journal* 5.4 (2020), pp. 438–444.

[116] Roberta Avanzato and Francesco Beritelli. “Hydrogeological risk management in smart cities: A new approach to rainfall classification based on LTE cell selection parameters”. In: *IEEE Access* 8 (2020), pp. 137161–137173.

[117] Ryota Nakazato et al. “Rainfall intensity estimation from sound for generating CG of rainfall scenes”. In: *2018 International Workshop on Advanced Image Technology (IWAIT)*. IEEE. 2018, pp. 1–4.

[118] Toni Heittola, Emre Çakır, and Tuomas Virtanen. “The machine learning approach for analysis of sound scenes and events”. In: *Computational Analysis of Sound Scenes and Events*. Springer, 2018, pp. 13–40.

[119] Ma Madecheen S Pangaliman, Febus Reidj G Cruz, and Timothy M Amado. “Machine learning predictive models for improved acoustic disdrometer”. In: *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*. IEEE. 2018, pp. 1–5.

[120] *Smaniotto*. URL: `http://www.smaniotto.eu/scale-della-natura.html` (visited on 03/12/2020).

[121] *Keras Speaker Recognition*. URL: `https://keras.io/examples/audio/speaker_recognition_using_cnn/` (visited on 08/17/2022).

[122] *Using TensorFlow Lite with Python*. URL: `https://www.tensorflow.org/lite/guide/python` (visited on 08/17/2022).

[123] Shih-Hau Fang and Yu-Hsuan Sylvia Yang. "The Impact of Weather Condition on Radio-Based Distance Estimation: A Case Study in GSM Networks With Mobile Measurements". In: *IEEE Transactions on Vehicular Technology* 65.8 (2016), pp. 6444–6453. DOI: `10.1109/TVT.2015.2479591`.

[124] Jaime Gallego et al. "Glomerulus classification and detection based on convolutional neural networks". In: *Journal of Imaging* 4.1 (2018), p. 20.

[125] José A Godoy-Rosario et al. "An approach to rain detection using sobel image pre-processing and convolutional neuronal networks". In: *International Work-Conference on Artificial Neural Networks*. Springer. 2019, pp. 27–38.

[126] Sameer Sawant and PA Ghonge. "Estimation of rain drop analysis using image processing". In: *Int. J. Sci. Res* 4.1 (2015), pp. 1981–1986.

[127] S Tawade, P Ghonge, and K Tuckley. "Drop-size analysis using 2D Double Density Dual Tree Discrete Wavelet Transform". In: *International Conference on Communication and Signal Processing 2016 (ICCASP 2016)*. Atlantis Press. 2016, pp. 411–418.

[128] Fawzi Nashashibi, Raoul de Charrette, and Alexandre Lia. "Detection of unfocused raindrops on a windscreen using low level image processing". In: *2010 11th International Conference on Control Automation Robotics & Vision*. IEEE. 2010, pp. 1410–1415.

[129] Corrado Rametta et al. "A Smart VPN Bonding Technique for Drone Communication Applications". In: *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2019, pp. 612–618. DOI: `10.1109/DCOSS.2019.00112`.

[130] Atul K Varma. "Measurement of precipitation from satellite radiometers (visible, infrared, and microwave): Physical basis, methods, and limitations". In: *Remote Sensing of Aerosols, Clouds, and Precipitation*. Elsevier, 2018, pp. 223–248.

[131] Dani Cherkassky, Jonatan Ostrometzky, and Hagit Messer. “Precipitation classification using measurements from commercial microwave links”. In: *IEEE Transactions on Geoscience and Remote Sensing* 52.5 (2013), pp. 2350–2356.

[132] Aart Overeem, Hidde Leijnse, and Remko Uijlenhoet. “Rainfall monitoring using microwave links from cellular communication networks: The dutch experience”. In: *2018 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE. 2018, pp. 110–114.

[133] Congzheng Han and Shu Duan. “Impact of atmospheric parameters on the propagated signal power of millimeter-wave bands based on real measurement data”. In: *IEEE Access* 7 (2019), pp. 113626–113641.

[134] Sebin Sabu et al. “Effect of rainfall on cellular signal strength: A study on the variation of RSSI at user end of smartphone during rainfall”. In: *2017 IEEE Region 10 Symposium (TENSYMP)*. IEEE. 2017, pp. 1–4.

[135] Vasilis Christofilakis et al. “Rainfall measurements due to radio frequency signal attenuation at 2 GHz”. In: *Journal of Signal and Information Processing* 9.03 (2018), p. 192.

[136] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[137] Antonio Cimmino et al. “The role of small cell technology in future smart city applications”. In: *Transactions on Emerging Telecommunications Technologies* 25.1 (2014), pp. 11–20.

[138] Md Shipon Ali, Ekram Hossain, and Dong In Kim. “LTE/LTE-A random access for massive machine-type communications in smart cities”. In: *IEEE Communications Magazine* 55.1 (2017), pp. 76–83.

[139] Filippo Malandra et al. “Traffic characterization and LTE performance analysis for M2M communications in smart cities”. In: *Pervasive and Mobile Computing* 48 (2018), pp. 59–68.

[140] ETSI 3GPP. *Physical Channels and Modulation, Release 11, Section 6.10*. Tech. rep. Sept. 2013. URL: https://www.etsi.org/deliver/etsi_ts/136200_136299/136211/11.05.00_60/ts_136211v110500p.pdf.

[141] ETSI 3GPP. *User Equipment (UE) Procedures in Idle Mode, Release 11, Sections 5.2.1, 5.2.2, 5.2.3, 5.3*. Tech. rep. Sept. 2013. URL: https://www.etsi.org/deliver/etsi_ts/136300_136399/136304/12.02.00_60/ts_136304v120200p.pdf.

[142] ETSI 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer; Measurements, Release 11, Section 5.1.1.* Tech. rep. Dec. 2012. URL: `https://www.etsi.org/deliver/etsi_ts/136200_136299/136214/11.01.00_60/ts_136214v110100p.pdf`.

[143] ETSI 3GPP. *Requirements for Support of Radio Resource Management, Release 11, Section 4.2.* Tech. rep. Sept. 2013. URL: `https://www.etsi.org/deliver/etsi_ts/136100_136199/136133/11.04.00_60/ts_136133v110400p.pdf`.

[144] *Fischer Discriminant Ratio.* URL: `https://en.wikipedia.org/wiki/Linear_discriminant_analysis` (visited on 08/15/2022).

[145] Francesco Beritelli and Andrea Spadaccini. "A statistical approach to biometric identity verification based on heart sounds". In: *2010 Fourth International Conference on Emerging Security Information, Systems and Technologies*. IEEE. 2010, pp. 93–96.

[146] Francesco Beritelli and Andrea Spadaccini. "The role of voice activity detection in forensic speaker verification". In: *2011 17th International Conference on Digital Signal Processing (DSP)*. IEEE. 2011, pp. 1–6.

[147] Andrew P Bradley. "The use of the area under the ROC curve in the evaluation of machine learning algorithms". In: *Pattern recognition* 30.7 (1997), pp. 1145–1159.

[148] Luiz Fernando Afra Brito and Marcelo Keese Albertini. "Data mining of meteorological-related attributes from smartphone data". In: *INFOCOMP Journal of Computer Science* 15.2 (2016), pp. 1–9.

[149] Zheng Wu et al. "Online multimedia traffic classification from the QoS perspective using deep learning". In: *Computer Networks* 204 (2022), p. 108716.

[150] Thi Ha Ly Dinh et al. "Distributed user-to-multiple access points association through deep learning for beyond 5G". In: *Computer Networks* 197 (2021), p. 108258.

[151] Weifeng Sun, Zun Wang, and Guanghao Zhang. "A QoS-guaranteed intelligent routing mechanism in software-defined networks". In: *Computer Networks* 185 (2021), p. 107709.

[152] Joseph Isabona et al. "Atmospheric propagation modelling for terrestrial radio frequency communication links in a tropical wet and dry savanna climate". In: *Information* 13.3 (2022), p. 141.

[153] Emanuele Goldoni et al. “Correlation between weather and signal strength in Lorawan Networks: An extensive dataset”. In: *Computer Networks* 202 (2022), p. 108627.

[154] Roberta Avanzato, Francesco Beritelli, and Corrado Rametta. “An DL-based Approach to Packet Error Compensation Using Radio Mobile Network Quality Parameters in a Rainfall Scenario”. Manuscript Accepted for publication. 2022.

[155] F. Brouwer et al. “Usage of link-level performance indicators for HSDPA network-level simulations in E-UMTS”. In: *Eighth IEEE International Symposium on Spread Spectrum Techniques and Applications - Programme and Book of Abstracts (IEEE Cat. No.04TH8738)*. 2004, pp. 844–848. DOI: 10.1109/ISSSTA.2004.1371820.

[156] Klemens Freudenthaler, Andreas Springer, and Joachim Wehinger. “Novel SINR-to-CQI mapping maximizing the throughput in HSDPA”. In: *2007 IEEE Wireless Communications and Networking Conference*. IEEE. 2007, pp. 2231–2235.

[157] Klemens Freudenthaler, Andreas Springer, and Joachim Wehinger. “Novel SINR-to-CQI Mapping Maximizing the Throughput in HSDPA”. In: *2007 IEEE Wireless Communications and Networking Conference*. 2007, pp. 2231–2235. DOI: 10.1109/WCNC.2007.417.

[158] S Fouziya Sulthana and Rangaswamy Nakkeeran. “Development and analysis of downlink scheduling algorithm in LTE system with imperfect channel quality indicator”. In: *International Journal of Advanced Intelligence Paradigms* 18.1 (2021), pp. 79–93.

[159] Alejandro De La Fuente et al. “Subband CQI Feedback-Based Multicast Resource Allocation in MIMO-OFDMA Networks”. In: *IEEE Transactions on Broadcasting* 64.4 (2018), pp. 846–864. DOI: 10.1109/TBC.2018.2789578.

[160] Muntadher Qasim Abdulhasan et al. “An adaptive threshold feedback compression scheme based on channel quality indicator (CQI) in long term evolution (LTE) system”. In: *Wireless Personal Communications* 82.4 (2015), pp. 2323–2349.

[161] Rong Zeng et al. “Novel Channel Quality Indicator Prediction Scheme for Adaptive Modulation and Coding in High Mobility Environments”. In: *IEEE Access* 7 (2019), pp. 11543–11553. DOI: 10.1109/ACCESS.2019.2892228.

[162] Jihun Kim and Dong Seog Han. "Deep learning-based channel quality indicators prediction for vehicular communication". In: *ICT Express* (2022).

[163] Abitha K Thyagarajan et al. "SNR-CQI Mapping for 5G Downlink Network". In: *2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*. 2021, pp. 173–177. DOI: 10.1109/APWiMob51111.2021.9435258.

[164] Andson Balieiro, Kelvin Dias, and Paulo Guarda. "A Machine Learning Approach for CQI Feedback Delay in 5G and Beyond 5G Networks". In: *2021 30th Wireless and Optical Communications Conference (WOCC)*. 2021, pp. 26–30. DOI: 10.1109/WOCC53213.2021.9603019.

[165] Francesco Beritelli et al. "Rainfall estimation based on the intensity of the received signal in a LTE/4G mobile terminal by using a probabilistic neural network". In: *IEEE Access* 6 (2018), pp. 30865–30873.

[166] F Beritelli et al. "An advanced QOS analysis and evaluation method for mobile internet access". In: *International Journal of Wireless and Mobile Networks* 8.3 (2016), pp. 55–70.

[167] Ravi Prasad et al. "Bandwidth estimation: metrics, measurement techniques, and tools". In: *IEEE network* 17.6 (2003), pp. 27–35.

[168] Roberta Avanzato et al. "Exploiting VPN Bonding for Time Critical Video Transmission on Board Drone". In: *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Vol. 2. IEEE. 2021, pp. 808–812.

[169] Reza Rassool. "VMAF reproducibility: Validating a perceptual practical video quality metric". In: *2017 IEEE international symposium on broadband multimedia systems and broadcasting (BMSB)*. IEEE. 2017, pp. 1–2.

[170] Binjie Qin et al. "Accurate vessel extraction via tensor completion of background layer in X-RAY coronary angiograms". In: *Pattern Recognition* 87.6 (2019), pp. 38–54.

[171] Oulefki Adel et al. "Automatic COVID-19 lung infected region segmentation and measurement using CT-SCANS images". In: *Pattern Recognition* 114 (2021), p. 107747. ISSN: 0031-3203.

[172] Roberta Avanzato and Francesco Beritelli. "Heart disease recognition based on extended ECG sequence database and deep learning techniques". Manuscript submitted for publication. 2022.

[173] Roberta Avanzato et al. "Robust Recognition of Cardiac Pathologies Based on PCG Analysis and Deep Learning". Manuscript submitted for publication. 2022.

[174] Malay Mitra and RK Samanta. "Cardiac arrhythmia classification using neural networks with selected features". In: *Procedia Technology* 10 (2013), pp. 76–84.

[175] Stephen M Salerno, Patrick C Alguire, and Herbert S Waxman. "Competency in interpretation of 12-lead electrocardiograms: a summary and appraisal of published evidence". In: *Annals of Internal Medicine* 138.9 (2003), pp. 751–760.

[176] M Vijayavanan, V Rathikarani, and P Dhanalakshmi. "Automatic classification of ECG signal for heart disease diagnosis using morphological features". In: *International Journal of Computer Science & Engineering Technology* 5.4 (2014), pp. 449–455.

[177] Mario Sansone et al. "Electrocardiogram pattern recognition and analysis based on artificial neural networks and support vector machines: a review". In: *Journal of healthcare engineering* 4.4 (2013), pp. 465–504.

[178] Pranav Rajpurkar et al. "Cardiologist-level arrhythmia detection with convolutional neural networks". In: *arXiv preprint arXiv:1707.01836* (2017).

[179] U Rajendra Acharya et al. "Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network". In: *Knowledge-Based Systems* 132 (2017), pp. 62–71.

[180] Bahareh Taji, Adrian DC Chan, and Shervin Shirmohammadi. "Classifying measured electrocardiogram signal quality using deep belief networks". In: *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE. 2017, pp. 1–6.

[181] U Rajendra Acharya et al. "Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals". In: *Information Sciences* 415 (2017), pp. 190–198.

[182] Shalin Savalia and Vahid Emamian. "Cardiac arrhythmia classification by multi-layer perceptron and convolution neural networks". In: *Bioengineering* 5.2 (2018), p. 35.

[183] Jinghao Niu et al. "Inter-patient ECG classification with symbolic representations and multi-perspective convolutional neural networks". In: *IEEE journal of biomedical and health informatics* 24.5 (2019), pp. 1321–1332.

[184] Junxian Cai et al. “Multi-ECGNet for ECG arrythmia multi-label classification”. In: *Ieee Access* 8 (2020), pp. 110848–110858.

[185] Mohamed Hammad et al. “A multitier deep learning model for arrhythmia detection”. In: *IEEE Transactions on Instrumentation and Measurement* 70 (2020), pp. 1–9.

[186] Ehab Essa and Xianghua Xie. “An ensemble of deep learning-based multimodel for ECG heartbeats arrhythmia classification”. In: *IEEE Access* 9 (2021), pp. 103452–103464.

[187] Oh Shu Lih et al. “Comprehensive electrocardiographic diagnosis based on deep learning”. In: *Artificial intelligence in medicine* 103 (2020), p. 101789.

[188] Abdelrahman M Shaker et al. “Generalization of convolutional neural networks for ECG classification using generative adversarial networks”. In: *IEEE Access* 8 (2020), pp. 35592–35605.

[189] George B Moody, Roger G Mark, and Ary L Goldberger. “PhysioNet: a web-based resource for the study of physiologic signals”. In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 70–75.

[190] George B Moody and Roger G Mark. “The impact of the MIT-BIH arrhythmia database”. In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50.

[191] George Moody. “A new method for detecting atrial fibrillation using RR intervals”. In: *Computers in Cardiology* (1983), pp. 227–230.

[192] *Scikit Learn, Machine Learning in Python*. URL: `https://scikit-learn.org/stable/` (visited on 05/14/2020).

[193] Chaitra Sridhar et al. “Automated diagnosis of Coronary Artery Disease using nonlinear features extracted from ECG signals”. In: *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2016, pp. 000545–000549.

[194] Rakesh Ranjan et al. “A fuzzy neural network approach for automatic K-complex detection in sleep EEG signal”. In: *Pattern Recognition Letters* 115 (2018), pp. 74–83.

[195] Francesco Beritelli et al. “A novel training method to preserve generalization of RBPNN classifiers applied to ECG signals diagnosis”. In: *Neural Networks* 108 (2018), pp. 331–338.

[196] Muhammad Zubair, Jinsul Kim, and Changwoo Yoon. “An automated ECG beat classification system using convolutional neural networks”. In: *2016 6th international conference on IT convergence and security (ICITCS)*. IEEE. 2016, pp. 1–5.

[197] Dan Li et al. “Classification of ECG signals based on 1D convolution neural network”. In: *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE. 2017, pp. 1–6.

[198] Ulas Baran Baloglu et al. “Classification of myocardial infarction with multi-lead ECG signals and deep CNN”. In: *Pattern Recognition Letters* 122 (2019), pp. 23–30.

[199] Gui-Young Son, Soonil Kwon, et al. “Classification of heart sound signal using multiple features”. In: *Applied Sciences* 8.12 (2018), p. 2344.

[200] O El Badlaoui, A Benba, and A Hammouch. “Novel PCG analysis method for discriminating between abnormal and normal heart sounds”. In: *IRBM* 41.4 (2020), pp. 223–228.

[201] Anjali Yadav et al. “Machine learning-based classification of cardiac diseases from PCG recorded heart sounds”. In: *Neural Computing and Applications* 32.24 (2020), pp. 17843–17856.

[202] Palani Thanaraj Krishnan, Parvathavarthini Balasubramanian, and Snekhalatha Umapathy. “Automated heart sound classification system from unsegmented phonocardiogram (PCG) using deep neural network”. In: *Physical and Engineering Sciences in Medicine* 43.2 (2020), pp. 505–515.

[203] Francesco Renna, Jorge Oliveira, and Miguel T Coimbra. “Deep convolutional neural networks for heart sound segmentation”. In: *IEEE journal of biomedical and health informatics* 23.6 (2019), pp. 2435–2445.

[204] Fan Li et al. “Classification of heart sounds using convolutional neural network”. In: *Applied Sciences* 10.11 (2020), p. 3956.

[205] Juan P Dominguez-Morales et al. “Deep neural networks for the recognition and classification of heart murmurs using neuromorphic auditory sensors”. In: *IEEE transactions on biomedical circuits and systems* 12.1 (2017), pp. 24–34.

[206] Neeraj Baghel, Malay Kishore Dutta, and Radim Burget. “Automatic diagnosis of multiple cardiac diseases from PCG signals using convolutional neural network”. In: *Computer Methods and Programs in Biomedicine* 197 (2020), p. 105750.

[207] Chengyu Liu et al. “An open access database for the evaluation of heart sound algorithms”. In: *Physiological Measurement* 37.12 (2016), p. 2181.

[208] Ary L Goldberger et al. “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals”. In: *circulation* 101.23 (2000), e215–e220.

[209] Joachim Thiemann, Nobutaka Ito, and Emmanuel Vincent. “DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments”. In: *Proc. Meetings Acoust.* 2013, pp. 1–6.

[210] AM Thurston et al. “Bandpass sigma delta AD conversion”. In: *Analog Circuit Design*. Springer, 1993, pp. 259–281.

[211] Andrew Varga and Herman JM Steeneken. “Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems”. In: *Speech communication* 12.3 (1993), pp. 247–251.

[212] Takuya Iimura, Kazuo Murota, and Akihisa Tamura. “Discrete fixed point theorem reconsidered”. In: *Journal of Mathematical Economics* 41.8 (2005), pp. 1030–1036.

[213] Jun-ichi Sato and Hidefumi Kawasaki. “Discrete fixed point theorems and their application to Nash equilibrium”. In: *Taiwanese Journal of Mathematics* 13.2A (2009), pp. 431–440.

[214] Simone Sagratella. “Computing all solutions of Nash equilibrium problems with discrete strategy sets”. In: *SIAM Journal on Optimization* 26.4 (2016), pp. 2190–2218.

[215] Wei Dai et al. “Very deep convolutional neural networks for raw waveforms”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 421–425. DOI: 10.1109/ICASSP.2017.7952190.

[216] Forrest N. Iandola et al. *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and &lt;0.5MB model size*. 2016. DOI: 10.48550/ARXIV.1602.07360. URL: https://arxiv.org/abs/1602.07360.

[217] Frank Rosenbaltt. “The perceptron–a perciving and recognizing automation”. In: *Cornell Aeronautical Laboratory* (1957).

Appendix

Appendix A: Tools: Qucell 4G/LTE Small Cell, Accuver XCORE/XIMS, GMON and GeoDevice

.0.1 Qucell 4G/LTE Small Cell

The femtocell is a small base station of limited radius coverage and low power designed for home or small business use. In particular, the Qucell LTE Small Cell was used for the test beds in this paper. It is also called HeNB (Home evoluted Node UTRAN B); it has the function of wireless connection with the UE (User Equipment) to process packets with the LTE Air standards between the EU and EPC. The LTE Small Cell has the interaction of the HeNB function with the Femto GW server (Gateway), SeGW (Security Gateway), HeMS (HeNB Management System), see Figure A.1. Small Cell (HeNB) is connected to the SeGW and, through the authentication process, creates an IPSec connection between Small Cell and SeGW with which communication is activated. At the end of the authentication process, Small Cell is connected to HeNB GW (Femto GW), ready to serve the services by communicating with the MME (Mobility Management Entity). The process of setup, management, alarm, statistics, etc. is managed through the HeMS Server which has a connection with the Small Cell through the IP network.

The hardware and software specifications of the femtocell are described in Table 1.

.0.2 Accuver XCORE/XIMS

Accuver XCORE / XIMS is a software that implements EPC NF in a PC and provides tests and environment for E-UTRAN equipment: eNB or Home eNB. The EPC network includes MME ($S_1 - MME$) functions for mobility management and the EU-User Equipment session, S-GW ($S_1 - U$) functions for IP routing,

Table 1: LTE Femto AP – General H/W specification.

Title		Spec.	Note
Frequency Band	LTE	1805 – 1880 MHz downlink	LTE Band 3
		1710 – 1785 MHz uplink	
Centre Frequency		1849.9 MHZ DL 1754.9 MHz UL	1649 EARFCN DL 19649 EARFCN UL
Transmit Power		17 dBm/path	2 RF output types
Bandwidth		20 MHz FDD	
Antennas		2 x 2 MIMO	External Antenna (Gain: 3dBi)
Main Chipset	Baseband	Qualcomm FSM9905	SoC
	RF	Qulacomm FTR8900	RF
External Interface	Ethernet	Qualcomm AR8033 1 Gbps x 1 (Backhaul)	1000Base-T/RJ-45
Memory	RAM	2GByte	DDR3L RAM
	ROM	4GByte	Emmc

HSS functions for managing subscriber information and the PCRF functions for QoS.

.0.3 GMON

G-MoN is an application made in Android to control the signal level of a Wi-Fi network and to obtain very complete and detailed analysis of a 2G / 3G / 4G network. When running the application, apart from basic settings, G-MoN offers several options in its graphical interface, such as:

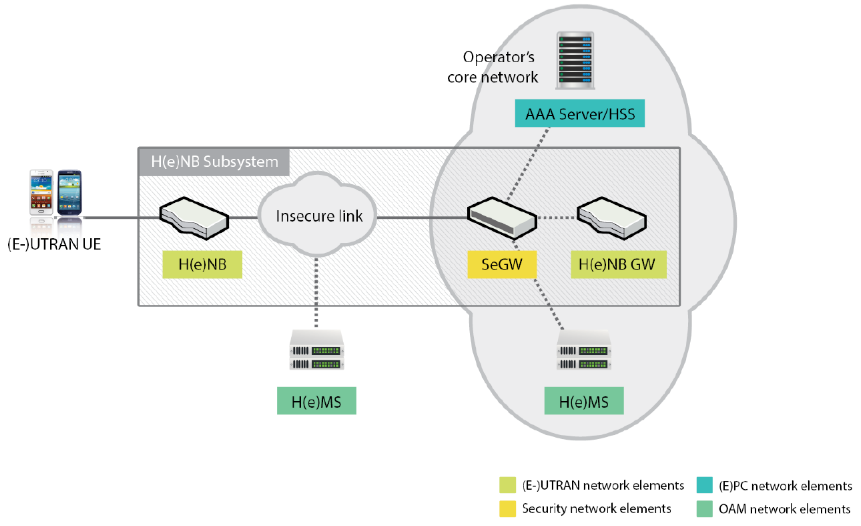


Figure A.1: Connection diagram for LTE Small Cell and EPC.

- “2G / 3G / 4G”: presents information and parameters relating to the mobile radio network (“CID”, “LAC”, “RXL”, “RSRP”, “RSRQ”, etc.). In particular, the information given by “RSRP” and “RXL” serves as the basis for the localization algorithm proposed in this paper;
- “Wi-Fi”: presents all the data concerning the wi-fi connection (“SSID”, “CH”, “RXL”, etc.);
- “Cell History”: contains what you need to know about the coordinates (latitude “LAT” and longitude “LON”), precision (“ACC”) and altitude (“ALT”). In this case, obviously, the GPS must be enabled.

.0.4 GeoDevice

In order to carry out a series of simulations of the scenario and the techniques proposed in this paper, a specific tool called *GeoDevice* was designed and implemented. The software implements a graphic engine to meet the project specifications. The graphic library adopted (low level) is OpenGL, which ensures extreme flexibility, not only in terms of graphic performance but also (and above all) cross-compiling on Linux, Windows and Mac. The structure of the engine is based on a series of low-level operating classes designed to reconstruct the three-dimensional primitives needed in the rendering phase. The rendering is made by objects, therefore it is easy to extend the graphic functionalities defining new classes and leaving the reconstruction operations to the aforementioned low level classes. The engine also implements a moving camera that allows you to explore the three-dimensional scenario in virtual reality.

The tool implements the:

- Classification Phase: which allows the classification of the terminals within or outside the various monitoring areas, considering, or not, the additional attenuation;
- Localization Phase: which allows the insertion of the scene in question and the data obtained from the measurements in the field, subsequently applying the localization algorithm of the mobile terminal.

As regards the “Classification Phase” the tool provides for the insertion of mobile terminals in one or more monitoring areas, simulates the power received from the terminals when the femtocell runs along the perimeter of the area and generates the polar diagram determining if the terminals they are classified within or outside the monitoring area. As for the “Localization Phase”, the tool is meant to outline a scene which, for the study carried out in this paper, corresponds to a building of a certain length, height and depth. Once the scene has been defined it is possible to load the data through a file with the “.json” extension. The data represent the power values measured at precise coordinates defined a priori. Once the data is loaded, the tool automatically proceeds to apply the algorithm for locating the terminal, adopting one of the three methods described above, based on the “power wall” created.

Appendix B: Game Theory

In this short section we summarize the basic concept of GT required to define the Nash equilibrium, in the case of two player game in a strategic form. Each player is given a finite strategy set:

$$A_1 = \{s_1, \ldots, s_n\}, \;\; A_2 = \{t_1, \ldots, t_m\}, \tag{1}$$

and a vector $(s,t) \in A = A_1 \times A_2$ is called a strategy profile.

Each player expresses a preference relation on the set A, , i.e. the player can order all the strategy profiles, for instance, from the best to the worst.

Players act so as to maximize their own welfare, which is described (for each player) through a utility function of the whole strategy profile:

$$u_1 : A_1 \times A_2 \to \mathbb{R}, \;\; u_2 : A_1 \times A_2 \to \mathbb{R}. \tag{2}$$

Each player only controls own strategies; the player has to solve an optimization problem for each strategy chosen by the other player. A Nash equilibrium is a strategy profile (s^*, t^*) for which no player can improve own utility by unilaterally changing strategy, that is, the following two inequalities must hold:

$$u_1(s^*, t^*) \geq u_1(s, t^*), \text{ for each } s \in A_1, \tag{3}$$

$$u_2(s^*, t^*) \geq u_2(s^*, t), \text{ for each } t \in A_2. \tag{4}$$

In the present case of two players and a finite set of strategies the “game” is usually represented using a table, which in the simplest fashion provides only two strategies available for each player (see below). Such a table corresponds to two matrices (one for each player) thus such a game is often called bimatrix game.

Specifically, the rows represent the two possible actions of player 1, while the columns represent the two possible actions of player 2, and the ordered pairs of numbers in each cell are the values of (u_1, u_2) which correspond to the strategy profile selected by the cell.

	t_1	t_2
s_1	$u_1(s_1,t_1),\ u_2(s_1,t_1)$	$u_1(s_1,t_2),\ u_2(s_1,t_2)$
s_2	$u_1(s_2,t_1),\ u_2(s_2,t_1)$	$u_1(s_2,t_2),\ u_2(s_2,t_2)$

Similarly, the concept of Nash equilibrium can be reformulated using the so called *best response function* combined with the bimatrix representation of the game also provides a method for the computation of the possible equilibrium.

The best response of player 1 to the strategy t chosen by player 2 is defined as follows:

$$B_1(t) = \{s^* \in A_1 : u_1(s^*, t) \geq u_1(s,t),\ \forall s \in A_1\},$$

that is, $B_1(t)$ is the subset of points of A_1 where the function $u_1(\cdot, t)$ reaches its maximum value. However, it should be noted that in the case of a finite strategy the set $B_1(t)$ can be either singleton or containing more points, but it cannot be empty.

Analogously:

$$B_2(s) = \{t^* \in A_2 : u_2(s, t^*) \geq u_2(s,t),\ \forall t \in A_2\}.$$

The definition of Nash equilibrium can thus be reformulated according to the following:
(s^*, t^*) is a Nash equilibrium if and only if

$$s^* \in B_1(t^*) \text{ and } t^* \in B_2(s^*). \tag{5}$$

Furthermore, we can introduce the map $B : A \to A$ defined by:

$$B(s,t) = (B_1(t), B_2(s))$$

which allows us to characterize a Nash equilibrium as a *fixed point* of B, i.e. a point $(s^*, t^*) \in A$ such that:

$$(s^*, t^*) \in B(s^*, t^*).$$

Although the best response of each player is always non-empty, it must be highlighted that in the case of a finite strategy set the existence of a fixed point of the map B is not guaranteed.

While there are numerous studies on fixed point theorems we have observed that there are very few papers dealing with the discrete case relevant to the purposes of the present study, where A only contains a finite number of points (see, e.g. [212–214]) and the established conditions therein do not fit our framework.

Indeed, the fact that Nash equilibria may fail to exist is well known and the standard approach to overcome this problem is to pass from the finite strategy scenario to the so-called *mixed strategies* which generally stands for a probability distribution over the finite strategy space and establishing the Nash equilibrium in this new framework. This approach, using a two-drone coverage, has been suggested in [100].

Considering the vital application scenario of our study, we do not intend to change the nature of the problem from a deterministic to a probabilistic one; instead, we aim to introduce a procedure which allows the drones to continue their search in the cases where Nash equilibrium does not exist. Numerous experiments conducted show that such cases are extremely rare and constitute a very low percentage of the total number of instances.

Appendix C: Very deep CNN for Raw Waveforms

CNN are a specialized kind of neural network for processing data that has a known grid-like topology. Examples include time-series data, which may be considered as a 1-D grid taking samples at regular time intervals, and image data, which may be considered as a 2-D grid of pixels.

The general characteristics and architecture of this network are described in [215], where the only difference is the sample rate used. In all studies using the CNN network described in this thesis, the sampling rate is 44.1 kHz instead of 8 kHz.

The deep convolutional neural network is mainly composed of:

1. 1D convolution layers;

2. Batch normalization layers;

3. RELU layers;
4. Pooling layers;
5. Softmax.

Only in the first convolution, a convolutional kernel composed of 80 elements is used. The subsequent convolution layers have it set to 3, with the aim of reducing the computational cost. After each convolution, batch normalization is carried out to avert the explosion of parameters and avoid the phenomenon of "vanishing gradients". Batch normalization allows training deep networks and is applied after each convolutional layer and before performing the ReLU (rectified linear activation function). The level of pooling in CNN, placed before RELU, reduces the problem of data overfitting by the network, taking the input size by half the actual input.

Unlike the classic CNN that uses fully connected neurons as their output layer, this network performs a single AvgPool and then a LogSofMax softmax followed by a natural logarithm log (softmax (x)). The structure of the proposed network is illustrated below and in A.2:

- Input Layer: Vectors of signal (PCG, ECG, Audio Rainfall) Samples (44100).
- Layer 1:
  - Conv1D (1, 128, 80, 4): 1 input channel, 128 output channel, 80 kernel size, stride 4;
  - BatchNorm1D (128): 128 features;
  - MaxPool1D with kernel size 4.
- Layer 2:
  - Conv1D (128, 128, 3): 128 input channel, 128 output channel, 3 kernel size;
  - BatchNorm1D (128): 128 features;
  - MaxPool1D with kernel size 4.
- Layer 3:
  - Conv1D (128, 256, 3): 128 input channel, 256 output channel, 3 kernel size;
  - BatchNorm1D (256): 256 features;

- MaxPool1D with kernel size 4.

- Layer 4:
  - Conv1D (256, 512, 3): 256 input channel, 512 output channel, 3 kernel size;
  - BatchNorm1D (512): 512 features;
  - MaxPool1D with kernel size 4.
- Output Layer:
  - AvgPool1D (30): 30 kernel size;
  - Linear (512, $num_{classes}$): input 1 × 512, 7 output classes;
  - Log Softmax.

The input of the neural network is a vector that contains sequences of raw waveforms (PCG, ECG, Audio Rainfall) each with duration equal to $W_{in} = 2$ seconds. The CNN neural network derives the index associated with one of the N_{C_i} different classes $C_i (i = 1, ..., N)$ using the LogSoftMax function.

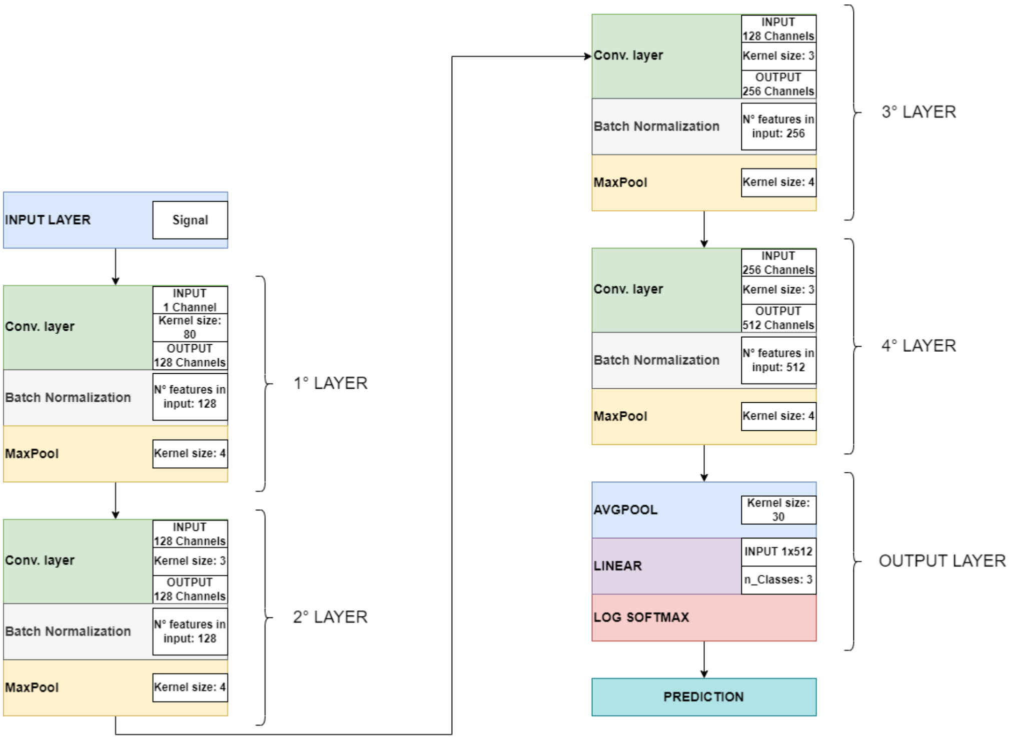


Figure A.2: CNN architecture for RAW waveform.

Appendix D: CNN: SqueezeNet

Convolutional Neural Network, called SqueezeNet [216],is a completely convolutional network with reduced complexity and with layers of dropouts. This network allows to obtain good accuracy and increase performance. To create a CNN network with reduced complexity, three main strategies are necessary:

i. Replace 3×3 filters with 1×1 filters. Given a budget of a certain number of convolution filters, we will choose to make the majority of these filters 1×1, since a 1×1 filter has $9X$ fewer parameters than a 3×3 filter.

ii. Decrease the number of input channels on "3×3" filters. Consider a convolution layer that is comprised entirely of 3×3 filters. The total quantity of parameters in this layer is $(numberofinputchannels) \cdot (numberoffilters) \cdot (3 \cdot 3)$. So, to maintain a small total number of parameters in a CNN, it is important not only to decrease the number of 3×3 filters (see i.), but also to decrease the number of input channels to the 3×3 filters.

iii. Delay downsample on the network so that convolution levels have large activation maps. In a convolutional network, each convolution layer produces an output activation map with a spatial resolution that is at least 1×1 and often much larger than 1×1. The height and width of these activation maps are controlled by: (1) the size of the input data (e.g. 256×256 images) and (2)the choice of layers in which to downsample in the CNN architecture. Most commonly, downsampling is engineered into CNN architectures by setting the $(stride > 1)$ in some of the convolution or pooling layers. If early3 layers in the network have large strides, then most layers will have small activation maps. Conversely, if most layers in the network have a stride of 1, and the strides greater than 1 are concentrated toward the end4 of the network, then many layers in the network will have large activation maps. Our intuition is that large activation maps (due to delayed downsampling) can lead to higher classification accuracy, with all else held equal.

Strategies 1 and 2 are about judiciously decreasing the quantity of parameters in a CNN while attempting to preserve accuracy. iii. is about maximizing accuracy on a limited budget of parameters.

A Fire module is comprised of: a *squeeze* convolution layer (which has only 1×1 filters), feeding into an expand layer that has a mix of 1×1 and 3×3 convolution filters; we illustrate this in Figure 1. The liberal use of 1×1 filters in Fire modules is an application of i. from Section 3.1. We expose three tunable dimensions (hyperparameters) in a Fire module: $s_{1\times1}$, $e_{1\times1}$, and $e_{3\times3}$. In a Fire module, $s_{1\times1}$ is the number of filters in the squeeze layer (all 1×1), $e_{1\times1}$ is the number of 1×1 filters in the expand layer, and $e_{3\times3}$ is the number of 3×3 filters in the expand layer. When we use Fire modules we set $s_{1\times1}$ to be less than ($e_{1\times1}$ + $e_{3\times3}$), so the squeeze layer helps to limit the number of input channels to the 3×3 filters, as per ii..

In Figure A.3 that SqueezeNet begins with a standalone convolution layer (conv1), followed by 8 Fire modules (fire2-9), ending with a final conv layer (conv10). In this type of network, the number of filters per fire module from the beginning to the end of the network increases. SqueezeNet performs max-pooling with a stride of 2 after layers conv1, fire4, fire8, and conv10; these relatively late placements of pooling are per iii..

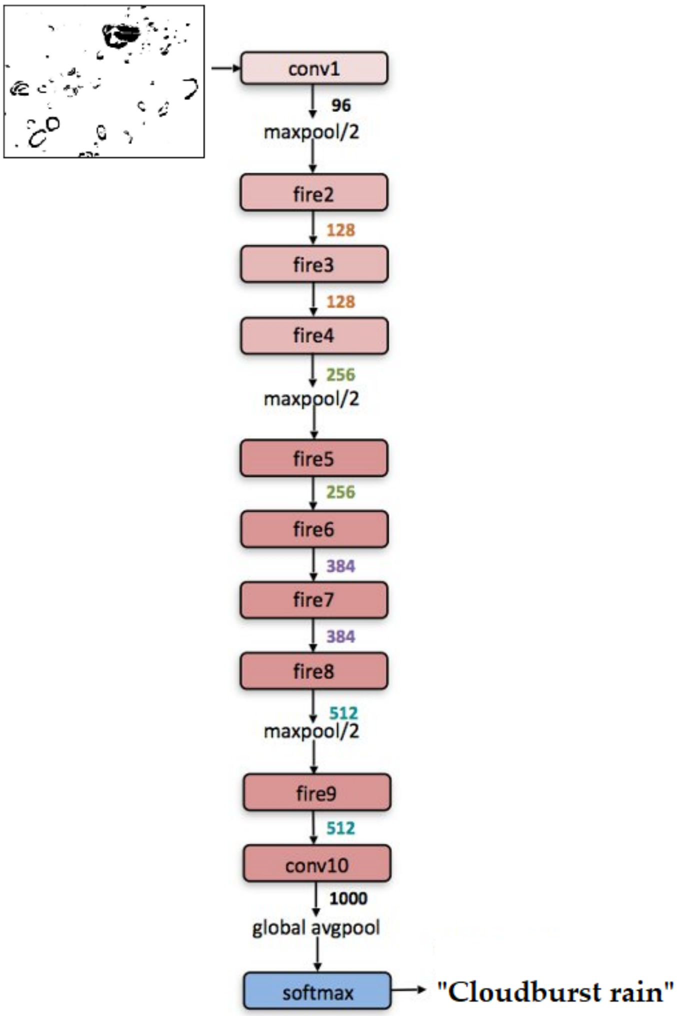


Figure A.3: Macroarchitectural view of our SqueezeNet architecture.

Appendix E: Multi-Layer Perceptron

One of the first algorithms used in machine learning for supervised learning is the SLP. Frank Rosemblat in [217] published the first concept of the Perceptron learning rule based on the McCullock-Pitts (MCP) neuron. This learning rule consists in an algorithm that automatically expresses the optimal weight coefficients to be multiplied by input characteristics. This multiplication operation allows making the decision on whether to activate the neuron or not; with the possibility, therefore, to define whether or not a given sample belongs to a particular class. Mathematically, it is possible to have input signals x_i and weights w_i. One must define a function of activation, which works on a linear combination of certain X input values and a corresponding vector of W weights, where Z is the neural network input (6):

$$\begin{gathered} W = \left[w_i \cdots w_m\right], X = \left[x_i \cdots x_m\right], \\ Z = \left(w_1 \cdot x_1 + w_2 \cdot x_2 + \cdots + w_m \cdot x_m\right), \\ \Theta\left(z\right) = \begin{cases} 1 & ifz \geq 0 \\ 0 & otherwise. \end{cases} \end{gathered} \tag{6}$$

If the activation of a particular sample $x^{(i)}$, which is the output of the $\Theta(z)$, is greater than the given threshold, one can predict which class it belongs to. In the Perceptron algorithm, the activation function is defined in sections. The input of the $z = W^T \cdot X$ network is reduced to a binary output by the Perceptron activation function, thus being used to linearly discriminate between the two classes.

SLP is a very simple neural network, which manages well to classify only if the variables are linearly separable. Based on the statistical analysis described in the following sections, it is possible to note that the variables involved in this study cannot be separated linearly.

For this reason, we have used a MLP network which, unlike the SLP which has a single hidden layer and a single neuron, contains multiple hidden layers and for each layer there are multiple neurons. The substantial difference is that sigmoid functions such as logistic regression are used in MLP networks, to activate neurons, according to (7).

$$\Theta\left(z\right) = \frac{1}{1 - e^{-z}} \tag{7}$$

MLP is a typical example of a feedforward artificial neural network. The term feedforward refers to the fact that each level acts as input to the next level, without loops. As for our study, it is possible to observe in Figure A.4 that we

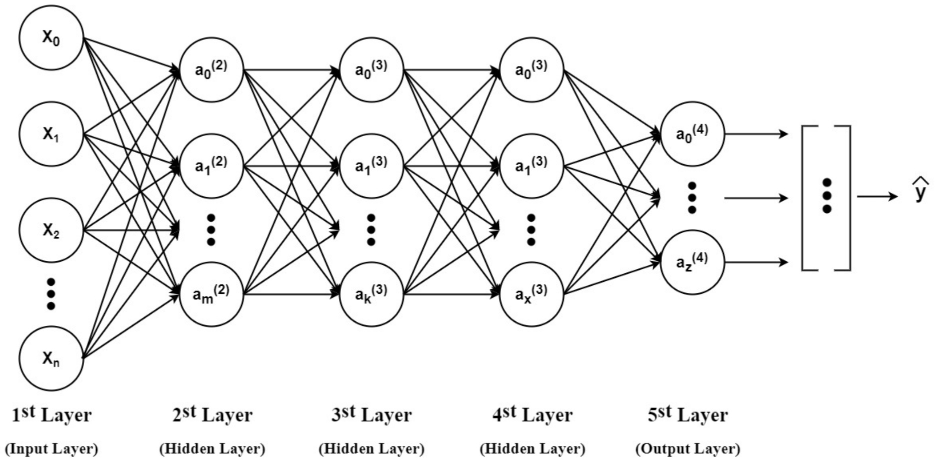


Figure A.4: Multi-Layer Perceptron architecture.

used a five-layer MLP. The first is the input layer, the second, third and fourth are hidden layers with 150, 100 and 50 neurons, respectively, while the last layer is the output layer.

Appendix F: Statistical Classification Parameter

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa.

The metrics of evaluation of the confusion matrix are:

- Accuracy (A): is how close or far off a given set of measurements (observations or readings) are to their true value;
- True Positive Ratio (TPR) or Sensitivity or Recall: refers to the probability of a positive test, conditioned on truly being positive;
- True Negative Ratio (TNR) or Specificity: refers to the probability of a negative test, conditioned on truly being negative;
- False Positive Ratio (FPR) or fall-out: is the probability of falsely rejecting the null hypothesis for a particular test;
- $F_1 Score$: is a measure of a test's accuracy. It is calculated from the precision and recall of the test.

These metrics are calculated through (8), (9), (10), (11), (12), respectively. Where TP is the number of True Positive, FN is the number of False Negative, TN is the number of True Negative e FP is the number of False Positive.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$TPR = \frac{TP}{TP + FN} \tag{9}$$

$$TNR = \frac{TN}{FP + TN} \tag{10}$$

$$FPR = 1 - TNR \tag{11}$$

$$F_1 Score = \frac{2TP}{2TP + FP + FN} \tag{12}$$