Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

PH.D THESIS

Integrating Connected UAVs into Future Mobile Networks

Author:
Erika G. P. da FONSECA

Supervisor:
Prof. Ivana DUSPARIC

Co-Supervisor:
Prof. Luiz A. DASILVA
Dr. Boris GALKIN 2022

2022

This thesis is submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

in the

CONNECT Centre for Telecommunications Research,
School of Computer Science and Statistics

November 20, 2022

Declaration of Authorship

I, Erika G. P. da FONSECA, declare that this thesis titled, "Integrating Connected UAVs into Future Mobile Networks" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Abstract

With the increasing number of Unmanned Aerial Vehicles (UAVs) and their applications, such as performing search and rescue or transplant organ delivery, the need for improving the UAV connectivity grows. Currently, User Equipment (UE)s have a range of connectivity options, such as WiFi and Lora, depending on the manufacturer's implementation. The integration of the UAV as a UE of the mobile network can increase the guarantee of the UAV's Quality of Service (QoS) and the range of its available connectivity, due to higher reliable and range of mobile networks, as the mobile network coverage is much bigger than a WiFi router coverage, for example. This would, in turn, enable wider and more reliable applications of UAVs. In this thesis, we investigate how to improve the QoS of a UAV connected to the mobile network, without requiring changes to the mobile network.

First research question that this thesis answers is "Which are the main challenges a connected UAV may encounter if deployed in a typical modern-day cellular network?". To derive our contributions, we investigate the challenges an operator should consider as UAVs become UEs of the network. We analyse the 3rd Generation Partnership Program (3GPP) specifications and the existing research literature. We illustrate our points by analysing a real-world UAV connectivity dataset. We discuss the challenge of planning network coverage when considering coverage for flying UEs and the physical cell identifier (PCI) collisions issues aggravated by such UEs. We observe that the identified challenges can be addressed either by the operator, or by the UAV. In the remainder of the thesis, we focus on the implementation on the UAV side, as it can accelerate the UAV

integration to the network.

The second research question we investigate is "How to adapt the UAV height in order to increase UAV's QoS?". In contrast with existing approaches that define the horizontal path only, we propose a Reinforcement Learning (RL)-based algorithm to optimise the QoS by adapting the height of a UAV, as it moves dynamically within a range of legally allowed heights, focusing on increasing its throughput. We investigate the proposed solution on a dataset of a UAV flying in different altitudes in the city of Dublin. As the empirical dataset is limited to two locations, in order to assess the wider applicability of the proposed approach, we investigate the performance of the proposed RL approach in a simulated environment. We vary the number of base stations and density of buildings to investigate how do they affect the optimal height of the UAV. Our results show that, in most scenarios, the proposed RL-based approach outperforms the closest related work baselines, achieving up to 6% in both investigations using real data and in simulation.

In order to facilitate UAV's use of the unlicensed spectrum, in this thesis we answer the third and final research question "How to better characterise the unlicensed spectrum in order to enable more efficient spectrum usage?". We propose a Convolutional Neural Network (CNN) approach to characterise the spectrum and facilitate dynamic access. We perform classification by processing spectrograms as images. We propose using object detection and a feature extraction module to extract features from spectrograms. In contrast to other methods, our proposed approach can recognise not only different Radio Access Technologys (RATs) in the shared spectrum but also identify critical parameters such as

inter-frame duration, frame duration, centre frequency, and signal bandwidth. The scenario considered is the coexistence of WiFi and Long-Term Evolution (LTE) transmissions in a shared spectrum. We implement and evaluate our solution with a real-world transmissions dataset and a test-bed environment. Our results show that our approach has an accuracy of 96% in the classification of RATs with the real-world dataset.

In summary, the three main contributions of this thesis, towards the integration of UAVs into 5G and beyond are: the identification of the mobility challenges a mobile operator may encounter if a UAV is integrated as a UE of the mobile network; a RL approach to optimise the UAV's QoS while adapting UAV's height; and an object detection approach that classify different RATs and extract features from the transmissions.

"Be strong and of a good courage; be not afraid, neither be thou dismayed: for the Lord thy God is with thee whithersoever thou goest."

Joshua 1:9

Acknowledgements

The PhD. process was a challenge that I am glad I passed in Connect Centre. First, I thank God for the opportunity to do my PhD at Trinity College. I want to thank my first supervisor, Prof. Luiz da Silva, as, without him, I would not even start my PhD at Trinity. Luiz believed in me when neither I did, and I will always be thankful for that. Due to Luiz changing university, I changed supervisors though the change was fortuitous; Prof. Ivana Dusparic is the role model I have for my life. Ivana is patient, kind, humble, and always make you give your best. Her kindness and understanding during the COVID19 situation changed how I dealt with the PhD and my insecurities.

I also would like to thank the pos-docs who worked with me during these PhD years, starting with Boris Galkin, which motivated me to work with drones. I also want to remember Francisco Paisana, who supervised the spectrum sensing work and received me when I was still an intern. In addition, I had the chance to learn from Andrei Marinescu, Jacek Kibilda and Maicon Kist. All of whom contributed to my development as a researcher. In Connect, I learned not only from my supervisors but also from great researchers such as Ramy Mohamed, Joao Santos, Nima Afraz and Jernej Hribar. Thanks to all my colleagues that shared the lab for these years.

I had all the support from my family in Brazil, who always believed in me. I will forever thank my mom, Eliana, who was always supportive in every moment I needed, even not knowing anything about telecommunications. I also thank my dad for introducing me to the telecommunications world. My brothers Daniel, Everton and Leonardo for

always being funny and receiving me with open arms when I came to visit. To my step-sisters for giving me my lovely nephews, Davi, Eduardo and Manuella. My aunts Sonia and Solange who always pray for me, and Elisabeth and Arminda for being an example of people that valuate research.

I could not forget mentioning my lovely boyfriend, Cian O'Sullivan, who helped me through the difficult times and supported me in every way I needed.

I was fortunate to have all these people in my life during my PhD. All of them helped me in a way so I could keep doing it until the end.

Contents

List of Acronyms

3GPP	3rd Generation Partnership Program
3G	Third Generation Mobile Networks
5G	Fifth Generation Mobile Networks
AP	Average Precision
ANR	Automatic Neighbouring Relation
AMF	Access and Mobility Management Function
BS	Base Station
BLER	Block Error Rate
CNN	Convolutional Neural Network
CR	Cognitive Radio
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
DNN	Deep Neural Network
DQN	Deep Q-Learning
ECGI	evolved cell global identifier

ESN	echo state network
EVLOS	extended visual line-of-sight
FDD	Frequency Division Duplexing
FNN	fully connected neural network
gNB	next generation NB
GUE	Ground User Equipment
GP-KNN	Genetic Programming with K-Nearest Neighbours
GPU	Graphics Processing Unit
IoT	Internet of Things
IoU	intersection over union
ISM	Industrial, Scientific and Medical
LAA	license-assisted access
LoS	Line-of-Sight
LTE	Long-Term Evolution
LTE-U	LTE in unlicensed spectrum
mAP	mean Average Precision (AP)
MAC	Medium Access Control
mAP	mean Average Precision
MIMO	Multiple Input Multiple Output

ML	Machine Learning
NN	Neural Network
NR-U	New Radio Unlicensed
NRT	Neighbour Relation Table
LSTM	long short term memory
NLoS	Non-Line-of-Sight
OAM	Operations Administration and Maintenance
OD	Object Detection
OFDM	Orthogonal Frequency Division Multiplexing
PCI	physical cell identifier
PPP	Poisson Point Process
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
ReLU	Rectified Linear Unit
RF	Radio Frequency
RL	Reinforcement Learning
RForest	Random Forest
RSSI	Received Signal Strength Indicator

RSRP	Reference Signal Received Power
SDR	Software-defined Radio
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal-to-Noise Ratio
SON	Self Organising Networks
SVM	Support Vector Machine
TL	Transfer Learning
UAV	Unmanned Aerial Vehicle
UE	User Equipment
USRP	Universal Software Radio Peripheral
YOLO	You Only Look Once

Chapter 1

Introduction

Nowadays, it is common to hear news on Unmanned Aerial Vehicles (UAVs), not only as a toy but also performing roof inspections or streaming events. However, all these applications require Line-of-Sight (LoS) between the pilot and the UAV. If the UAV had the possibility of being controlled from any distance, the number of possible applications using them would even increase. Fifth Generation Mobile Networks (5G) are the latest generation of mobile networks, and one of its innovations is the support for different types of User Equipments (UEs), including UAVs. The research community is investigating what technologies and protocols are needed to make this possible. In this thesis, we explore how to improve UAV's Quality of Service (QoS) when they operate as one of the 5G or Long-Term Evolution (LTE) UEs.

In this introductory chapter, we outline the motivation for our research and state the main contributions of this thesis. The chapter is structured as follows. We begin by discussing the differences between a connected UAV and a standard UE. We then describe

the envisioned architecture considered in this thesis. Then we describe the main contributions of this thesis and list the publications associated with this research.

1.1 Motivation

As soon as the cellular mobile device started being part of society, it impacted our lives, changing the way we communicate. In the UK, 87% of adults have smartphones, accounting for almost 50% of the internet traffic [1]. 5G provides data more efficiently when compared to previous generations, enabling new use cases for the mobile cellular network. 5G is a paradigm shift from previous generations of the mobile network; it enables a massive amount of data transfer and a higher degree of connectivity, lower latency and several new user types, for example, autonomous cars and UAVs applications.

A UAV, also known as a drone, is an aircraft that does not need a pilot on board, although it needs some communication with its pilot. Nowadays, UAVs use the unlicensed spectrum for their communications. The most used Radio Access Technologys (RATs) are WiFi and Lora for general control with LoS [2]. The industry has also shown some concern about the UAV security and has invested in encryption to protect the messages exchanged between the pilot and the UAV. The security of transmissions in the unlicensed spectrum is a concern for all UEs that use this spectrum. Figure 1.1 shows an example of a UAV that can transmit in 2.4 to 2.48 GHz (unlicensed spectrum).

Even though UAVs are starting to be used for delivering goods and other applications, they still need LoS with the pilot or a pilot helper extended visual line-of-sight (EVLOS). Many countries, as stated in regulations in European countries and the United Kingdom

FIGURE 1.1: DJI-mini 2 [3].

[4], [5] expressly mention the need for LoS between the pilot and the UAV. LoS is needed because, to date, there is no other way to guarantee a continuous connection to the UAV, which is indispensable for its secure operation and localisation of the communication between the UAV and its pilot or controlling entity.

With the rise of UAV technology, UAVs may play the role of infrastructure for the 5G network (providing connectivity for ground users), or as the users of the network themselves. They can operate as a flying Base Station (BS) and provide extra coverage to

isolated areas, after natural disasters, for example. They can also be users of the network, making use of the cellular connectivity to provide security surveillance, search and rescue operations, building inspections (roofs, chimneys, siding), agricultural surveys, underwater inspections, mapping and surveying, delivery, streaming data at shows and events, etc. [6]. Considering all of these possible applications and several others which may appear in the future, it is essential that these new users can be connected to the 5G and future networks.

The connected UAV is a fundamentally new UE of the network, as its movement and altitude are different from those of the Ground User Equipment (GUE), which were the leading clients in the previous mobile network generations.

The mobile network, so far, has been designed for GUEs. A smartphone is an example of GUE, as much as an autonomous vehicle. These users are usually located around 1.5 meters off the ground and sometimes inside buildings [7]. The way these GUEs move usually depends on the street design and the speed they can achieve; for example, if a smartphone is inside a car, it can move much faster than if it is carried by a person walking on the street. However, all these possible behaviours can be easily predicted, as if the GUE is inside a car, it needs to move on the street subject to a maximum speed. If a walking person is carrying it, one can also predict the person is following a walking path and at an average speed. It means that the existing mobile network has had, so far, predictable user behaviours and could optimise its coverage and QoS to improve the UE experiences.

With the introduction of UAVs as a UE of the mobile network, there is a paradigm-shift from the perspective of UE behaviour, location and movement. For example, a UAV may

need constant connection with its pilot, or be transmitting data during all of its flight (in case of security and surveillance, for example), or periodically send its location updates to regulatory entities. These connection requirements are different from the ones a smartphone would have in its normal utilisation. Table 1.1 shows the requirements defined in LTE Release 15 regarding UAV connectivity. They specify, for example, a high altitude for UAVs to maintain connection when compared to the maximum altitude allowed by regulators in most European countries [4], [5]. To meet the requirements presented in Table 1.1 and the even stricter requirements which are expected in future releases, the network will need to adapt to be able to serve the connected UAVs.

Parameters	**Value**
Latency for traffic	50 ms
UL/DL data rate	60-100 kbps
Application data rate (UL)	up to 50 Mbps
UAV UE height	up to 300 m
UAV UE speed	up to 160 km/h

TABLE 1.1: UAV requirements in 3GPP Release 15.

The UAV also brings a challenge from the BS coverage perspective, as the coverage was designed for the GUEs. The network was designed to serve the GUEs with the best coverage, which means that the planned and optimised coverage is at the ground level. UAVs can fly much higher than building heights, in addition to having a less predictable movement pattern, as they can move in any direction.

Previous work, such as [8], [9], investigates the feasibility of using existing network infrastructure to provide reliable wireless connectivity for UAVs. These studies conclude

that currently deployed networks would need to adapt some of their design configurations, such as increasing BS heights [10] or changing the tilt of the antennas [11] so as to enable connectivity for UAVs. Redesigning the terrestrial network infrastructure may be unfeasible, and an adaptive solution on the UAV side may be necessary to accelerate the UAV integration into the network.

To model the performance and behaviour of UAV connectivity in order to devise a solution, one can use stochastic geometry as in [12]. Stochastic geometry is a study of random spacial patterns that uses statistical information of the environment for its models. However, as the UAV will fly in many different cities and with different paths, the researcher would need to have access to the statistical distribution of the features of the environment for each position the UAV takes. Also, if any changes happen to the environment, they could affect the calculations. That is why the use of Machine Learning (ML) is well suited to solve UAV connectivity issues [13], [14]. If the UAV is using the unlicensed spectrum (as they are nowadays), there is also the difficulty in achieving efficient access to the spectrum, as it is susceptible to jamming attacks and can be overused by greedy UEs. Works as [15], [16] investigate applications of intelligent RATs in unlicensed spectrum and their future trends. The authors argue that intelligent access to the unlicensed spectrum is fundamental for connected UAVs, and demonstrate that ML technologies are a suitable solution to access the spectrum opportunistically.

One way to possibly improve the UAV's QoS when connected to the mobile cellular network is to optimise its flight path. Several works in the literature have looked into the

problem of optimising the UAV's path with different objectives, like shortest path discovery so as to avoid building collisions [17], or considering battery recharging requirements [18]. There are even a few works that consider the UAV connectivity to the cellular mobile network when deciding the UAV's horizontal path [14], [19]. However, these works do not consider UAV dynamic height optimisation in their solutions.

Another way to improve the UAV's QoS is for the UAV to improve how efficiently it uses the spectrum. Nowadays, UAVs primarily use unlicensed spectrum, which is increasingly being shared by a variety of UEs. It is important that the UAVs manage to have access to the spectrum efficiently. In [20], authors present the issue that if no changes are made to the way UAVs communicate, UAVs can lead the unlicensed bands to serious spectrum scarcity and security issues. They also propose the use of cognitive radios for UAVs. Cognitive radio is a programmable radio that can be configured dynamically to use different wireless channels. It is commonly used to increase spectrum efficiency, avoiding interference and congestion. In case a UAV is equipped with such a radio, it can dynamically change wireless channels in order to guarantee its best QoS on the licensed or unlicensed spectrum. In [21], authors study the spectrum sensing for aircraft in order to not interfere with other transmissions with the command tower, assuming that the aircraft also will have access to a cognitive radio.

A possible way the UAV could acquire information to improve its QoS is by performing more detailed sensing of the spectrum in order to use it dynamically. A UE usually makes periodic measurements on the spectrum that are used by the BS to decide when the UE should perform a handover [22]. As the UAV faces a completely new and unplanned

environment, it could benefit from performing more detailed sensing in order to make a more reasoned decision about its mobility management[1], and consequently, improve its QoS. In the case of the UAV being able to operate on licensed and unlicensed bands, a spectrum characterisation mechanism could help the UAV to choose which band is the most advantageous to its application. Furthermore, spectrum characterisation can also provide information to the BS to take a centralised decision about the UAV mobility.

To address the issues identified with UAV connectivity, in this thesis we investigate the issue of how to adapt the height at which the UAV flies to improve the UAV's connectivity and how to effectively perform spectrum characterisation in shared spectrum. In addition to the challenges described above, we consider the challenge of providing UAV connectivity using the current infrastructure. We take this approach to understand what can be achievable to these UEs without additional infrastructure being deployed nor modified in order to accommodate connected UAVs. To evaluate our approaches, we make use of both real-world data and simulations.

1.2 Scenario of Interest

The envisioned architecture in our research consists of 3 main components, as shown in Figure 1.2: the connected UAV, the BS from LTE or 5G, and the GUEs. We also consider other users of the unlicensed spectrum, like sensors, WiFi users and routers, etc., when investigating the unlicensed spectrum usage.

[1] In this thesis when we refer to mobility management we mean the mobile cellular network handover management.

We define connected UAV as a UAV that is a UE of the cellular mobile network, connected by LTE or 5G technology. We assume the GUEs and BSs will behave as described in the 3rd Generation Partnership Program (3GPP) regulations, being able to operate in licensed and unlicensed spectrum. The UAV is equipped with an omnidirectional antenna that is used for the communication. The BS, when in a real environment, is divided into cells, each of them with a directional antenna[2] that employs Multiple Input Multiple Output (MIMO) technology. In the simulator environment, we assume a directional antenna for each BS.

We assume the UAV as a UE of the network does not have any extra modifications when compared to a GUE, as in [19]. The main difference between a UAV and a GUE is the 3D movement and height of the UAV that can be adapted.

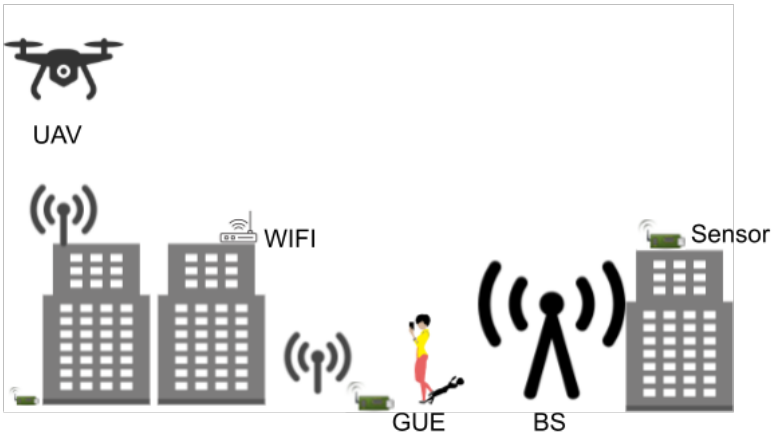


FIGURE 1.2: Network architecture considered in the thesis.

[2]More details on the antenna characteristics of real-world measurements are described in Chapter 4

Figure 1.3 illustrates how the UAV connection may arise. Usually, BSs antennas are pointed to the ground, where the GUEs stay. The angle β is the antenna's azimuth that can be configured mechanically or electrically by the network operators. This strategy is used so the main lobe of the antenna, which is the one where most of the transmission power is directed, would be covering most of the UEs. The signal coming from the serving BS is a desirable one, and the signal coming from other BSs are interference. Buildings, mountains and other objects can cause physical blockage, which helps avoid the interference from other main-lobes for GUEs. The side lobes usually do not generate interference for the GUE, as they are pointed to the sky. As there was no need to understand what was happening in the sky coverage, it was a subject that was ignored for long time.

With the use of connected UAVs, the air coverage became a topic of investigation. Researches from Qualcomm investigated in [23] that UAVs have coverage from the side-lobes of the antennas. Even though the side-lobes have a weaker signal power, as there are no obstacles between the UAV and the BSs, the strength is enough to provide connection to the UAV.

An issue that the UAV faces, that GUEs did not meet before, is that the UAV has LoS to many antennas at once. LoS is usually desirable when one wants a reliable connection. However, LoS with more than one BS generates interference and other challenges that will be discussed further in Chapter 3.

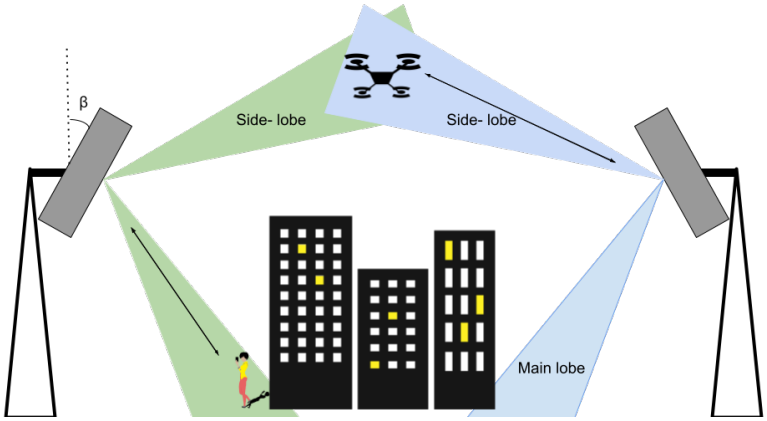


FIGURE 1.3: BS-UAV connection.

1.3 Research Questions

Motivated by the main challenges a UAV can encounter when it is connected to the mobile cellular network and the research gaps, this thesis aims to provide an answer to the research question of "How to improve the QoS of a connected UAV?". We propose that there is more than one way to answer this question, and in Chapter 3 we introduce the main challenges a mobile operator will encounter as more UAVs become UEs of the network. We propose several directions in which these challenges can be addressed on the network side. However, as the adaptations on the network side may take several years to deploy or may not be implemented by a network operator at all, we also present directions on how the UAV itself can improve its QoS or help the network to do so. In order to avoid poor coverage and other issues presented in Chapter 3, we investigate more deeply the biggest difference a connected UAV has to a GUE: its freedom to change its height. This narrows down the research question to "How to adapt the UAV height in order to

increase UAV's QoS?". Once the UAV avoids the poor coverage areas, if it is using the unlicensed spectrum it is essential to determine when there is a transmission or not, to share it. In order to not only use it, but use it optimally, we investigate the efficient use of the unlicensed spectrum (the one that is used for UAVs so far). This narrow down the research question "How to better characterise the unlicensed spectrum in order to enable more efficient spectrum usage?".

The research questions are described and enumerated as follow:

- RQ1: Which are the main challenges a connected UAV may encounter if deployed in a typical modern-day cellular network?
- RQ2: How to adapt the UAV height in order to increase UAV's QoS?
- RQ3: How to better characterise the unlicensed spectrum in order to enable more efficient spectrum usage?

In possession of the research questions, next we discuss the proposed solutions.

1.4 Main Contributions

Our contribution in this thesis is to investigate the main challenges associated with integrating UAVs into existing cellular networks and propose directions to improve UAV's QoS. In the proposed approaches, we utilise ML techniques to assist the UAV to characterise the spectrum and to take decisions of its height path, as illustrated in Figure 1.4.

Our main contribution point-by-point are:

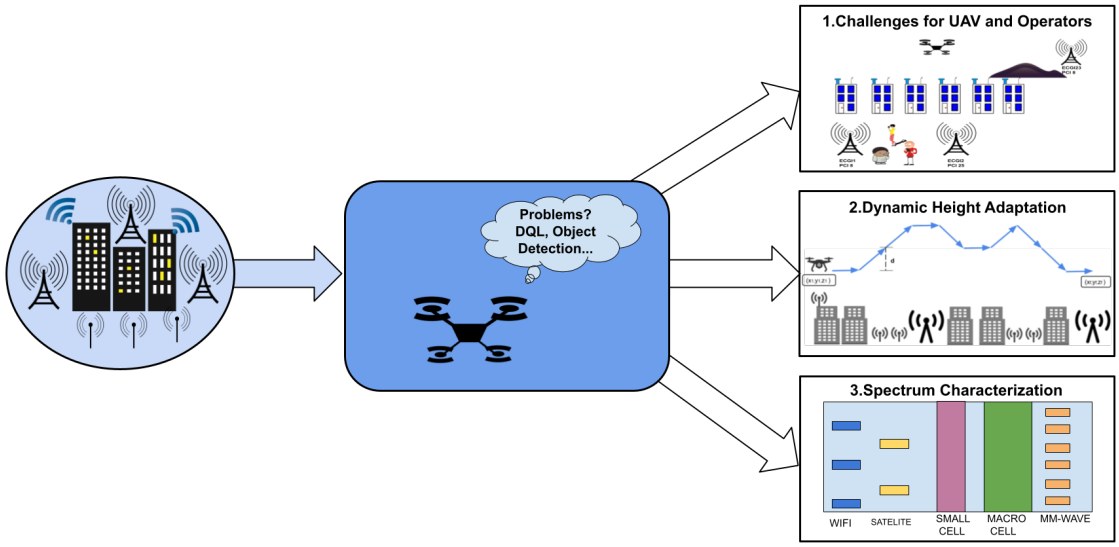


FIGURE 1.4: Summary of contributions.

- C1: it is a contribution provided to address RQ1. We investigate in detail the 3GPP regulations and analyse what could be an issue in the mobility management perspective, in case a connected UAV start being a UE of the mobile network without any change to the actual network configuration. We discuss the challenge of planning network coverage when considering coverage for flying users, and the physical cell identifier (PCI) collision and confusion issues that can be aggravated by these users. We also explain the future handover challenges the UAV might encounter, specifically the frequent number of handovers and the possibility that the UAV disconnects because of handover issues. We propose directions to solve the challenges a UAV could bring to the network, either in the network and the UAV, in order to

avoid main issues to UEs.

- C2: It is a contribution provided to address RQ2. We investigate how the UAV could improve its own QoS adapting its height. The height adaptation helps the UAV to maximise its own throughput as it moves through the environment. We propose novel Reinforcement Learning (RL) approach based on common UE measurements, as Reference Signal Received Power (RSRP) and UAV height. The UAV optimise its height in order to increase throughput. We show that the RL proposed model can adapt to different scenarios.

- C3: It is a contribution provided to address RQ3. As another way for improving an UE's QoS in unlicensed spectrum, we propose an approach that combines the application of a Convolutional Neural Network (CNN) model that makes Object Detection (OD) on spectrograms for classifying different RATs and a feature extraction component for characterising the RATs. We show that OD applied to real-world data is an efficient way to detect and classify different RATs under different levels of interference, and that the precision of the extracted features are defined by the size of the analysed image. The proposed approach can be useful in spectrum monitoring that is needed for unlicensed UEs as UAVs, for facilitating the coexistence of different RATs in a shared spectrum.

During the thesis we noticed lack of studies with commercial transmissions and data, to minimise it, we provide all the algorithms developed for analysing the real-world data, either for the UAV collected data in the Dublin city centre, as well for the one used for

OD with WiFi and LTE transmissions collected in Belgium. The algorithms are available to the community in our public available GitHub repository[3].

Labelled data is a challenge to OD as it has to be done manually. In order to run our experiments and help the research community, we provide a dataset of labelled spectrograms of real-world transmission of LTE and WiFi. This dataset can be access in our public available GitHub repository[4].

1.5 Outline

The remainder of this thesis is organised as follows:

- Chapter 2 *Concept Review* introduces the necessary background to understand the investigated challenges presented in this thesis. It explains concepts as mobility management in 5G, UAV in the unlicensed spectrum and ML concepts to understand the proposed approaches.

- Chapter 3 *Connected-UAV Integration to the Network: Open Challenges* answers RQ1 and presents C1. It introduces a deep investigation on the mobility management challenges a connect-UAV would encounter if no adaptations are done at the network and UAV level. Following the discussion about the challenges, this chapter presents analysis of existing literature addressing research questions related to ours, to motivate the literature gap our thesis is addressing

[3]https://github.com/Erikagpf/WiFi-LTE-commercial-data–labelled-for-OD
[4]https://github.com/Erikagpf/DQN-for-UAV-height-adaptation

- Chapter 4 *Adaptive Height Optimisation for Connected UAVs - Investigation based on real-world measurements* introduces a RL solution for RQ2 and presents C2 in an experimental scenario approach. The proposed solution is trained and tested on real-world data collected in Dublin city centre in two different locations.

- Chapter 5 *Adaptive Height Optimisation for Connected UAVs - A simulation investigation* answers RQ2 and presents C2 with an investigative solution proposed in Chapter 4. We have extended the evaluation to simulation environment, so that we can vary the building density and BS density and evaluate how it affects the UAV's QoS.

- Chapter 6 *Radio Access Technology Characterisation Through Object Detection* addresses RQ3 and presents C3. It examines the application of OD to not just classify the RATs in the spectrum, but also to extract features through the position of the transmissions on a spectrogram.

- Chapter 7 *Conclusion and future investigation* discusses the main conclusions from this PhD thesis and future directions for the research into connect-UAVs. It presents some other challenges not investigated in this thesis that are still open for investigation.

1.6 Dissemination

In this section, we list the dissemination of the research work during the PhD project. Here we list papers written during the PhD project and the research question to which they are related.

RQ1:

- *Fonseca, E.*, Galkin, B., Kelly, M., DaSilva, L. A., and Dusparic, I. "Mobility for Cellular-Connected UAVs: Challenges for the Network Provider," 2021 Joint European Conference on Networks and Communications and 6G Summit (EuCNC/6G Summit), 2021, pp. 136-141. I was the primary researcher in this paper, the non supervisor author (Kelly, M.) helped on the analysis of the challenges, and the supervisors helped with comments and improving of the content. This paper is presented in Chapter 3.

- Galkin, B., Amer, R., *Fonseca, E.* and DaSilva, L.A. "Intelligent Base Station Association for UAV Cellular Users: A Supervised Learning Approach," 2020 IEEE 3rd 5G World Forum (5GWF), 2020, pp. 383. My role in this paper was to discuss the content and write the Related work section of the paper, the non supervisor author (Amer, R.) helped on the analysis of the results and created the antenna model used in the simulation. The supervisors helped with comments and critiqued the content. This paper is cited in Chapter 2 and as a base of the simulation scenario.

- Galkin, B., *Fonseca, E.*, Amer, R., DaSilva, L.A. and Dusparic, I. "REQIBA: Regression and Deep Q-Learning for Intelligent UAV Cellular User to Base Station Association," IEEE Transactions on Vehicular Technology, 2021. I was the primary researcher in this paper, the non supervisor author (Amer, R.) helped on the analysis of the results and created the antenna model used in the simulation. The supervisors helped with

comments and critiqued the content. This paper is cited in Chapter 2 and as a base of the simulation scenario.

RQ2:

- Galkin, B, *Fonseca, E.*, Lee, G., Duff, C., Kelly, M., Emmanuel, E., and Dusparic, I. "Experimental Evaluation of a UAV User QoS from a Two-Tier 3.6GHz Spectrum Network," 2021 IEEE International Conference on Communications Workshops (ICC Workshops), 2021, pp. 1. My role in this paper was to discuss the content and analyse the data with the partners. I also helped describing the experiment. The non supervisor authors (Lee, G., Duff, C., Kelly, M., Emmanuel, E.) helped on the analysis of the results and conducted the operator portion of the experiment. The supervisors helped with comments and improving of the content. This paper generates the data used in Chapter 4, proving the experimental data for the analyse of the proposed height optimisation model.

- *Fonseca, E.*, Galkin, B., Amer, R., DaSilva, L. A., and Dusparic, I. "Adaptive Height Optimisation for Cellular-Connected UAVs: A Deep Reinforcement Learning Approach". To be submitted to Elsevier, Computer Communications, 2022. I was the primary researcher in this paper, the non supervisor author (Amer, R.) helped on the analysis of the challenges, and the supervisors helped with comments and development of the content. This paper is presented in Chapters 4 and 5.

RQ3:

- Fontaine, J., *Fonseca, E.*, Shahid, A., Kist, M., DaSilva, L. A., Moerman, I., and De Poorter, E., "Towards low-complexity wireless technology classification across multiple environments," Elsevier, Ad Hoc Networks, 2019, 91, pp. 101881. My role in this paper was to apply the CNN solution proposed in the same paper. This paper is presented in Chapters 2 as reference and used to compare the performance of the proposed method for RAT analysis in Chapter 6.

- Utrilla, R., *Fonseca, E.*, Araujo, A. and Dasilva, L.A.. "Gated recurrent unit neural networks for automatic modulation classification with resource-constrained end-devices," IEEE Access, 2020, 8, pp. 112783. My role in this paper was to discuss the possible ML techniques used in the literature and help on the experiments. Together with the comments and reviewing the content. This is part of Utrilla, R.'s thesis.

- *Fonseca, E.*, Santos, JF., Paisana, F., and DaSilva, LA.. "Radio Access Technology characterisation through object detection." Elsevier, Computer Communications 168, 2021, pp. 12. I was the primary researcher in this paper, the non supervisor authors (Santos, JF., Paisana, F.) helped on implementation of the approach and also reviewing the paper. The supervisors helped with comments and improving of the content. This paper is presented in Chapter 6.

Chapter 2

Background

In the previous chapter, we outlined our vision for UAVs as a UE of the mobile cellular network. In this chapter, we will introduce state of the art in mobility management in 5G, spectrum usage in 5G and Future Networks, and UAV movement optimisation.

Section 2.1, introduces the definition and explanation of UE mobility management in 5G networks. Section 2.2 introduces the 5G use of unlicensed spectrum. We then, introduce the ML techniques used in this thesis in Section 2.3. In Section 2.4, we conclude the chapter.

2.1 Mobility Management in 5G

In order for a UE to not lose its connection with a network once it moves outside the BSs coverage, it is necessary to establish a new connection to another BS. A BS usually has more then one cell, where each one has a different coverage area. Handover is the process by which a UE changes its serving cell. It is typically triggered when the UE moves out

of the coverage area of its current serving cell. Ideally, the handover should be seamless to the UE, such that it would not suffer any data interruption during the process. If a UE experiences multiple handovers, a handover delay might occur, resulting in substantial deterioration to the UE QoS [24]. To proceed with a handover, the UE needs to detect pilot signals from neighbouring cells. The list of neighbour cells is defined on the Neighbour Relation Table (NRT) that is stored in the connected cell. In 5G, this list is generated locally by Automatic Neighbouring Relation (ANR) based on UE measurements of RSRP from nearby cells. ANR was introduced in Third Generation Mobile Networks (3G) and was shown to reduce planning and operational costs for mobile networks providers [25].

Mobility management in 5G is performed by three main entities, illustrated in Figure 2.1 and specified in [22]. These are the Access and Mobility Management Function (AMF), next generation NB (gNB) (that is the BS equivalent as defined in 5G, which may comprise one or more cells)[1], and the UE. The AMF is at the core of the network, which is a central part of the mobile network, and the gNB and UE are in the Radio Access Network (RAN), which is responsible for connecting the UEs through radio links to the core of the mobile network. AMF is responsible for managing the UE registration, initiates the authentication, and for handling connection and mobility management for UEs. The gNB is a transceiver that provides connection to the UE; it has a connection to AMF via the NG interface and to other gNBs via the Xn interface. The last entity is the UE itself.

In 2G and 3G networks, the NRT is deployed as part of the operations and maintenance system, which is equivalent to Operations Administration and Maintenance (OAM)

[1]In this thesis, if we refer to the serving BS as gNB, we are considering the UAV connected to the 5G network. When we use the generic term BS we refer to any technology BS, not necessarily 5G.

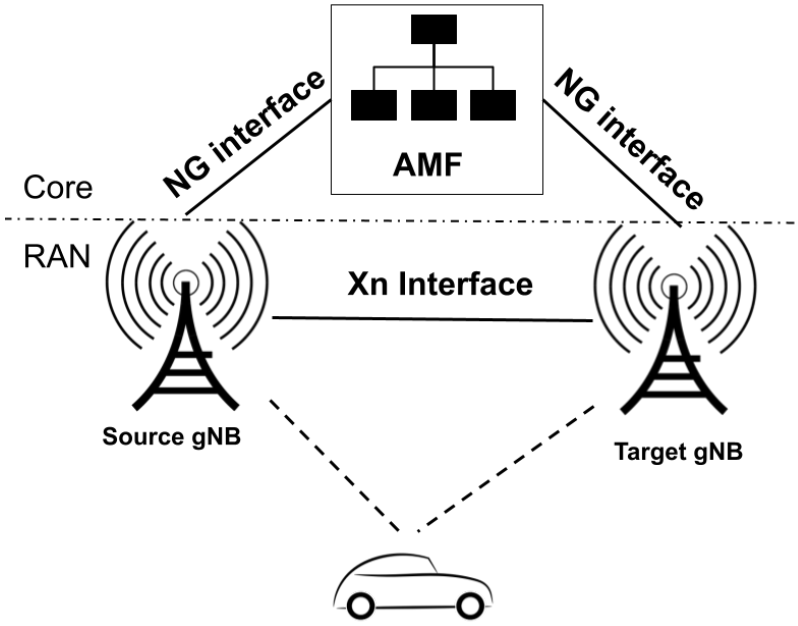


FIGURE 2.1: Entities involved in a UE handover.

in 5G. In 5G, the gNB has the permission to create new entries in the NRT. The ANR determines which cell should be added based on UE measurements and OAM updates. The UE can perform measurements to check for new cells, measure signal quality, determine if it needs to make a handover, or add a new cell to the ANR [22].

The purpose of this procedure is to transfer measurement results from the UE to the network, in order to allow the network to decide how to improve performance for the UEs and the network itself. The UE can initiate the measurements only after successful security activation in the network.

These measurements occur as often as determined by the gNB and vary based on the implementation of each operator. If the measurement is made in the same frequency band (intra-frequency), it can be done without any specific preparations to make the measurements. If the measurements are in another frequency (inter-frequency), the network needs to schedule a measurement gap where the UE stops receiving and transmitting data, changes to the frequency where it has to make the measurements, and senses it in order to find more suitable BSs. These gaps can affect the performance observed by the UE, if the UE is in dedicated mode (transmitting and receiving data). In idle mode, the UE can perform the inter-frequency measurements without impacting its QoS. The measurements are sent to the serving cell, which uses them to check for events to trigger a handover, or to add a new cell to the ANR, for example.

The information regularly decoded from a measurement by the UE includes the local identifier of the cell, named PCI in LTE and 5G. If the PCI is not in the NRT, then the

serving cell can send a message instructing the UE to sense the evolved cell global identifier (ECGI) of that cell, that is its global ID, in order to introduce this new cell into the ANR. To determine the ECGI, the UE needs to decode more data from the sensed BS, and to decode the ECGI it needs more than a single measurement gap. If the UE is in connected mode, actively receiving and transmitting data, the UE might not have time to perform the inter-frequency measurement and to decode the ECGI, as a result of which the UE might be disconnected.

The mobility events, defined by the 3GPP [22], may happen after the measurements are made and passed to the gNB. These events are described below. It is essential to understand them so we can understand how the handover occurs in detail and how it impacts the challenges a UAV will find when introduced to the mobile network. They are divided into intra-RAT, denoted as events A, and inter-RAT, denoted as events B.

- Event A1: The serving cell signal becomes better than an operator-defined signal quality threshold, i.e. the cell is providing good signal quality. This event is commonly used to cancel an ongoing handover procedure, to avoid a ping-pong effect from the handover.

- Event A2: The serving cell signal becomes worse than an operator-defined signal quality threshold, i.e. the cell is not providing a good signal quality. This event can trigger Inter-RAT measurements, for example, as new connectivity options must be considered for the UE.

- Event A3: The neighbour cell signal becomes better than the serving cell signal by a certain offset. This event can trigger the handover process to the neighbour cell.

- Event A4: The neighbour cell signal becomes better than an operator-defined signal quality threshold. This event is commonly used to trigger a handover. In this event, the handover is not triggered by the radio-signal conditions, but due to a network strategy specified by the operator, such as load balancing across cells, for example.

- Event A5: The operator defines 2 thresholds, refereed to as threshold1 (with lower value) and threshold2 (with higher value) in 3GPP. This event occurs when the serving cell signal becomes lower than threshold1 and the neighbour cell signal higher than threshold2. This event can trigger a handover based on the absolute measured signal strength values. This time-critical handover can be useful if the UE is leaving the serving cell coverage area and needs to handover, even if the target cell is not better by an offset than the serving cell to trigger an event A3.

- Event A6: The neighbour cell signal becomes higher by an offset than the serving secondary cell signal. In the case the UE has a multi-connection to more than one BS. It can trigger a handover from its current secondary cell to a new one.

- Event B1: An inter-RAT neighbour provides a stronger signal than an operator-defined signal quality threshold. This event may trigger a inter-RAT handover.

- Event B2: The operator defines 2 thresholds, referred to as threshold1 (with lower value) and threshold2 (with higher value). The signal from the serving cell becomes

lower than threshold1, and an inter-RAT neighbour provides a signal higher than threshold2. This event can trigger an inter-RAT handover.

Although the UE measurements are already described in the 5G standard [22], in LTE there are two other ways in which a cell may be added to an ANR [26], which might be adopted in 5G depending on the individual operator. The first alternative to the UE taking measurements is the UE transmitting an uplink ID, which should be unique locally. The cells that detect the signal above a certain threshold will add the serving cell of the source UE into their NRT. Another possible solution is to add a cell to the table once a UE loses connection and re-connects in a new cell. The new cell would add the last cell to which the UE was connected into its NRT. As this method makes use of a UE disconnection, it cannot be applied if the operator wants to provide seamless handover at all times.

This section reviews the mobility management concepts to understand the challenges a UAV may encounter related to handovers and sensing. We defined the components related to mobility management, their functions in the network and how the handover process occurs. Then, we introduced what is measured by the UE and the possible events that may trigger a handover. In the next section, we will investigate the use of the unlicensed spectrum by LTE and 5G networks and how the UAV may be included as a UE of this system.

2.2 UAV in the Unlicensed Spectrum

As presented in Chapter 1, 5G will operate in the unlicensed spectrum. 3GPP Rel. 16 [27] presents New Radio Unlicensed (NR-U) with UE operating in the license-exempt spectrum and therefore required to coexist with other RATs. NR-U are the new radios in the 5G that will work not just in the licensed parts of the spectrum but also unlicensed ones. There are two possible ways of using LTE and 5G in the unlicensed spectrum, anchored or standalone. The anchored, is the license-assisted access (LAA) method, that was first introduced in LTE-LAA. This method, the UE has a control channel in the licensed spectrum, although, its data is mostly transferred through the unlicensed spectrum. In this scenario, the sensing done is inter and intra-frequency, as it needs to sense in the unlicensed and licensed spectrum. In the standalone method, the UE operates only on the unlicensed spectrum. Its application is focused on neutral hosts (BSs's equipment that could be used by different operators depending on demand) and in the industry 4.0, where it is possible to implement the 5G technology locally without the need of having a dedicated bandwidth. The bandwidths envisioned to be used for the NR-U so far are in the 5 GHz, 6 GHz and 60GHz, and 3.5, 3.6, 3.7, 3.8 GHz where allowed.

To work in the unlicensed spectrum, the NR-U devices will need to coexist with the existing RATs dynamically. Spectrum characterisation could be used for better understanding of the unlicensed bands. As a UAV is a UE of the 5G network, it has the capability of using the license-exempt spectrum, and this is what has happened so far.

A challenge that comes with this coexistence is that usually the RATs used in cellular

networks are not designed to coexist with unlicensed spectrum users, as the mobile cellular RATs were created to work in the licensed spectrum. Consequently, there is research in how to adapt RATs that were initially designed to use license spectrum, to allow to share the unlicensed spectrum with unlicensed UEs in a fair manner.

The unlicensed spectrum consists of multiple of radio channels that can be used by a transmitter without previous authorisation. There, devices can select which channels to transmit on, but before transmitting they need to determine the presence of other users on the radio channel. RATs like WiFi and Bluetooth, for example, were created to act in the unlicensed spectrum. Therefore, taking WiFi as an example, it has a mechanism to avoid collision between users and adapt in a situation where the spectrum is overloaded, using a technique called Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA). To apply this technique, WiFi transmitters need to sense the spectrum before transmitting. CSMA/CA, also tracks the number of re-transmissions and collisions during its transmission, to understand how overloaded is the spectrum. Being able to sense the spectrum is a necessary step to avoid a collision when using the unlicensed bands.

For communication in cellular mobile networks, the LTE and 5G standard does not specify any sensing before transmitting. The measurement reports are used only with mobility management purposes. In the licensed spectrum, the network operator owns the spectrum and manages the access through the scheduling of the spectrum resources. The scheduling is performed by a centralised entity, meaning that sensing and collision detection by users themselves is not necessary. Some researchers [20], [28] consider a central entity for unlicensed spectrum too, where all the unlicensed users would connect to

this central, which would manage the spectrum. The implementation of such spectrum sharing solution would need the cooperation of all unlicensed UEs and also, the modification of many of them to work in such a centralised approach, as the WiFi and Bluetooth.

An alternative to the centralised approach is the decentralised approach, where each UE senses the spectrum in order to avoid collisions with other transmissions. A common approach from the unlicensed UEs is to have a Cognitive Radio (CR) to allow the use of different bands of the spectrum. The common steps a unlicensed UE or BS needs to implement before using the network in a distributed manner are:

- Spectrum sensing: as the use of the spectrum should occur when there are no other transmissions, it is crucial to sense if there are transmissions occurring. This is considered the most important function of the use of the unlicensed spectrum [29], [30].

- Spectrum decision: this function receives information from the spectrum sensing function and decides which is the best channel to use. For this, the spectrum decision function considers also the type of UE and its requirements, such as latency and rate.

- Spectrum sharing: once the channel is chosen, this function decides when to transmit in order to avoid interference and to have a fair use of the spectrum.

- Spectrum mobility: in case the chosen channel does not provide the requirements of the UE anymore, this function would perform the mobility to another channel, once it is decided by the spectrum decision function.

In this thesis, we investigate the spectrum sensing function in order to not only sense the spectrum, but also, characterise it. The characterisation provides more information if the channel is in use or not. Our approach provides information about the duration of the transmission, the bandwidth they are using, their centre frequency, and the RAT used for them. This information can be used by the spectrum decision and spectrum sharing functions in order to take a more informed decision. We do not investigate the other functions of the cognitive approach because they are likely to be highly application oriented. However, all applications could get advantage of the proposed spectrum characterisation.

In the works of [20], [28] it is introduced several reasons for the connected UAV to use the unlicensed spectrum in a dynamic approach (cognitive UE). These reasons are described bellow:

- Security: UAVs are considered critical users as it needs continuous connection to its pilots. Conventional attacks to these users are jamming attacks, that is a high interference to a spectrum band, or the overload of that band. In this attack, a UE using WiFi for example, increases its energy consumption due to package re-transmission. When a cognitive UE is under this attack, it would change its channel as it would understand that the channel is busy.

- Energy efficiency: due to packet loss of over-crowed bands, the UAV presents a high re-transmission rate [20]. Although, the opportunistic use of licensed and unlicensed spectrum allows the UAV to make a smarter use of the spectrum. [31]

presents a method to maximise energy efficiency based on optimisation of the physical and Medium Access Control (MAC) layers. Authors conclude that the possibility of using more channels increase the efficiency in bits/joule and bits/second throughput.

- Opportunistic spectrum use based on UAV requirement: depending on UAVs application, it can use the spectrum opportunistically. If a UAV is being used for a real-time video transmitting it needs high bandwidth, but tolerate packet loss. In the case of connectivity being required for real-time control of its path and operation, it needs high reliability (low packet loss) and low latency. The UAV can live-stream a video, and once finished, could move to a lower band to transmit a file, for example. These features can improve communication performance of 5G networks [28].

In this section, we introduced 5G in the unlicensed spectrum and the challenges it may encounter when sharing the spectrum dynamically. We present how the usual communication happens with UEs using cognitive radios and the advantages for a UAV to use the unlicensed spectrum dynamically. In the next section, we will introduce the ML concepts and techniques needed to understand the proposed approaches in this thesis.

2.3 ML Techniques for UAV Integration

To understand the solutions proposed in this thesis, this section introduces some ML concepts, techniques and strategies. ML are algorithms that can improve performance through data or experience [32]. ML algorithms enable the system to make predictions,

decisions, or classifications without being explicitly programmed to do it in a specific way. ML algorithm development and deployment usually consist of two stages: training, where it learns values to its internal parameters in order to provide a good enough output; and validation, where the model is tested against a new unseen dataset and it is evaluated how good the model can perform.

One of the earliest applications of ML techniques was Self Organising Networks (SON). Since early studies of SON [33] in 3G networks, ML techniques are being applied to define the NRT. The application of ML is growing with the newer generations as they are focusing on automation and even more independent autonomous configuration as show the first works focusing in 6G [34]–[37]. This subject is so important that it is being a focus in the research community to investigate how and which methods can be applied to several issues in mobile networks, such as Internet of Things (IoT) [38], mobility management [39], and traffic forecast [40].

The rest of this section is divided in the concepts necessary to understand the proposed solutions in this thesis. We detail two ML techniques, supervised and reinforcement learning. In supervised learning, the model is trained with a dataset that should represent the environment where the model will be used. It is usually validated with a smaller unseen dataset that also represents the environment it will be used. The model has access to a dataset of inputs and corresponding outputs, and the model learns a relationship between them. In RL, the model learns through experience interacting with the environment and learning the best policy to achieve its objective. Therefore, RL does not

rely on the availability of an existing dataset. Instead, the model learns through the interaction between the agent and the environment and observing the results, thereby through trial and error to find the relationship between the given observations and the resulting outputs.

2.3.1 Supervised Learning

A supervised learning model learns the relation between input and output based on labelled data. Labelled data is a group of samples divided by an associated feature with one or more labels. Labelled data is usually applied for classification, prediction and regression, such as image classification, pattern recognition, and natural language processing [41]. In mobile networks, supervised learning is used to investigate many mobile networks problems, such as Orthogonal Frequency Division Multiplexing (OFDM), channel equalisation, cognitive radio signal classification, cognitive radio channel adaption, and beam-forming configuration [42].

A Neural Network (NN) consists of connected mathematical functions named neurons, each of them producing a sequence of real-valued activation [43]. It has an input layer, at least one hidden layer and an output layer. A layer is a collection of neurons on a specific depth in a neural network. The connection between neurons is called weight. The input layer is activated by information about the environment, and the other neurons are activated by weighted connections from the neurons in the previous layer. Learning is finding weights that make the NN behave as desired.

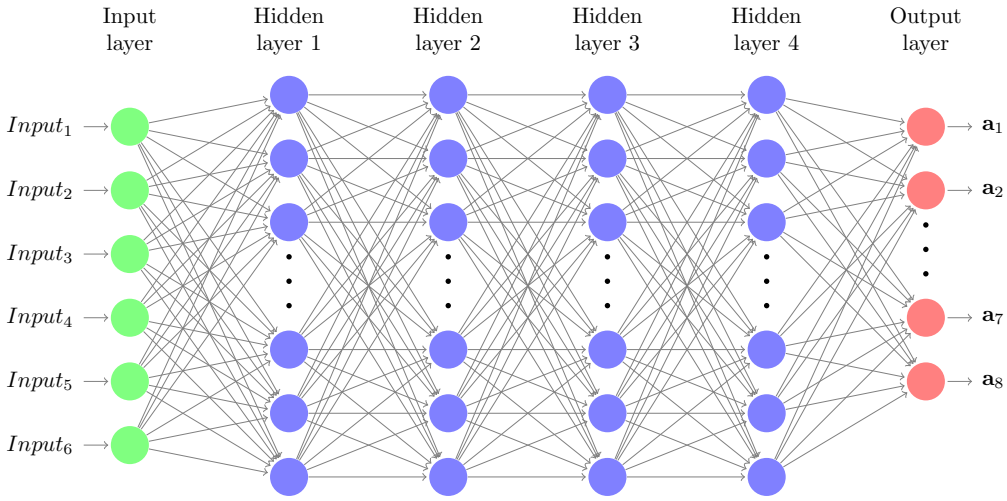


FIGURE 2.2: Graphical representation of a DNN example with 6 inputs, 4 hidden layers and 8 outputs.

Deep Neural Network (DNN) is a NN that has more then one hidden layer. The complexity of the model grows with the number of layers in it. However, deeper models started to accomplish good accuracy in more complicated tasks [44]. It discovers relations between large datasets applying backpropragation to adapt the model internal parameters. For classification tasks, more layers amplifies the input aspects that can be analysed, discriminating some aspects and suppressing others. Figure 2.2 is an example of a DNN model with 6 inputs, 4 hidden layers and 8 outputs. An example of a DNN model application is the real-time resource allocation of optical fibres. As input, the model could have the past years of used resources, detailing information as time of the day, weekdays, and months and output the action of allocating what the model predicted as necessary to attend the demand for that time.

Fully connected networks are a DNN where each neuron is connected to all neurons in the next layer. CNNs are a regularised version of fully connected networks. The regularisation happens so the model can adapt its internal parameters without overfiting to the training data. Overfitting refers to when the model learns how to classify or predict the training set, but struggles to do the same in the validation dataset. It happens because the model did not learn how to analyse the data, but adjusted the weights to analyse only the training data correctly and cannot generalise its learning. Underfitting is when the model does not learn even how to classify or predict the correct outputs to the training data.

The activation function is the function that models the extent to which a particular input impacts the resulting output value. If it decides a relevant impact, the neuron can be activated or not. After each layer with weights, non-linear activation layers are used. As these layers are not linear, it enables the CNN model to learn complex non-linear relations. Rectified Linear Unit (ReLU) is an example of an activation function and introduces non-linearity to the decision function. It transforms values in the activation map below 0 to 0, converting all the input values into positive numbers. As a result, it trains faster than other NN without decreasing significantly the accuracy [45].

A CNN has different layers, and each has the purpose of analysing the data in a different way. The first layers learn and extract high level features, and the deeper layers learn and extract the low level features [46]. Bellow we describe the layers mostly used in this thesis.

- Convolutional layer: it is comprehended as filters, also named kernels, with a receptive field that extends the depth of the input of this layer. Each kernel convolve

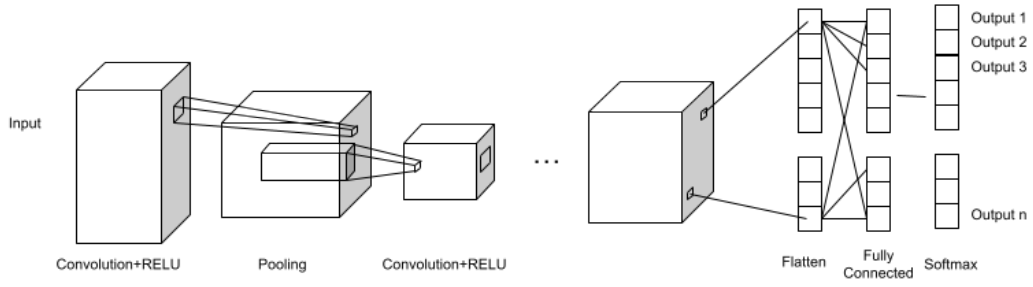


FIGURE 2.3: Example of an CNN model.

to the input, what produces a 2-dimensional activation map for each filter. Consequently, the kernels are activated only when they detect a specific type of feature [47].

- Pooling layer: it is a layer that applies non linear down-sampling. One has to define which function will be applied in the pooling layer. The objective is to reduce the size of the representation of the data, decreasing the computational power to analyse, without losing important information. It also avoid overfiting. It is common to insert a pooling layer between convolutional layers.

- Fully connected: it is a feed forward neural networks. Fully Connected Layers form the last layers in the network. They are used before the output, where all the neurons are connected to the previous layer.

Figure 2.3 illustrates an example of a CNN with convolutional layers, pooling layers, a fully connected layer and the output layer.

The last layer of a CNN is always the output layer that calculates the prediction error generated by the CNN model over the training data. The loss function calculates this prediction error. The loss function informs the network how close the prediction is to the actual label. This error is then optimised during the learning process. Softmax loss function, or softmax activation function and cross-entropy loss function, is a well used loss function that uses softmax activation in the output layer. The softmax loss function measures the performance of a CNN when the possible output is only one class, which means probability $p_i \in \{0, 1\}$. p_i is the probability of the output be from a specific class.

In order to train a CNN model it is necessary to have a significant amount of labelled data, which is one of the most critical issues of implementing this technique [48], [49]. Usually, to have labelled data, one needs to have a huge number of examples and label it manually. To decrease the amount of training data several techniques were created, such as Transfer Learning (TL) and create synthetic data.

In our approach to perform object detection for spectrum characterisation, we need to use TL to decrease the number of labelled data to train our model, so we describe this technique in more detail below.

Transfer Learning

One way to save computational power and time during the training stage of a CNN is using TL. TL relies on the partial reuse of a previously trained model (trained on a different set of tasks) for addressing a new task. This implies retraining an existing network, typically by fine tuning the weights from the hidden layers close to the output layer, to make

the network more suitable for the new task. As such, the first layers, which are typically good at extracting basic features such as edge detection in computer vision tasks, are also reused for the new task. TL significantly decreases the amount of data required for the training process and, consequently, the duration of the training.

The application of TL requires the choice of a previously trained network as a starting point. A broad range of pre-trained networks already exists; these are suitable for different problems, e.g., predictive text, speech recognition, and image object detection [50].

We applied TL in our object detection approach for spectrum characterisation. As our task required object detection techniques, we discuss those in the next section.

Object Detection

Object detection is a computer vision task that detects visual objects of a specific class in images. It provides the class of the detected object and where it is positioned [51]. Objects have different features that determine them, for example, a car has wheels, or a table has legs and a top area. An object detection algorithm is able to detect these features and recognise a class for the objects. Some of them provide a bounding box around the detected object, which determines the object's coordinates in the overall picture.

There are two main categories of object detection techniques, the ones without CNN and the ones with CNN. In order to apply the object detection techniques without CNN as in [52]–[54], it is necessary first to define the features to be detected[2], then use a classification method. However, with the CNN approaches, one may apply only the CNN to

[2]In this thesis when we mention "feature detection" it is related to the features of the transmitted objects on the spectrogram.

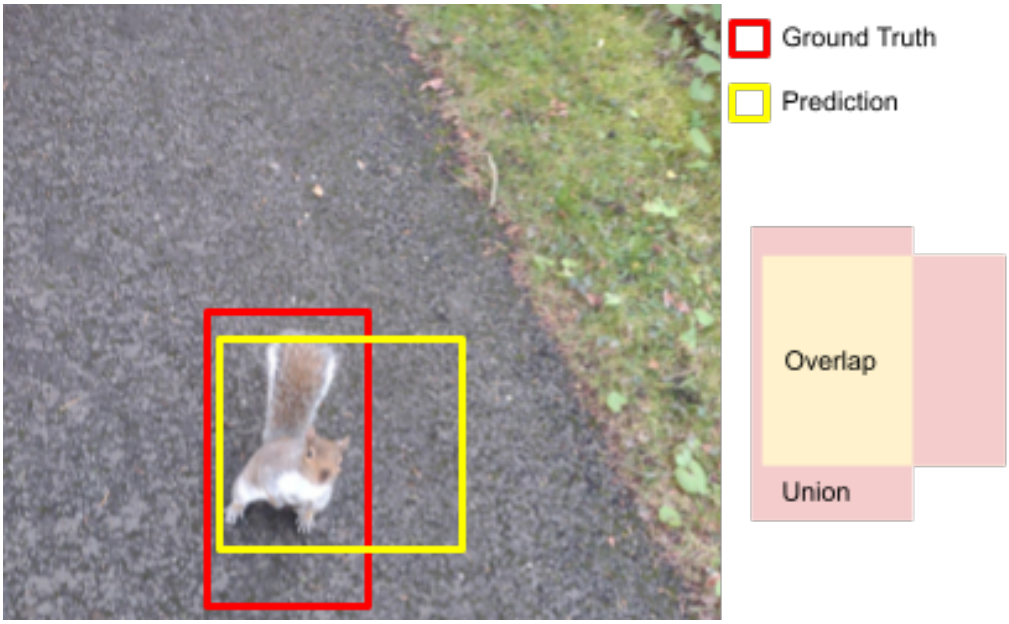


FIGURE 2.4: Example of ground truth and prediction bounding boxes.

perform the detection and classification. Another advantage is that the model may find detective features that the human training it would not see, which can achieve excellent performance. The downside of the CNN is that for this method to learn, it needs a significant amount of labelled dataset, and this will probably be required to be done manually.

We now specify metrics used for evaluation, as required for understanding the evaluation in Chapter 6. The usual ways to access the classification accuracy of an object detection model is with the precision metric. Precision is defined in object detection as the

percentage, among the detected objects, of correct classifications.

$$Precision = \frac{\text{Correct Object Classification}}{\text{Correct Object Classification + Incorrect Object Classification}} \tag{2.1}$$

However, in object detection, a model can detect any number of false objects in an image, which means there is an infinite number of possible incorrect detection. To tackle this issue, in object detection, the metric precision is used to express how reliable are the predictions from a model, i.e., it illustrates the percentage of the predictions' correctness, as shown in Equation 2.1. Furthermore, for estimating the number of misclassifications, we can simply calculate $1 - Precision$.

It is considered that the object that was detected is the same object that is in the ground truth if the intersection over union (IoU) of the bounding boxes is over 0.5. IoU is the area of overlap divided by the area of the union of the ground truth bounding box and the classified bounding box as described in Equation 2.2 and in Figure 2.4. In Figure 2.4, we can see that the prediction matches with the ground truth, although it varies a bit in the covered area, which is common with object detection models. In other words, a labelled data point for object detection task consists of the box around each object and its class. The IoU measure how well the model-generated boxes overlap with the labels.

$$IoU = \frac{\text{Area of overlap}}{\text{Area of union}} \tag{2.2}$$

Recall, specified in Equation 2.3, measures how accurate is the classification of the model. The area under the measured precision versus recall curve is smooth is the Average

Precision (AP) value. The mean Average Precision (mAP) value is the mean of all APs.

$$Recall = \frac{\text{Correct frame classification}}{\text{All frames}} \tag{2.3}$$

In the field of object detection, the evaluation of the efficacy for generating accurate bounding boxes resorts to a metric named mAP. This metric was introduced during the PASCAL VOC 2012 [55] competition and have been used to calculate the accuracy of an object detection model. mAP is the area under the curve precision-recall.

The first step for the calculation of the mAP is the calculation of the AP for every class of each model. The ground truth of the object position is necessary to make this evaluation. The model predicts bounding boxes that are sorted by decreasing confidence and are assigned to the ground truth files.

2.3.2 Reinforcement Learning

RL is a ML technique that applies a trial and error process where an agent performs actions and interact with the environment. At each moment, or in a simulation time step, the agent has access to the state information of the environment and acts from this given state to a new one. The specific action may have a reward or not. It learns from interaction with the environment to achieve long term goals [56]. RL can provide an output without a detailed description of the environment, which makes it easier to adapt to not complete information. RL was studied in the theory of optimal control [57]. This theory mainly investigates the existence and characterisation of optimal solutions, especially in the absence of a mathematical environment model. It can provide a solution without detailed

data of the environment.

The main components required for a RL representation of the problem are an agent, state space, action space, and environment reward. We define these below:

- Agent: The component that observes the state information and takes actions accordingly.
- State Space S: A set of relevant observations from the environment that the agent receives before and after taking an action.
- Action Space A: The set of all possible actions that the UAV can take.
- Reward R: the primary goal of the approach should be reflected in the reward.

Q-learning is common RL algorithm. Q-learning is an algorithm where agents learn how to optimally act, using an incremental method for dynamic programming. Its implementation successively improves the quality of actions at specific states. It is showed in [58] that the algorithm converges to the optimal action-values if the actions are sampled in all states. In classical Q-learning a Q-table would be learned, which maps the Q-values of each action for a given state. Q-value is the expected discounted reward for executing a specific action to a particular state considering a long term reward, and following a policy. The objective of Q-learning is to estimate the optimal policy. A Q-learning algorithm learns a policy, which tells the agent which action to take in a given situation. Q-tables are tables with one entry for each state (state-action pair) and are called tabular methods.

This technique utilises rewards to train the algorithm to take the appropriate action for a given state. A reward is a scalar value received after the agent takes an action and

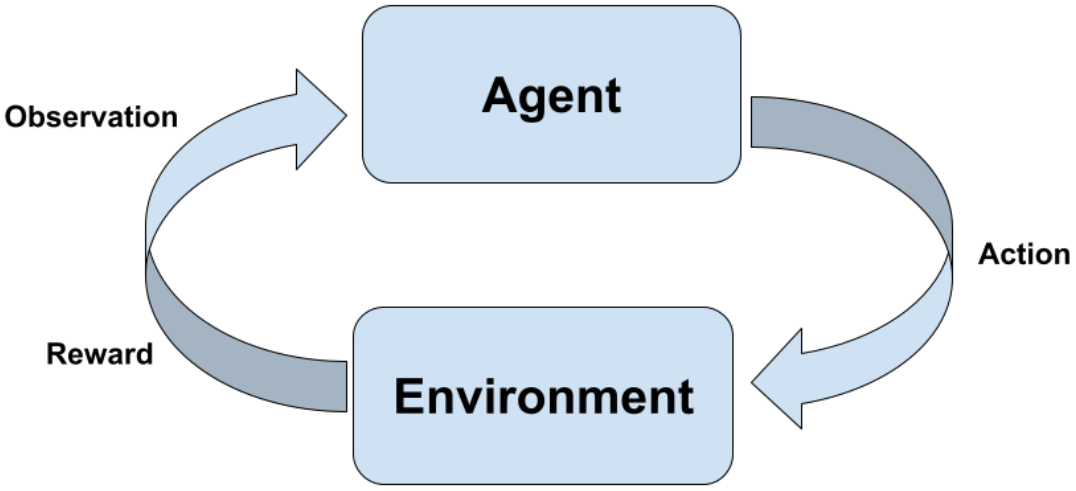


FIGURE 2.5: Reinforcement learning cycle.

transitions to a new state. The agent uses the reward values to determine which action is the most appropriate for its given state. The algorithm calculates the quality of each action on a specific state, the Q-value, in order to learn which is the best action on that state.

The RL cycle is described in Figure 2.5. The agent takes an action a in state S and receives environment feedback in the form of a reward R; its goal is to learn the policy, i.e. action for each state, that maximises the long term cumulative reward. To make more efficient use of data samples gathered in the environment, we adopt experience replay [59], which is a technique that uses a buffer to store experiences so they can be randomly sampled and reused during the training.

During the training, a RL algorithm needs to explore the possible actions it can take in

order to find the optimal policy in a trial and error search for a good policy. These stages of exploring possible actions and applying the optimal policy are defined as exploration and exploitation, respectively. If one uses its policy to calculate the possible actions, these actions are called greedy actions, and it is determined that you are exploiting your current knowledge. The feedback that indicates how good is the greedy action cannot determine if that action was the worst or the best possible. Also, if the action is always random to explore the possibilities, one could know the optimal action to take, although it does not mean that it would randomly select that action. In RL these two stages, exploitation and exploration, are balanced depending on a strategy.

An ϵ-greedy policy is one of these strategies. It is a simple strategy to sometimes explore the environment, with probability ϵ. We apply decaying-ϵ-greedy, where ϵ decreases its value at each action taken. An advantage of ϵ-greedy methods is that the likelihood of taking a random action decreases with time, as considering an infinite amount of time, all the actions will be sampled an endless amount of time, ensuring the algorithm convergence, however, with less exploration when the policy is already trained.

In some practical cases, there are more states than could be entries in a table. Therefore, the functions must be approximated using a more compact function representation for these cases [56]. Deep Q-Learning (DQN) is a DNN model in the RL field. The advantage of DQN is that it uses neural networks to approximate the function that maps a state-input pair to a Q-value. This makes it valuable for scenarios where the state or action space is too large to apply tabular RL.

2.3.3 Hyperparameters

The policy training itself requires fine-tuning parameters related to the learning rate and convergence of the model output, known as hyperparameters. Hyperparameters are parameters chosen before the training process that control the learning process. We detail the hyperparameters used in this thesis below:

- Learning Rate: is a tuning parameter used in optimisation algorithms that determines the step size taken in each interaction while moving to a minimum loss function [60].
- Epoch: indicates the number of passes of the entire dataset the ML model completes. If a model is trained with too many epochs, it can overfit the training data, while if a model uses too few epochs, it might not learn the necessary features to perform the classification.
- Mini-batch: is a part of the dataset used to update the network's weights. The first approaches in ML used the entire dataset to update the weights in the network; however, the work of [61] argues that this update should use a minor part of the dataset, called a mini-batch. Generally, minibatch values used in practice have values between 2 to 64 samples.
- Optimiser: is the function that modifies the weights of each neuron with the purpose of minimising the loss function.

Future details on specific values of these parameters in our evaluation and design choices that led to their selection are discussed in later chapters.

2.4 Chapter Summary

In this chapter, we introduced the background needed to understand both our proposed solution as well as the most closely related literature. We presented the mobility management in 5G to understand the mobility challenges presented in Chapter 3. We discussed the idea of the use of unlicensed spectrum by GUEs and UAVs. Then discussed the ML concepts needed for the proposed solutions, specifically DQN, TL, object detection, and RL.

Understanding the process of mobility management is essential to understand the challenges a connected UAV will face. In this chapter we reviewed the main architecture involved during the handover process and how it is defined for 5G networks. It is important to notice that the UE has a crucial importance on its own mobility as it makes the sensing and the measurement report. Based on this information, the mobility of the UE is defined.

The use of unlicensed spectrum for licensed UEs is a considerable new perspective, and one of the biggest challenges of this new approach is to share the spectrum fairly with the other UEs. RATs that were created to work in such spectrum are already adapted to share it in a fair way, although, there is the need of adaptation of the licensed spectrum RATs to use it. How the licensed RATs as LTE and 5G will share the spectrum is still being investigated, but as in other specifications, the operators can implement extra features to improve its performance. We believe that with more information of the spectrum usage these licensed UEs and BSs will be able to plan its transmissions in advance and also, make a more efficient use of the spectrum.

Finally, to understand the designed solutions in this thesis, we present present specific ML techniques used in this thesis. ML has proven to be an efficient way to learn how to make a decision for complex tasks based on analysing large amounts of data. In this thesis, we needed two types of ML methods, the object detection techniques, which are a subset of supervised ML, and the RL techniques. In the first, we wanted to characterise the spectrum to answer RQ3. We realised that a human looking at a spectrogram would be able to analyse the spectrum and understand it if it knew which are the RATs transmitted there. Based on this, we looked for a technique that could make classification and detect the position of each object. The ML object detection technique was the most appropriate as it performs classification and detection in an image. As the amount of data was not enough to the model learn by scratch, we applied TL.

The UAV movement optimisation needs interaction with the environment, because depending on its decision, the environment characteristics will change to the agent. As the problem itself requires interaction with the environment, the RL solution was the most appropriate solution for RQ2. Now that we have basic concepts, we move in a more detailed analysis of open challenges and most closely related literature.

Chapter 3

Connected-UAV Integration to the Network: Open Challenges

3.1 Introduction

Integrating UAVs to the mobile network represents a paradigm shift from providing connection to GUE. 3GPP created several working groups in the LTE technology [22], [62] and 5G technology in the 5G Enhancement for UAVs [63]–[66]. These working groups investigate the capability of the network to provide connection to the UAVs, research which requirements a UAV should have to be a UE of the network, and study the main challenges the network and the UAV may encounter. In this chapter, we investigate challenges a network operator may encounter when dealing with UAV's mobility and what would be the impact of a UAV on the network, as introduced in RQ1 "Which are the main challenges a connected UAV may encounter if deployed in a typical modern-day cellular network?". This chapter therefore provides the contribution C1 that is the investigation

of the challenges a network operator may encounter with the introduction of the UAV as a UE and introduce two possible approaches to allow the increase UAV's QoS without change the network configuration. This investigation also has lit RQ2 and RQ3.

This chapter comprises the following sections: We present the challenges a UAV may encounter if it starts using the actual mobile network in Section 3.2. The challenges are divided in coverage, PCI and handover challenges. To support our challenges, we analyse experimental data using real-wold measurements. We then propose a few directions of solutions to be implemented on the network. In section 3.3, we present the state of the art for our two different approaches, UAV height adaptation and spectrum characterisation. In Section 3.4, we conclude the chapter by discussing why it is essential to address the proposed challenges and why the proposed solutions address these challenges.

3.2 Network Challenges

UAVs were introduced as a new type of user of the cellular network in LTE and they are expected to increase in numbers and applications in 5G networks and beyond. As the LTE and 5G networks will work in unlicensed spectrum in addiction of the licensed spectrum, we believe that the challenges of a connected UAV can increase compared to when it is only connected to the licensed spectrum, as it will also need to co-exist with other UEs using different RATs. However, in this section we will present challenges that may occur in both bands, and are inherent to the band of operation.

One of the biggest challenges for connected UAVs is the presence of simultaneous LoS channels with several cells which may be far away. In [23], authors demonstrated via

simulations and experiments that a UAV can sense significantly more cells than a GUE. They also demonstrated that UAVs are capable of detecting more distant cells, compared to GUE. The fact that UAVs can detect a larger number of cells across a greater area means that the network should treat the UAV UE differently from a GUE, in terms of mobility management.

The principal mobility-related challenges that a UAV can introduce to 5G networks operators are discussed in this section.

3.2.1 Coverage challenge

Before the cellular network starts its operation, the operators need to plan the geographic locations of the BSs, along with configuration parameters such as their antenna azimuth and mechanical tilt. If UAVs become a significant user of the network, they ideally should be taken into consideration from the planning stage of network deployment.

Network coverage planning is essential to avoid interference and unnecessary handovers. For the previous generations of cellular networks, the coverage was planned only for GUEs, and the main lobe of the BS antennas was often the only one taken into account. For the next generations of cellular network, the coverage needs to be planned to also include connected UAVs, and needs to consider what kind of network coverage will be provided in the air, taking into account how the main lobe will perform but also the side-lobes. A common way to plan a cellular network is by using software tools that consider 3D maps of a given area and antenna radiation patterns. To integrate UAV users,

the tools used to plan the network coverage need to be adapted to consider antenna side lobes and should also project the signal propagation into the sky during its simulations.

3.2.2 PCI challenges

Another critical part of network planning that becomes harder with the introduction of flying UEs is the PCI distribution. The flying UEs can exacerbate PCI confusion and collision, which have been reported in LTE networks and persist for 5G networks. Usually, the PCI planning is made to allocate identical PCIs to BSs that are distant from each other, to ensure that a UE will be unlikely to detect the same PCI being transmitted by more than one BS at a time. However, considering connected UAVs, it will be necessary to understand the air coverage in advance to plan the PCI distribution.

In Section 2.1, we introduced the events triggered after the RSRP measurement reports realised periodically by a UE. The first piece of information a UE senses about a neighbouring BS is the PCI, that is the local cell identifier. Each cell in 5G or LTE has its own PCI. If the PCI assignment is poorly planned, it can affect the handover process and delay the downlink synchronisation. Another possible consequence is increased Block Error Rate (BLER) and decoding failures of physical channels. In LTE, there are 504 unique PCIs, compared to 1008 in 5G. If there are different tiers of the network, the network needs to divide these PCIs between each tier.

Consider a two tier network with macro-cells and small-cells, for example. The PCI values contained in set A will be reserved to the macro-cells and those in set B for small-cells. A and B have no intersection. This rule cannot be violated inside the same network.

This division decreases the number of possible PCIs for each tier, which can aggravate the issue of PCI availability. Due to the fact that the GUEs usually connect to cells that are close to them, with good network planning it is possible to avoid most cases of PCI collision and confusion for GUEs.

Figure 3.1 illustrates a well-planned network, where the same PCIs have a significant distance between them, which means that PCI confusion is not likely to happen for GUEs. The main issue occurs when a UAV flies overhead, as it senses more distant cells that can have the same PCI as the serving cell, which results in PCI collision, or be already on the NRT of the serving cell, which results in the PCI confusion. Both issues are detailed below.

PCI Confusion

PCI confusion happens when the detected PCI is in the NRT of the serving cell. The serving cell assumes that the sensed cell is already in the NRT and does not request a check of the ECGI. The situation is made worse in the scenario where the UAV tries to handover to this concurrent cell because all the handover configuration will be carried out with the wrong cell and the UE could have its connection broken. The opposite can also happen: if a UAV adds a distant cell to the list and a GUE senses a closer cell with the same PCI, the closer one would not be added to the NRT, which would result in the handover configuration being sent to the far away cell. It may even result in the concurrent cell being added to a block-list, as many attempts to handover to this cell would fail. A neighbour should be block-listed if there are repeated attempts of unnecessary connections, and once block-listed, the cell is not an option for handover anymore.

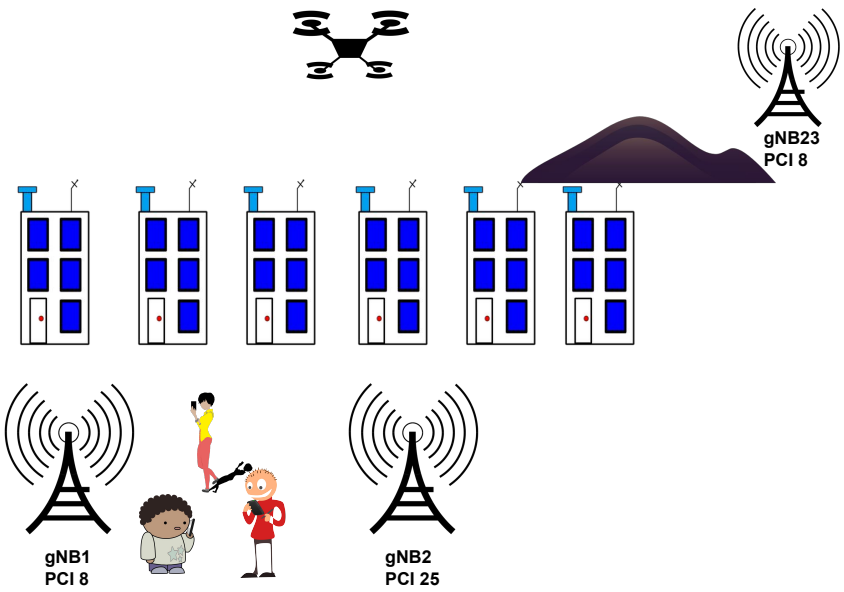


FIGURE 3.1: PCI confusion/collision challenge.

As an example, assume that in Figure 3.1 the UAV is connected to the *gNB2*. In the *gNB2* NRT, the *gNB1* is a neighbour, and its PCI is saved in the table corresponding to *gNB1*. Once the UAV flies and senses a strong signal from *gNB23*, it detects its PCI. As the PCI of *gNB23* is the same as that of *gNB1*, the serving gNB, *gNB2*, decides that the signal sensed by the UAV is from *gNB1* and does not ask the UAV to verify the ECGI. If the UAV tries to handover to *gNB23*, all the configuration for handover will be sent to *gNB1*, and the network might not be able to detect that there is a problem before the UAV disconnects.

PCI Collision

PCI collision happens when two cells that cover the same area are allocated with the same PCI. In this situation, the UE connected to one of them will not sense for another cell with the same PCI, which can result in the UE not being connected to the best serving cell. For example, consider that the UAV is going in the direction of the hill and is connected to *gNB1*. Even if *gNB23* has a strong signal and is the only gNB available in that direction, the UAV will not consider it as as option and will disconnect before trying to connect to *gNB23*.

A possible consequence of PCI confusion and collision is that the network has to be updated with more appropriate PCIs once these issues happen. To update the PCI of a cell, the gNB needs to be restarted, which can take more than one hour.

To solve the PCI distribution issue, one possible solution would be for UAVs to have two radios, one for communication and one for measurements. Radio one (R1), would be used for communication, but its priority would be sensing. Radio two (R2), would be used for communication only. When the UAV needs to sense and make measurements, we propose that the UAV would always sense the ECGI directly to avoid PCI confusion/collision. During the measurements, R1 should stop any communication that could be using the radio. If used in unlicensed spectrum, R1 would be all the time sensing the spectrum in order to make spectrum characterisation. R2 would not stop its transmission and data reception at any time during the measurement reports. This method would ensure that UAV does not lose connection during the measurements. The drawback of this approach is that having two radios is more expensive and takes up additional space on the device.

Nevertheless, the use of two radios should be considered by vendors and regulators.

3.2.3 Handover Challenges

Once the PCI confusion and collision issues are resolved, additional challenges related to handover need to be addressed to ensure that UAV UEs do not overload the network and do not unnecessarily disconnect. We discuss those below.

Frequency of handovers for UAVs

Authors in [67] reported that UAVs perform, on average, five times more handovers when compared to a GUE. These values show that the mobility of a UAV tends to generate more signalling overhead in the network and that the parameters used to trigger event A3 need to be adjusted for UAVs.

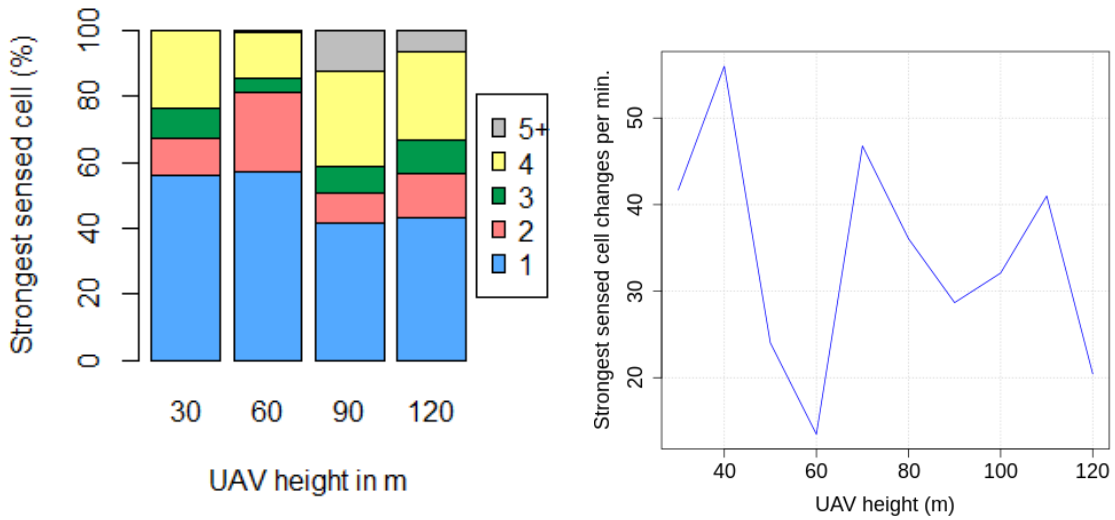
Connection interruption time

Authors in [68] show that sometimes the handover does not start for UAV users because the RSRP measured by the UAV from neighbouring cells does not have a minimum difference of an offset (usually 3 dB) between the serving cell and the possible handover target cell. As a result, the UAV UE does not send event A3, which is required to trigger the handover. A consequence of this is that UAVs will experience more frequent disconnection from the network than GUEs [68]. Once the UAV moves, it moves between side lobes and antennas nulls quickly, and there is no time to make a seamless handover, resulting in disconnection when the UAV enters the nulls of the antenna [9].

3.2.4 Experimental Validation

To support our claim that the connected UAV may experience a performance different of the expected when the network was planned, we made use of the real-world dataset SATORI DenseAir Smart Docklands, available in [69], with signal to noise power measurements made by a UAV-mounted handset. Figure 3.2 illustrates the analysis. The network is a two-tier cellular network in Dublin city centre that operates in the 3.6 GHz band.

Typically, for GUEs it is a fair assumption that the UE will be connected to the closest cell, a common assumption made by the research community [13], [70]. This analysis investigates how often the UAV sensed the strongest signal as coming from the geographically closest cell during its flight. Figure 3.2a illustrates the most potent sensed cell relative to its distance to the UAV, for four different altitudes, 30 m, 60 m, 90 m, and 120 m. At 30 m and 60 m, the UAV senses more than 50% of the time the strongest signal as coming from closest cell. The same does not happen at higher altitudes: when the UAV is at 90 m and 120 m, it senses the closest cell as the strongest for around 40% of the time; for almost 30% of the time, it senses the signal from the fourth closest cell as being the strongest one. The behaviour presented in the results clearly differs from the expected behaviour from a GUE. Figure 3.2a reinforces the idea that the coverage in the air needs to be considered for new BSs, as the UAVs can connect to much more distant BSs. It also highlights that the research community's assumptions that the UE will connect to its closest BS is no longer holds in the case of UAVs.



(A) Percentage of time the UAV was sensing the n-th closest cell as the strongest signal cell.

(B) Number of strongest cells changes sensed per minute during the path per altitude.

FIGURE 3.2: Graphical analysis on real-world dataset.

Using the real-world dataset, we carry out an additional analysis of the small-cell deployment, by looking at how often there is a change of the strongest cell when the UAV is flying through the network. Figure 3.2b illustrates how often the UAV experiences a change of strongest cell during its path. The collected values show that the strongest cell fluctuates dramatically across different heights. This is due mainly to the planned service area for the network being primarily at ground level. At other heights there are no dominant cells and hence several cells are received with similar signal levels. Further investigation is needed on how the handover performance can be optimised at these

heights.

3.2.5 Designing the Mobile Network to Incorporate Connect UAVs

In order to accelerate the use of UAVs in the mobile network without decreasing its performance, we propose solutions to minimise UAV's impact on the network. One possible approach to avoid lack of coverage, PCI confusion and collision, and handover challenges is to adapt planning tools to plan the coverage on the air.

Another solution that could be implemented to minimise the coverage and handover problems in the licensed and unlicensed spectrum is the use of a spectrum characterisation tool by the network with a UAV before allowing the full use of the network by UAVs, in order to analyse the coverage in the sky. The approach would be to analyse the transmissions on the spectrum, BS's coverage in licensed and unlicensed spectrum and, depending on operator's strategy, improve the coverage in a specific region, or recommend UAVs to not fly in the areas with poor coverage, or even, recommend UAVs in specific regions to use priority one RAT over another (for example, use LTE but not 5G in a specific region).

Analysis of the spectrum is such an important task to solve the coverage and PCIs challenges that we also propose that the task should be more complete and divided by specific heights. As the height is the main difference between the UE and UAV, and only with increase of height that a UAV would sense PCIs from further BS due to the LoS, we suggest that the network could implement a maximum recommended height for UAVs in order to not deteriorate the mobile network to others UEs. So far, regulatory national

bodies have defined the allowed height range at which that the UAV can fly, although we propose a study where the mobile networks also could have localised maximum heights (under the maximum height the national bodies allow). This height would be decided by being the maximum height a UAV can fly in that city without encountering PCI confusion and collision.

These solutions can mitigate the challenges a connected UAV could bring to the network and increase UAV's QoS, however, they would take long time to be implemented in all the network for the different mobile generations. That is the reason in this thesis we focus on the mitigation of these challenges through intelligent design of the UAV itself rather than the network.

3.2.6 Section Summary

In this section we presented the mobility related challenges a connected UAV can bring to the network. The challenges presented were the coverage planning, PCI confusion and collision, high frequency of handovers for UAVs and higher interruption time to UAVs. These challenges are recognised, as an issue based on other research investigations backed up by our own UAV connectivity dataset analysis. With these works, we have shown that the UAV can connect to further BSs, differently of GUE. It also shows that the coverage expected for UAVs are not the same as planned for GUE, and that the UAV can experience degraded performance.

We propose directions to these challenges to be solved on the network side, where the network operator would need to make changes to its planning stage, such as simulation

tools, after site implementation drive tests, PCI allocation. However, a change in all the network side might take a long time. This is because it would need to be implemented on all new sites, and it would be necessary to make a study on the sites that are already working. That is the reason we investigate the solution on the UAV in the next section.

3.3 Designing Connected UAVs

Usually, UEs of the network are passive when considering its QoS and mobility management, and do not take decisions in order to improve it. The GUEs role in the mobility is to take measurement report and send them to the BS, and so far, the responsibility of providing the required QoS is only of the network. We propose that instead, the UAV can adapt its path considering the improvement of its QoS. The connected UAVs and the unlicensed BSs can make use of a more detailed sensing of the spectrum in order to take more informed decisions related to the UAV's mobility management. This sensing information could be sent to the BS and the network could take the handover decisions, or could be used by the UAV to decide about its own mobility. These two approaches are proposed on the UAV as the network adjustments could take long time to be implemented. Also, as we still do not have connected UAVs available in the market, it is still a open discussion of what it will be required and which function it should provide. In this section, we investigate the state of the art for the approaches designed to increase UAV's QoS in the deployed mobile network.

3.3.1 UAV Handover Optimisation

Researchers are investigating the UAV-BS association [13], [39], [71]–[73] using ML with the goal of integrating UAV to the mobile network as a UE. One of the first studies to improve the link UAV-BS was presented in [74], where authors propose a ML model to adjust antenna's tilt on the BSs with the purpose to provide the best possible link to UAVs and GUEs. They use a model-free RL on the BS, that uses Signal-to-Interference-plus-Noise Ratio (SINR) of both UAVs and GUEs as input to it model. Authors show that with this adjust, the number of handovers for UAVs decreases without a great change in GUE's QoS.

In [72], authors propose a RL solution for selection of BS when the UAV trajectory is known. They first define geographical areas where the UAV should analyse the need for a handover, if it is necessary they apply the DQN model to decide which of the BS the UAV should handover. The authors conclude that their model is effective in decreasing the number of handovers while maintaining a relatively good RSRP. In [71], authors present a RL approach to optimise the cell selection and the resource allocation of connected UAVs. In the simulations, authors discovered that an increase in UAV velocity decreases the number of handovers; however, UAV height increases the number of handovers necessary for the UAV to maintain its QoS.

In [13], the work investigates the UAV channel estimation with a supervised learning approach. This work assumes that the UAV is not moving and has two antennas, one omnidirectional for sensing, and a directional antenna for UAV-BS link connection. With

the knowledge of the BS locations and the sensed channel information from the omnidirectional antenna, the model deduces which BS will provide the best channel conditions on the directional antenna. An extension of this work [39], assumes that the UAV moves through a city. In this case, the UAV can perform handovers, and the number of handovers are optimised. In this scenario, this work assumes interference from other antennas and building blockage. The proposed solution doubles the total throughput when compared to benchmarks.

Handover optimisation for connected UAVs has been studied as a way to improve UAV's QoS. However, most of the solutions propose a change in all the network infrastructure parameters [74], or that the UAV will be able to decide about its handover in a decentralised manner. Changing the configuration of all the antennas in the network is a considerable change because the network was already optimised to work with these parameters. Having decentralised mobility management is a fundamental change in how the mobile network works. It would be necessary to investigate a secure way for this to happen. Both solutions are physically possible, although unlikely to occur due to the operator's risk to its ground coverage and its security. A less compromising way to solve mobility management challenges would be an approach to the network mobility parameters. For example, studies for fast train's handover [75] are in the direction of adapting the parameters to trigger the events depending on the train's velocity and sensing. This approach is a promising investigation area for connected UAV.

3.3.2 UAV Movement Optimisation

One of the biggest challenges for the UAVs is its 3D path optimisation. The problem can be divided in UAV horizontal movement optimisation and vertical movement optimisation. This optimisation can be in order to achieve different purposes, as to have the shortest path while ensuring minimum connection quality [17], for example; or in [18], where the authors consider the battery life of the UAV when planning the UAV path. In this section, we first present UAV horizontal trajectory optimisation followed by vertical optimisation in scenarios where UAVs are used as BSs.

UAV trajectory optimisation

The works on trajectory optimisation focus on 2D optimisation and rarely mention the optimal height. In this section, we introduce works that optimise the trajectory considering the UAV-BS access link.

In [19], the authors propose optimising the horizontal path of a cellular-connected UAV that flies from an initial to a final location, while maintaining reliable communication with the underlying mobile network. This approach proposes that the UAV flies at the fixed minimum height allowed by the regulatory entities. In this study, the authors do not consider the interference from BSs to which the UAV is not connected and blockage from the buildings blocking the link from UAV to BS. To accomplish the study objectives, a graph representation of the network is proposed, with 3 solutions: first, a graph where each node is a BS; second, a graph where the nodes are the handover points between the BSs; and third, where the handover points are the optimal point in an intersection

area. Dijkstra algorithm is used to find the route of the UAV and show it is close to the optimal solution. Height optimisation was not considered, and the authors conclude that introducing a height variable to the problem is not a trivial task and that their proposed horizontal trajectory solution is not the most appropriate one for 3D movement. They conclude that it would be unfeasible to represent all the possible heights a UAV could have at all the nodes, as each of them should be a new node increasing the system's complexity.

The work in [14] creates an optimised path with the objective of maintaining a uninterrupted connection to the BSs. This work only considers the uplink from the UAV to the BS network. This work also highlights the importance of the altitude of the UAV and calculates the upper and lower bounds for the height at which the UAV should fly to satisfy the minimum rate requirements of the uplink, considering the known BSs locations. Authors calculate a range of heights at which the UAV should fly to provide a minimum achievable rate. In addition, building blockage on the link UAV - BS is not considered. The solution is based on the game theory multi-agent approach, where each UAV is the player. They propose a echo state network (ESN) [76], which is a recurrent NN where the connectivity and weights of the hidden layers are fixed and randomly assigned. With this approach, each UAV decides its next horizontal location. The authors conclude that the altitude is vital to minimise the transmission delay of the UAV and that it should be a function of the ground network density, network parameters as the transmission power, ground network data requirements and the UAV's action. The exact height of the UAV is not calculated as it would increase the complexity of the algorithm exponentially.

Height optimisation for UAV BSs

The height planning of a connected UAV is a new field, however several works have studied the height placement of UAVs acting as BSs. The techniques used to optimise the heights at which a UAV acting as a BS should fly can also overlap with our problem of interest as it also consider the radio link between a UAV and a element that is located at lower heights.

In many examples of the prior art, works on UAV wireless connectivity, either for UAV as network end-user or UAV as BS, did not consider the effect of interference conditions. The quality of the link between UAV and a BS can suffer from interference coming from other BSs, from objects or buildings intercepting the directional connection between them (shadow zone), or even the natural fading on the propagation. The work in [77] assumes UAV as BS and provides coverage to GUEs. The authors propose a sigmoid model to investigate the probability of LoS channel in the UAV - GUE link as a function of the vertical angle between them. In the paper, a UAV as BS with an omnidirectional antenna flies over an urban area. They conclude that a bigger angle decreases the probability of a building block the link. They also add that there exists an optimal height for the UAV BS, which increases the coverage area.

In [78] and [79], authors applied stochastic geometry to model the coverage probability of a UAV-BS network in a fading-free and Nakagami-m fading channel. The authors fix the number of UAVs operating in an area at a certain height above the ground and demonstrate that with an increase in height, the coverage probability decreases. Also, in [79] authors demonstrate that bigger values of fading parameter reduce the variance of

the SINR for the GUE.

In [80], authors propose the approach to calculate the 3D position of UAV as a BS, by applying the interior point optimiser of bisection search. Their main objective is to maximise the coverage area by a UAV cell without providing to a GUE a QoS below a threshold. They consider building blockage and Non-Line-of-Sight (NLoS) between the UAV as BS and its users. The covered area changes depending on UAV's height, and for the lower density of GUEs. The coverage is larger when compared to higher density, showing the worst coverage for urban scenarios.

In [81], authors find the optimal position for a network of UAV as BS in order to minimise the number of BSs required to provide the needed QoS for their users. The study considers the blockage generated by buildings in an urban area and NLoS occurrences between the UAV and its users. The proposed solution used an heuristic algorithm based on the number of BS that can serve the GUE, coverage and capacity requirements. The number of users on the network was essential to define the height and number of UAVs serving as BSs. The authors concluded that with their solution it is possible to decrease the amount of UAV as BS and provide the same quality on data rate.

Similarly, the work in [82] proposes a 3-step solution for horizontal and vertical optimisation for UAV-BSs with different ML algorithms for each step. The bounds of the UAV height are the UAV maximum transmission power for its greatest height, and the minimum required distance between the UAV and the users defined by the regulatory entities, for the minimum height. In the first instance, the paper considers a static problem, where the users of the network do not move. As a first step, it partitions the area into cells for

the UAV-BSs to cover, by applying K-means (GAK-means) algorithm. Next, it uses a Q-learning algorithm, where each UAV is an agent and has to decide its position by learning from its mistakes. As the final step, they consider a scenario where users move between BSs and the network have to adapt to these movements. The authors apply DNN, as it enables each UAV to gradually learn the dynamic movements of the users. They conclude that the proposed solution outperforms the K-means algorithm and IGK algorithm with low complexity.

The UAV-BS scenario defers from the connected UAV problem because the connect UAV moves through the city and do not divide it into cells, so the use of K-mean for clustering, for example, is not applicable. However, the use of reinforcement learning to adapt its height depending on the cellular network radio technology and the regulatory entities definitions is valuable insight. To apply DNN into the connected UAV scenario, one needs to investigate what is relevant to a UAV as UE, which are the information a UE has from its connection, how the UAV can interact with the environment, and design a model that can learn all these characteristics and be effective through different topologies.

Table 3.1 provides a summary of the state of the art in connected UAV movement optimisation.

Paper	Type of UAV	Optimise	Method	Validation	Available code
[17]	Connected UAV	Distance to BS	Graph	Simulation	No
[70]	Connected UAV	Coverage Prediction	Cauchy's inequality	Simulation	No
[19]	Connected UAV	Horizontal Optimisation	Graph	Simulation	No
[14]	Connected UAV	Horizontal Optimisation	Deep RL	Simulation	No
[80]	UAV BS	3D position	Bisection search	Simulation	No
[81]	UAV BS	3D position	Particle swarm optimisation	Simulation	No
[82]	UAV BS	3D position	DQN	Simulation	No
This thesis	Connected UAV	Height Optimisation	DQN	Experiment/Simulation	Yes

TABLE 3.1: UAV movement optimisation works.

While some existing work has looked into optimising the height of UAV BSs, there is a significant lack of work looking at UAVs when they are the end users. Our contribution C2, which will be described in Chapters 4 and 5, is closely related to the presented state of the art but optimise the height of the connected UAV.

In this thesis, we propose optimising the altitude of the UAV, separating it from the problem of a horizontal trajectory decision, to evaluate only the height of the UAV. We separate it from the horizontal optimisation trajectory as in some applications, such as organ delivery, the horizontal path will be defined by the application, and only the altitude will have the freedom to be adapted. As mentioned in Chapter 1, the height optimisation is proposed in order to answer RQ2, supporting the UAV to improve its own QoS while avoiding areas with poor signal.

The proposed solution applies RL to decide, based on environmental measurements, if the UAV needs to move above, below, or stay at the same height in order to experience the best QoS possible from the cellular network in the long run and avoiding areas with poor coverage. To the best of our knowledge, this is the first time one dynamically optimise the height of a cellular-connected UAV, focusing on improving its connectivity.

Our proposed UAV height optimisation could be used with the following purposes:

- Be used by the UAV to take informed decisions of its next height in order to improve its QoS.
- Be used to automatically avoid areas with poor signal in the sky.

- Be used by different UAV applications as it does not require modification of the horizontal path.

3.3.3 Spectrum Characterisation

The communication of the UAV and its pilots is made trhough the unlicensed spectrum for now, although it is expected to change in the future. As the use of this spectrum is allowed to any technology that can share the spectrum, the UAV coexists with other UEs from different technologies (WiFi, Bluetooth, and Zigbee, for example). The tendency with 5G UEs operating in the unlicensed spectrum is that the use of this spectrum will increase with the cellular mobile network utilisation [27]. In order to propose the first step to the UAV connection to the mobile network, we investigate the use of the unlicensed spectrum for the cellular mobile networks. Expanding the spectrum possibilities will increase the chances of the UAV having coverage and better QoS.

The first step in allowing an LTE or 5G BSs to share the spectrum in an efficient manner, is to understand what is happening on the channel that the BS wants to transmit on, such as which transmissions are happening and how busy is the spectrum. For this, several research efforts have focused on spectrum characterisation [83]–[85]. In the case of LTE in unlicensed spectrum (LTE-U) researchers relied on contextual information about the spectrum usage to carry out spectrum characterisation, and enable spectrum usage of the shared spectrum [23], [86]. With more detailed information about the spectrum behaviour, it is possible to make more sophisticated decisions about how to use the spectrum.

Modulation Classification

Some of the early work on spectrum monitoring has relied on techniques such as cyclostationary feature detection and energy detection [87], [88]. The focus tended to be on detecting the presence of a signal in the band of interest, rather than characterising the signals being detected. More recently, there has been renewed interest in modulation classification, driven by spectrum sharing in military bands, and, in the commercial arena, by the possibility of operating LTE and 5G in unlicensed bands, sharing the spectrum with other RATs such as WiFi and radar communications. Current military and commercial spectrum sharing can benefit from more sophisticated awareness of what other transmissions are present in the band, and what the characteristics of those transmissions are, than the earlier spectrum monitoring solutions were able to provide. This has motivated a number of ML-based solutions for modulation classification.

In [89] and [90], authors propose the use of Support Vector Machine (SVM) for modulation classification, and in [91] they propose the use of Genetic Programming with K-Nearest Neighbours (GP-KNN) to achieve the same objective. However, both SVM and GP-KNN techniques are susceptible to frequency and phase offsets, which can compromise the signal classification accuracy under multi-path, fading, or other real-world Radio Frequency (RF) impairments. Following works have focused on how to make ML-based models more robust to SINR and Received Signal Strength Indicator (RSSI) variations, capturing real-world RF impairments and generating more reliable RF signal classifiers. For example, in [92], the authors generate their dataset through over-the-air transmissions

using Universal Software Radio Peripherals (USRPs), evaluating the accuracy of their classification model under different SINRs. The works of [93] and [90] also consider different SINRs when applying ML algorithms for modulation classification. In [93], the authors investigate the classification problem using dictionary learning. In [94], authors characterise the performance achieved by a CNN model when identifying distinct modulations for different SINR levels using spectrograms.

The work of [95] applies computer vision methods to modulation transmission detection. They perform object detection on spectrograms using their own generated dataset. Their model detects the transmitted frames, but it does not classify them. They used 20,000 spectrograms representing 3 ms to train their model, which illustrates the extensive amount of samples to train a model in computer vision.

The coexistence between different RATs in shared spectrum requires more information about the surrounding wireless devices than simply the knowledge of their modulation schemes, e.g., QPSK or QAM. For example, different RATs may employ the same type of modulation and yet use different medium access schemes. In [96], the authors propose a long short term memory (LSTM) model for modulation classification in large distributed networks of low-cost sensor nodes. As the input of their model they use two lists, one with the time domain amplitude and another one with the time domain phase of the signal. They made different analyses of the modulation classification per SINR and also about the technology classification. They conclude that a modulation model classifier is not always effective in classifying RATs, as different RATs might share the same modulation schemes.

The works summarised above provide solutions in the field of signal classification with, predominantly, high accuracy in what they propose to do. However, most of them are focused on modulation classification and not on RAT classification. These works also do not exploit a scenario with varying degrees of interference and overlapping transmissions among the classified signals, which significantly increases the difficulty in correctly classifying these transmissions.

RAT classification

In an environment where the spectrum is shared between multiple RATs and multiple access points belonging to the same RAT, e.g. in the Industrial, Scientific and Medical (ISM) band, transmissions that occur in the same frequency channel can happen with full or partial overlap. Recognising these cases and localising them can lead to better interference and coexistence management mechanisms; for example, a BS could define its bandwidth in order to avoid a part of the spectrum that is being heavily used.

Authors in [94] investigate RAT classification. They show that, in the context of radar signal detection for low SINR scenarios, a model that considers both amplitude and phase leads to better results than one just employing amplitude. In this scenario, the focus is on distinguishing between radar, LTE, and WiFi transmissions. In [97], authors analyse different ML techniques for wireless technology classification with two different datasets, and verify the ability of the model to generalise to unforeseen scenarios. These solutions are effective in performing RAT classification but do not provide a more detailed characterisation of the spectrum.

Table 3.2 summarises the state of the art in spectrum sensing and position our work in the literature.

Paper	Classification	ML	Validation	Different SINRs	Feature extraction	Available code
[90]	Modulation	SVM	Simulation	Yes	No	No
[92]	Modulation	Experimental/Simulation	Yes	Yes	No	Yes
[93]	Modulation	Dictionary Learning	Simulation	Yes	No	No
[94]	RAT	DNN	Experimental	No	No	No
[95]	Detection of Spectrum events	Experimental	Yes	No	No	No
[96]	Modulation	Recurrent NN	Simulation	Yes	No	No
[97]	RAT	SVM/NN/CNN	Real-world data	Yes	No	No
[98]	Received Power	SVM	Simulation	Yes	No	No
[99]	RAT	DNN	Experimental	No	No	No
This thesis	RAT	Object Detection	Real-world data/Experimental	Yes	Yes	Yes

TABLE 3.2: Spectrum sensing works.

Our contribution C3, which will be described in Chapter 6, is closely related to the presented state of the art in spectrum sensing, although it does not only classify RATs but provide extra information about the spectrum.

The proposed solution [100], applies object detection on spectrograms, which allow us to classify and extract key features of the sensed transmissions. The advantage of using object detection to the RAT characterisation problem is that this technique identifies the object independently of its location in the image and if there is more than one transmission at the time. This allows the detection and classification of transmissions with different bandwidths, different duty cycles, different times of transmissions and centre frequency. For this, we rely both on a dataset that we generated in the Iris Testbed [101] and on available datasets of real-world commercial transmissions.

To the best of our knowledge, this investigation is the first to detect and classify different RATs transmissions applying a object detection technique and to evaluate its performance under unfavourable noise and interference conditions. Our work is focused on RAT classification, exploring the ambiguity of different OFDM-based RATs, namely, LTE

and WiFi. We also believe that more information about the spectrum can guide a new user to make better choices to operate efficiently; therefore our solution also provides feature extraction functionality, which can be used to build efficient dynamic coexistence mechanisms in shared spectrum. Information as how long other transmissions are using the spectrum, their interval of transmissions, and their bandwidth would lead to an efficient dynamic spectrum coexistence. Table 3.2 provides a summary of the state of the art in spectrum characterisation.

We propose that the spectrum characterisation in the UAV context could be used with the following purposes:

- Be used by the UAV or the mobile network to take informative decisions on the UAV mobility management, avoiding ping-pong effects and disconnection time.
- Be used in order to make an efficient use of the spectrum and analyse the best way to attend the UE's QoS requirements.
- Be used for coverage analyses on the sky.
- Be used by governmental authorities to detect Jamming attacks.

The use of our technique is not limited to UAVs, and spectrum characterisation can be useful in for any unlicensed UE.

3.3.4 Section Summary

This section presented the state of the art on UAV approaches to improve its QoS. We start presenting works in UAV handover optimisation, in which authors propose adaptations to the mobility management for connected UAVs. Most works in this field apply ML because often the resulting optimisation problems are intractable for conventional optimisation schemes.

We also presented the UAV movement optimisation, which was investigated as an optimisation problem constrained by different factors, such as UAV's batteries, path minimisation, and connection to the mobile network. Although the optimisation considering the UAV's connection is newer and focused only on the horizontal path, even though some mention the UAV height optimisation, this optimisation is not often investigated in the literature. Therefore, in this thesis we investigate the height optimisation issue separated from the horizontal path optimisation. As a separate solution, even if other feature defines the horizontal path that is not related to the UAV's connection, the height optimisation can still improve UAV's QoS and help it to avoid poor coverage areas.

Finally, we presented the spectrum sensing state of the art, dividing them into modulation and RAT classification. The firsts approaches focused on only determining if the spectrum was occupied or not. Then, researchers started to classify the modulation classification using ML, and finally, the RAT classification using ML. These works are usually assertive on what they propose to do. However, we envision that the next step in spectrum sensing is the RAT classification and more informative details about the spectrum. This is due to the fact that mobile cellular networks will be using this spectrum with all

the other UEs and to maintain efficient use of the spectrum, the network will need more information.

3.4 Conclusion

In the chapter, we presented the challenges a UAV may bring to network operators, and the state of the art in the areas where the QoS of the UAV can be improved in the mobile network.

In this chapter we presented RQ1, that is "Which are the main challenges a connected UAV may encounter if deployed in a typical modern-day cellular network?", providing C1 punctuating the challenges during the chapter. The challenges presented highlight the need for adaptations on the network or on the connected UAV side for the introduction of connected UAVs, otherwise the network may experience QoS issues for both air as well as GUE. We divided the challenges into coverage, PCI and handover challenges. To support our claims regarding the challenges, we analysed a real-world dataset which contained measurements from a UAV user connecting to a small cell network in an urban environment. Then, we proposed directions to mitigate the effect of these challenges that should be implemented by the mobile operator.

As the modifications on the mobile operator side might take long time to be implemented, we introduce the state of the art of the possible ways a UAV can improve its QoS. The research areas are then divided in UAV handover optimisation, UAV movement optimisation and spectrum characterisation. In the rest of this thesis we will provide novel

and well-performing solutions to address the issues of UAV movement optimisation and spectrum characterisation in order to answer RQ2 and RQ3.

First, to answer the RQ2 "How to adapt the UAV height in order to increase UAV's QoS?" we present the UAV dynamic height adaption in order to improve UAV's QoS and avoid areas with poor or no coverage. Then, we propose a spectrum characterisation approach to answer RQ3 "How to better characterise the unlicensed spectrum in order to enable more efficient spectrum usage?".

Chapter 4

Adaptive Height Optimisation for Connected UAVs - Investigation Based on Real-World Measurements

This chapter together with the following Chapter 5 aims to answer RQ2 "How to adapt the UAV height in order to increase UAV's QoS?", adapting UAV's height in order to provide a better throughput. The proposed dynamic height optimisation is done applying RL method that has as input to its model data that is commonly find on UEs.

The first part of our contribution, covered in this chapter, focus on developing and evaluating the proposed RL approach for a real-world collected dataset. Then, in Chapter 5, we extend this evaluation in a simulation environment, where we can vary the density of BSs and buildings in order to generalise the proposed solution. The data used to evaluate the proposed solution in this chapter was collected with a smartphone attached to a UAV that flew in two different areas of Dublin city centre.

This chapter comprises the following sections: we introduce the problem in Section 4.1. Then we present the problem statement in Section 4.2, explaining the scenario we are assuming for this investigation. In Section 4.3, we explain in detail how the data was collected in this chapter. Then, in Section 4.4, we detail the proposed approach. In Section 4.5, we presented the pre-processing we needed to do to the data in order to use it for the height adaptation approach. Section 4.6 investigates the performance of the proposed approach compared with some benchmarks. We then, conclude the chapter in 4.7.

4.1 Introduction

UAVs can leverage 5G connectivity to perform security surveillance, search and rescue operations, building inspections, and more. However, providing reliable connectivity to such UAVs is still an open problem, as they present a paradigm shift when compared to their ground counterparts such as smartphones. According to the specifications (release 14 of 3GPP [102]), a UAV needs to maintain continuous connectivity with the mobile network at speeds up to 300km/h.

Previous work, such as [8], [9], investigates the feasibility of using existing network infrastructure to provide reliable wireless connectivity for UAVs. These studies conclude that currently deployed networks would need to adapt some of their design configurations, such as increasing BS heights [10] or changing the tilt of the antennas [11] so as to enable connectivity for UAVs. Redesigning the terrestrial network infrastructure may be unfeasible, and an adaptable solution on the UAV side may be necessary to accelerate the UAV integration into the network.

Due to the height at which UAVs fly, there are often no obstacles and therefore no blockage between the UAVs and their serving BS, which is good for the UAV connection. However, at high altitudes there is an increased probability of LoS to ground BSs in general, not only the serving one, which results in high levels of interference at the UAVs. The work in [10] states that the optimal height at which the UAV can fly to maintain reliable communication depends on the BS density and UAV's height. Similarly, the authors in [70] show that the vertical movements of the UAV affect its coverage probability.

However, no work addresses the dynamic height optimisation of UAVs acting as end-users of the cellular network. Motivated by the above, this chapter aims to answer the RQ2, "How to adapt the UAV height in order to increase UAV's QoS", by proposing an RL approach for dynamic optimisation of the height of a UAV connected to the cellular network once it moves in realistic city environments, consisting of buildings of various heights, as illustrated in Figure 4.1. To the best of our knowledge, this is the first work to optimise connectivity of a cellular-connected UAV by dynamically adapting the height at which it is flying.

4.2 Problem Statement

We consider an urban scenario where a UAV flies while connected to the cellular network. The UAV's initial and final positions are denoted as (x_1, y_1, z_1) and (x_f, y_f, z_f), with f representing the total number of discrete steps in the experiment. x and y denote coordinates on the horizontal plane, while z denotes height above ground. At each step, the UAV moves in the x coordinate in direction to its final destination. In this work, as the focus is

FIGURE 4.1: Side view of a UAV connected to the mobile network adjusting its height to maximise its throughput.

on UAV height optimisation and many approaches for optimising 2D trajectories already exist, we assume a simple horizontal path (ie. either a straight line towards the destination, or allowing minor deviations to move closer to serving BS). Note that simplification of the path does not affect the applicability of our proposed approach, as due to its design, it can be integrated with more complex horizontal path algorithms (which we reviewed in Section 3.3.2 and are out of the scope of this thesis). In other words, the only coordinate that can be optimised is z. We assume that the maximum height change at each time step is d, so $|z_t - z_{t-1}| \leq d$.

Usually, UAVs are allowed to fly in a height range defined by safety regulation, with the minimum allowed height denoted as Z_{min}, and the maximum allowed height as Z_{max}. We assume that the UAV starts at Z_{min}. Figure 4.1 illustrates the possible path of the UAV, where d is the maximum distance the UAV can move up or down in each step. d is a

representation of how much a UAV can move realistically up or down and horizontally in a time-step.

Our main objective is to optimise z coordinate at each step, in order to improve the QoS experienced by the UAV. The metric used to represent the QoS is the downlink throughput. The problem assumes the downlink throughput as we use real-world data, and in the data collection, this was the parameter measured. We understand that for a range of applications, the most used link would be the uplink. Therefore, the solution could be modified to work with the uplink throughput without modifications to the actual proposed method. Throughput is first calculated as symbols per second, where these symbols are defined by their modulation [103]. Depending on how many bits a symbol can carry, throughput is converted into bits per second (bps). Thus, throughput can be defined as $T = sM$, where s is the number of transmitted symbols per second, and M is the number of bits per symbol. The modulation is chosen based on the channel conditions.

We formulate the optimisation problem as follows:

$$\max_{(z_1,\ldots,z_{f-1},z_f)} \quad \sum_{t=1}^{f} T(t); \tag{4.1a}$$

$$\text{s.t.} \quad z_t > Z_{min} \quad \forall \quad t \tag{4.1b}$$

$$z_t < Z_{max} \quad \forall \quad t \tag{4.1c}$$

$$|z_t - z_{t-1}| \leq d \quad \forall \quad t \tag{4.1d}$$

We assume that the UAV will have access to the SINR measurements from its connection, the downlink throughput it is achieving at that SINR, and its height at all steps. SINR and throughput data is easily obtained by the UAV from its cellular connection, while the height information is obtained via other UAV sensors located on the UAV, as already present in UAVs as the DJI-mini 2 [3] presented in Chapter 1.

4.3 Data Collection

To evaluate the proposed height adaptation solution, in the first instance we use the real-world measurements obtained by a UAV connected to a two-tier cellular network in two different areas of Dublin city's Smart Docklands, which includes massive MIMO macro cells and MIMO small cells experiment. The dataset was gathered by DenseAir, and is publicity available under request, and a sample of the data is available in https://github.com/galkinb/De
I was not involved on the data collection directly, my role on the paper was focused on the analyses. Bellow, we recap the details of the experiment relevant for our evaluation, while full details of details of measurements are presented in [104].

4.3.1 Building distribution

The environment in which the radio waves interact usually impacts on their propagation. During the data collection flights, the UAV flew above the water, which can absorb some signals while reflecting others, causing multi-path effects and destructive interference. The experimental cellular network testbed, in which the measurements were conducted, is shown in Figure 4.2. It was situated around a river, with large open areas suitable for

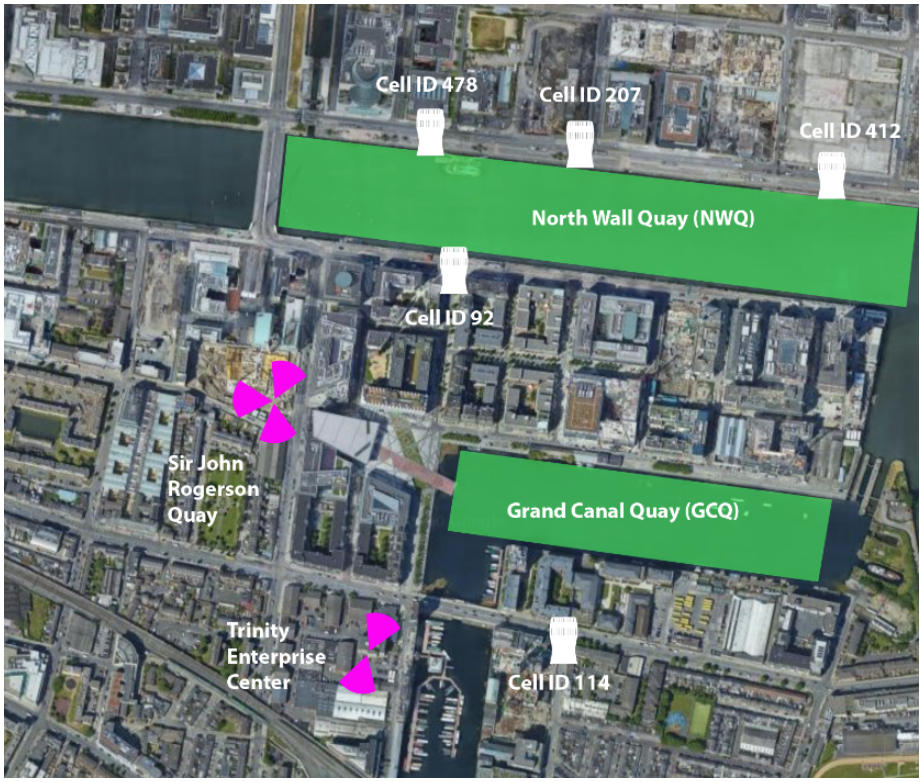


FIGURE 4.2: Testbed area from the top. Macro cells are labelled in purple, and small cells are denoted with white icons—the measurement area where the UAV flew is marked in green.

Time	Latitude	Longitude	SINR Carrier 1 (dB)	Serving Cell Identity	Throughput (kbps)
13:20:41.000	53.34342193604	-6.23032475884304	9.0	60	46872.35
13:20:43.000	53.34342193604	-6.23032503210378	8.6	60	46860.85
13:20:45.000	53.34342193604	-6.23032507764724	8.5	60	46870.39
13:20:47.000	53.34342193604	-6.2303251231907	8.1	60	46863.68
13:20:49.000	53.34342193604	-6.23032516873416	7.5	60	46846.63

TABLE 4.1: Example of collected data at 30 m in GCQ area.

UAV deployment. The buildings in the area have heights between 20 and 80 meters. The testbed consists of two macro cells deployed on building rooftops and a network of small cells deployed on lamp posts and traffic lights along the river at 6.5 meters above the ground.

4.3.2 UAV and BSs

The macro cell labelled Trinity Enterprise Center in Figure 4.2 is the macro cell that provided connection to the UAV in this area. This macro cell is a ZTE model ZXSDRB8300 with a 64 element antenna array Massive MIMO system, positioned 29 meters above the ground. The lamp post small cells are AirSpeed model1250 with four antenna elements that apply 2x2 MIMO. Both the macro cell and the small cells operate on the B42 channel of the 3.6 GHz, 5G frequency band. The maximum transmit power of the macro cells and small cells are 49.9 dBm and 25 dBm, respectively. The two-tier network has a hierarchical cell structure [105], where the small cell tier has a higher connection priority than the macro cell tier. It means that a UE prioritised connecting to the small cells even when it detects a stronger signal from a macro cell. In this experiment, the handset which was connected to the UAV had the operator DenseAir [106] SIM card.

The measurement flight consisted of flying a handset which operated on Band 42 (3.6 GHz) attached to a UAV through the urban area where the testbed is active at different heights above ground, while taking measurements. The UAV used was a DJI Matrice M300 carrying a Google Pixel 3 handset. Connectivity data was collected in two environments: Grand Canal Quay (GCQ) and North Wall Quay (NWQ), as illustrated in Figure 4.2. The UAV flew downloading an adaptive video from a commercial internet-based server, rather than seeking to saturate the wireless link.The UAV flew at a fixed height back-and-forth in the designated areas. This flight pattern was repeated at 10 meter increments for all heights between 30 and 120 meters (the legal flight ceiling in Dublin).

Table 4.1 shows a sample of the subset of the collected data in our evaluation. The first column represents the time the data was collected, followed by its GPS coordinates, the SINR observed for the serving cell, the serving cell identity and the throughput at that moment. Once the UAV connects to another cell, the cell identity value changes.

Table 4.2 summarises the main characteristics of the experimental environment for NWQ and GCQ. The Table shows: the height of the BS antennas; the velocity the UAV was flying; the total distance the UAV passed in each height; the quantity of measurements reports in each area, denoted as steps; the size of each step in meters; the variation of heights; the building height variation; and the distance d in each scenario. The measurements were reported every 2 seconds most of the time. We also observe that the flight in NWQ resulted in fewer measurements despite being the one where the UAV flies for a longer distance. While a UE is performing handover, it does not sense the spectrum; consequently, it does not report any measurement. In the small cell area, the UAV was

Variable	NWQ Value	GCQ Value
BS height	$6.5m$	$29m$
Speed	4.2 m/s	2.6 m/s
UAV travel distance	1160 m	890 m
Steps	161	171
Horizontal step size	7.2m	5.2
Allowed UAV height range	$[30-120]m$	$[20-120]m$
Building height variation	$[20-80]m$	$[20-80]m$
d	$10m$	$10m$

TABLE 4.2: Collection environments.

performing handovers, which resulted in fewer measurement reports when compared to the macro cell area, where the UAV did not perform measurement reports.

4.4 Proposed Solution

To solve the height optimisation problem for a specific position of the UAV given a particular topology of the BSs and buildings, one could apply stochastic geometry as in [12]. The main issue with this approach is that to represent this problem via stochastic geometry, one has to know the statistical distribution of the features of the environment for each position that the UAV assumes during flight. This can be computationally expensive to run and the environmental statistics may not be accurate to what the UAV would find once it is flying in the real world.

To tackle this issue our solution is based on RL. In particular, we apply DQN as it does not require a predefined model of the environment, since it learns by interacting with the environment in an online manner. In our implementation, we evaluated the proposed

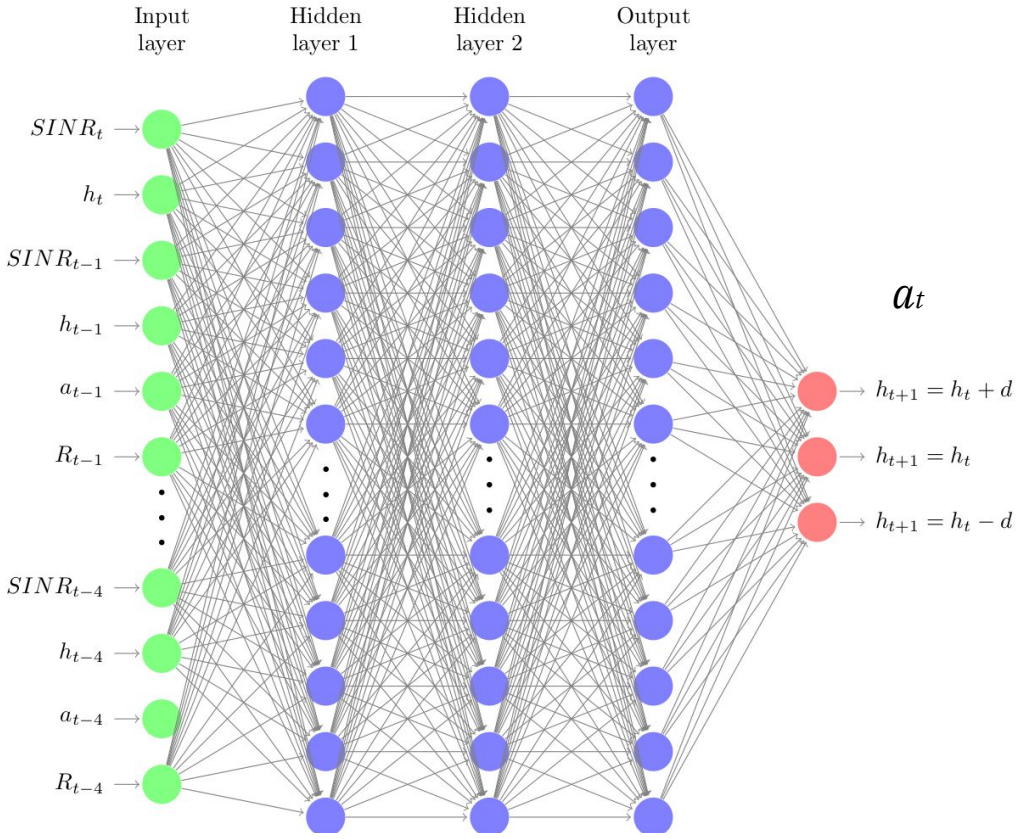


FIGURE 4.3: Graphical representation of the designed DNN.

model using a commercial off-the-shelf computer with Intel Core i7-6820 HK processor and GeForce GTX 1070 Mobile. The agent of our model is the UAV, as it is the one taking the action of changing the height. Bellow we define the other main components of our model:

State Space S

S is all the possible values of the state, and s is the individual single value of the state. We just considered in the state space values a normal UE would have from the network and measurements of sensors that a UAV should have to have a safe fly. Follow the components of S:

- Height z: which is obtained by UAV sensors and it is relevant for UAV's decision-making process, as in order to know whether to move next and stay within the hard limits.
- Received $SINR$: which is obtained by UE sensors to perform the measurement reports. This value impacts the UE QoS that is what we intent to maximise.
- 4 last z, $SINR$, a, r: which will be stored from the previous steps. In order to achieve better optimisation, we extended the state with the four previous z, $SINR$, action a and reward r, following the lines of the original DQN implementation [56], as well its implementation in UAV connectivity [39].

The agent has as input at each time step s_t, where t represents the time step the follow state. $s = \{SINR_t, z_t, SINR_{t-1}, z_{t-1}, a_{t-1}, r_{t-1}, SINR_{t-2}, z_{t-2}, a_{t-2}, r_{t-2}, SINR_{t-3}, z_{t-3}, a_{t-3}, r_{t-3}, SINR_{t-4}, z_{t-4}, a_{t-4}, r_{t-4}\}$.

Action Space A

The action is the adjustment of the UAV height. An action $a \in \{-d, 0, +d\}$ will be taken at the end of each time-step, where:

$$a = -d \Rightarrow z_{t+1} = z_t - d$$

$$a = 0 \Rightarrow z_{t+1} = z_t$$

$$a = d \Rightarrow z_{t+1} = z_t + d.$$

Reward R

As the primary goal of our approach is to improve the UAV throughput during flight, our reward at each time step r_t is defined as the throughput achieved after the action at point x_{t+1} at height z_{t+1}.

Our model has three hidden layers, with 200 neurons in each. Figure 4.3 illustrates the graphical representation of the proposed DNN.

In our solution, we use epsilon greedy approach, this is an strategy to balance exploration and exploitation. We selected the initial $\epsilon = 1$, where we select an action at random, and we decrease it at every step of the training process until it reaches 0.05, which in our experiment took 30 steps to reach.

4.4.1 RL algorithm for UAV height optimisation

The pseudo-code of the RL algorithm to optimise z_t is shown in Algorithm 1. Some parameters must be chosen and passed as input to the code to run the algorithm. They are the Z_{min} and Z_{max}, the minimum and maximum allowed height that the UAV could fly.

X and Y are the vectors with the horizontal coordinates the UAV should acquire during its movement. Where $X = x_1, x_2, .., x_f$ and $Y = y_1, y_2, .., y_f$, as the horizontal path is predefined. ϵ, ϵ_{Decay} and ϵ_{min} are needed to apply the ϵ-greedy approach. The input ϵ is the starting value for ϵ, and ϵ_{Decay} is a value that will multiply ϵ and reduce its value at each interaction until ϵ_{min}.

β and β_{min} are, respectively, the batch array and the minimum size of the batch needed to apply memory replay. While using memory replay, the number of $epochs$ to train the model and the $Discount$ factor to calculate the new Q value ($newQ$) is required. Finally, var is an integer that indicates how often the target model should be updated. The expected output of this algorithm is the UAV next height in the next step.

The first step of the proposed RL-based algorithm for UAV height optimisation is to initiate the DQN model and the target DQN model, lines 1 and 2, respectively. Then we initialise the UAV coordinates in line 3 and initialise variable t, which refers to the timestep the UAV is during the each step. The while statement in line 5 is the overall while loop that represents the full flight path of the UAV, and has as many steps as that set of X and Y.

Inside the step loop, it is needed to update t and collect the current value of $SINR$. Then, we update the state value in s_t. After that, we randomly select a number, $randomNum$, and compare its value to ϵ in line 9. This step is necessary to evaluate the comparison of the ϵ-greedy approach. In line 9, we also check t value to be at least 4, as we need the state information values from the last 4 states for the input of the model. If the condition

is satisfied, which means $randomNum > \epsilon$ and $t > 4$, we use the DQN model to predict the best action a_t. If the condition is not satisfied, we randomly choose the action a_t. Once the action a_t is defined, we execute it in line 15, ie modify the UAV height, moving the UAV up or down if it does not go above the permitted flight boundaries (Z_{min} and Z_{max}). Then, the UAV also moves based on the sets X and Y its horizontal coordinates to the next position in line 16. We obtain the reward which represents the quality of our selected action, and is later used to update the learning process. The reward r_t is equal to the measure throughout after executing the action, as shown in line 17. The ϵ decrease value is then performed in lines 18 to 20. The ϵ decrease is needed to decrease the amount of of random actions we perform once the model is being trained.

After decreasing the value of ϵ, we then save the new state s_{t+1} with the action a_t, reward r_t and the state s_t, in order to apply the replay memory later. Therefore, we need to discard the most old values from the previous 4, that refer to the $_{4t}$ timestep, so s_{t+1} has only its last 4 timesteps. Once we have the values of a_t, r_t, s_t and s_{t+1}, we can save them in the batch β, which will record the last values in order to train the model later using them.

To apply replay memory, the batch β needs to have a minimum size that is determined before the algorithm starts by β_{min}. In line 23, we check if this condition is satisfied. If it is not satisfied, we cannot yet apply replay memory. If it is satisfied, a batch sample of size βmin is taken from β and saved in the variable $tempBatch$, as illustrated in line 24. For each value in $tempBatch$ in the loop that starts in line 25, we keep in the variable $CurrentQValue$ the update of the Q value made by the actual DQN model in line 26.

Algorithm 1 RL-based algorithm for UAV height optimisation

Input: β; β_{min}; {//Batch parameters}
ϵ; ϵ_{Decay};ϵ_{min}; {//ϵ-greedy parameters}
d; Z_{min}; Z_{max};x;y;{//Coordinates parameters}
$epochs$;var;$Discount$;{//Replay memory parameters}

$DQNModel \leftarrow InitialiseDQLModel(modelparameters)$
$targetDQNModel \leftarrow InitialiseDQLModel(modelparameters)$
$(x_t, y_t, z_t) \leftarrow (x_1, y_1, Z_{min})$
$t \leftarrow 0$
while $(x_t, y_t) \neq (x_f, y_f)$ **do**
  $t \leftarrow t + 1$
  $SINR_t \leftarrow uav.GetCurrentSINR(x_t, y_t, z_t)$
  $s_t \leftarrow SINR_t, z_t, s_t$
  randomNum $\leftarrow RandomNumber(0 - 1)$
  if *randomNum* $> \epsilon$ *and* $t > 4$ **then** {//The model needs 4 last steps in the input}
    $a_t \leftarrow$ *DQNModel.predictMaxValue*(s_t)
  else
    $a_t \leftarrow$ *Random*$(+d, 0, -d)$
  end if
  {//Changing height according to action}
  uav.takeSelectedHeightAction(d,a, Z_{min}, Z_{max})
  $(x_t, y_t) \leftarrow (x_{t+1}, y_{t+1})$
  $r_t \leftarrow uav.GetThroughput(x_t, y_t, z_t)$
  if $\epsilon > \epsilon_{min}$ **then**
    $\epsilon \leftarrow \epsilon * \epsilon_{Decay}$
  end if
  $s_{t+1} \leftarrow a_t, r_t, s_t.removeState(s_{-4t})$
  $\beta \leftarrow StoreTransition(\beta, s_t, a_t, r_t, s_{t+1})$
  {//Applying replay memory}
  if $length(\beta) > \beta_{min}$ **then**
    $tempBatch \leftarrow BatchSample(\beta)$
    for *i in* $1 : length(\beta_{min})$ **do**
      $CurrentQValue \leftarrow$ *DQNModel.predict*$(tempBatch.state)$
      $FutureQValue \leftarrow$ *targetDQNModel.predict*$(tempBatch.nextState)$
      $maxFutureQ \leftarrow$ *FutureQValue.maxValue(i)*
      $newQ(i) \leftarrow$ *tempBatch.reward* $+ Discount * maxFutureQ$
      $newQTable \leftarrow CurrentQValue(tempBatch.a(i))$
      $newQTable \leftarrow$ *DQNModel.update*$(tempBatch.state)$
    end for
    $DQNModel.train(tempBatch.state, newQtable, epochs)$
  end if
  if $t\%var$ **then** {//Updating target model}
    $targetDQNModel \leftarrow setWeights(DQNModel.getWeights())$
  end if
end while

Then, update the Q value for the $tempBatch$ next state with the target model and save in the variable $FutureQValue$ in line 27. In line 28, for each value in $tempBatch$, we store the maximum Q value calculated by the target model in $maxFutureQ$. In order to update the new Q value in line 29, $newQ$, for each value in $tempBatch$, we weight the formula by $Discount$ the actual reward of the saved values with the calculated $maxFutureQ$. In possession of the $newQ$ value and the states from $tempBatch$, we calculate the new Q table in lines 30 and 31, $newQTable$, with the values of the chosen actions updated. Then we train the model with the $tempBatch$, the $newQTable$ a number of $epochs$ defined in the input. We then update, or do not update, the target model in line 35 to 37, and come back to the beginning of the loop. The target DQN model increases stability during the replay memory implementation, as the target network only updates its weights at each var step.

The code where we apply Algorithm 1 is available to the community in our public GitHub[1].

4.5 Experiment Design

The objective of our proposed algorithm is to maximise QoS to the UAV at all times, and to evaluate it. We use the data collected in the experimental flights described in 4.3, but in order to use the described data we needed to do some cleaning and pre-processing. The evaluation of our proposed height adaptation DNN model is done with a synthesis of the collected data. In this pre-processing, we do not interfere with any collected values for height, throughput or SINR.

[1]https://github.com/Erikagpf/DQN-for-UAV-height-adaptation.

For the synthesis, we have data in the NWQ area from ten different heights, and in the GCQ area nine different heights, each of them separated by 10 meters. As it is separated by 10 m, our increment d is 10. We merge this in a single dataset that simulates that the data was collected at the same horizontal coordinates at the same time. This enables us to know if the UAV can improve its throughput by changing its height. The goal of our experiment is to use part of this dataset to learn how to select the next movement by predicting which height the UAV needs to go to in order to achieve a higher SINR, then test in the unseen part.

The data was collected one after another on the same day for each location; however, the time of each flight is slightly different. As the data is collected at different times, meteorological conditions could impact the measured data. To be able to use the data, we ignore which time the data was collected. This assumption is needed because we let the UAV change its height in each time step (steps are 2 seconds); however, on the collected data, there is a considerable difference in time between them.

The horizontal coordinates of each height are also not exactly the same, the UAV was programmed to do the same path, but some factors, such as wind, made it move slightly from height to height. As we assume that the horizontal path only moves straight, we needed to make an approximation from the coordinates at each height. Moreover, measurements at certain time steps are missing due to the UAV not being able to measure as it was performing a handover, jumping 2 seconds on the table.

For a given flight path, segmented into parts with distinct heights, each segment can vary in the distance travelled. We only consider data points for a given segment where

they intersect with the segment with the shortest distance travelled. Otherwise we would not have a complete dataset. We call the flight segment with shortest distance the coordinates base, which for both areas was at 60 m. The coordinate base is the horizontal position we assume the UAV followed during our analysis at every step. We assumed that the measurements on the other heights were on the coordinate bases precisely coordinates. This means we assume that even if the horizontal coordinates changed slightly for different heights, such as half-meter far from the one at 60 m, we would consider it is the same as in the 60 m. We calculate the distance between the coordinate base and all the coordinates for each height. The smaller calculated distance is considered to be at the same point as the coordinate base. With this procedure, the UAV has the closest coordinates for all heights, and we can assume its path is approximately a straight line.

Each set of measurements at given coordinates corresponds to a single discrete step in our evaluation. The starting position z_1 is fixed, and afterwards, it is controlled by the action selection process of our proposed DQN algorithm described in Section 4.4 or by the baseline solution algorithm. Table 4.3 illustrates an sample of the data after the pre-processing. The data now does not have information of the time. The coordinates are aligned, the step is defined for each line, and it is defined that height of each measurement.

The hyper-parameters that provided the best results are illustrated in Table 4.4. However, we needed to perform a deep investigation to choose the hyper-parameters and design the model. We changed several of the hyper-parameters and inputs of the model until finding the proposed one. These parameters were experimentally selected among a number of model variations in which the number of layers, number of neurons per layer,

Step	Height	Latitude	Longitude	SINR Carrier 1 (dB)	Serving Cell Identity	Throughput (kbps)
1	20	53.34342193604	-6.23032475884304	10.3	60	46761.22
1	30	53.34342193604	-6.23032475884304	20.2	60	76651.78
1	40	53.34342193604	-6.23032475884304	11.1	60	46797.87
1	50	53.34342193604	-6.23032475884304	6.4	61	35082.27
1	60	53.34342193604	-6.23032475884304	3.4	61	29024.27
1	70	53.34342193604	-6.23032475884304	9.9	60	46738.80

TABLE 4.3: Sample of data from GCQ area after pre-processing to be used in the evaluation.

activation function, number of epochs, regularisation, and the inputs were varied. As we apply experience replay, the epochs are how many times the model is trained with the mini-batch at each time-step. Depending on the complexity of the DQN network (for example number of input features, number and size of layers), the training can be performed in a few steps, or require thousands or larger number of steps. However, for the UAV height adaptation scenario it is imperative to have as few training steps as possible, so that the model can learn to optimise height quickly in any new city environment is applied in. The value of epsilon decay in our evaluation is small compared to other DQN applications, which makes it learn quickly. In order to have a fast adaptation in a new environment, the model needs to adapt its weights quickly to not interfere with the UAV performance at the end of the path. For this evaluation, the value of ϵ and ϵ-decay are 1 and 0.9 respectively, effectively meaning that the proposed model trains in 30 steps. We apply replay memory as an strategy to accelerate the learning process, where at each step the model trains with the mini batch for the number of epochs.

Parameters	Value
Epoch	200
Epsilon	1
Epsilon_decay	0.9
Neurons per hidden layer	200
Number of hidden layers	3
Regularisation after hidden layers	RELU
Output layer	Softplus
Optimisation function	Adam

TABLE 4.4: Values of hyper-parameters for the proposed DQN model used in evaluation.

4.5.1 Baselines

We choose five different height selection strategies to which we compare performance of our proposed RL algorithm. For the first one, we use the baseline proposed by Zhang [19], which suggests that the UAV maintain the minimum allowed height during its flight. One of the most common approaches to UAV height selection is to maintain a constant height [8], [10], [11], [107], but there is no consensus on which height value to choose. To make a fair comparison, we also benchmark our approach against two constant height values. These heights will be the maximum possible height (120 m), and half of the maximum (60 m). When following these fixed height strategies, the UAV will begin at the minimum height at timestep 1, before increasing its height in each timestep until it reaches the required height, after which it will make no further adjustments.

To confirm that our solution is actually learning based on observed environment information and not acting randomly, we also compare it to a bounded Random walk height

selection strategy, in which the UAV in each timestep randomly selects one of three actions: increase the height, decrease the height, or keep the current height. It is bounded as all the solutions and cannot fly outside the allowed flight range.

In order to compare our solution to a more complex baseline, we implement an approach that we call One-step-ahead solution. In the One-step-ahead approach, the UAV knows whether the maximum SINR in the next time step will be found above or below its current height, and will move up or down (in a fixed increment of d = 10 m) depending on this knowledge. To be able to apply the One-step-ahead solution, the UAV needs previous information about the environment; this is not feasible in a real-world application, but we include this to assess whether and by how much such information would improve performance when compared to our RL approach.

We also compare our RL solution with one based on optimal height at each time step as obtained from the real-world dataset. In this approach, it is assumed that the UAV is able to move to any height in the next timestep, without restrictions of d. This represents the ideal-case performance which would not be possible in a real-world UAV application.

Bellow are benchmark approaches:

- **Zhang [19]**: this benchmark proposes that the UAV should maintain the minimal allowed height at all times.
- **Constant at 60 m**: this benchmark starts at the minimal height, like all others, and then moves up at every step until it achieves 60 m height. After achieving 60 m, the UAV should not move up or down.

- **Constant at 120 m**: this benchmark starts at the minimal height, like all others, and then moves up at every step until it achieves 120 m height. After achieving 120 m, the UAV should not move up or down.
- **Random walk**: this benchmark chooses its action randomly at each step.
- **Optimal height**: this is a reference benchmark of the maximum possible QoS values. In this approach, the UAV does not have any limitations on the maximum height change from step to step and also know which height has the maximum QoS.
- **One-step-ahead**: this benchmark follows the optimal height next position to decide its next action. If in the next step the optimal height is above the actual height of the UAV, the chosen action will be to move up. If in the next step the optimal height is bellow the actual height of the UAV, the chosen action will be to move down. In case the optimal height in the next step is the same as the actual height, the UAV should not move.

4.6 Evaluation Results and Analysis

We next evaluate how our RL approach can adapt the UAV heights with the objective to optimise the total throughput. We run the same algorithm 100 times over the data for this evaluation but always start the model from scratch, so it does not use the trained weights from the last run. Nevertheless, we do it so we do not overfit the model to the data and have a fair evaluation of the learning over an unseen scenario. We evaluate performance of our approach in two different sets of experimental data, NWQ with small

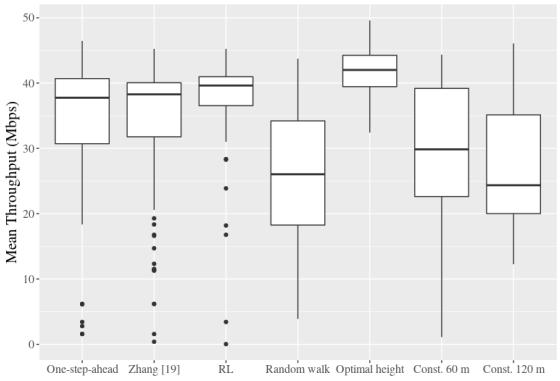
Approach	Throughput (Mbps)
Zhang [19]	35
Constant at 60 m	30
Constant at 120 m	30
Random walk	32 +- 2
One-step-ahead	35
Optimal height	43
RL	37 +- 1

TABLE 4.5: Mean throughput (Mbps) over 100 trials. Based on flight data obtained in the NWQ area.

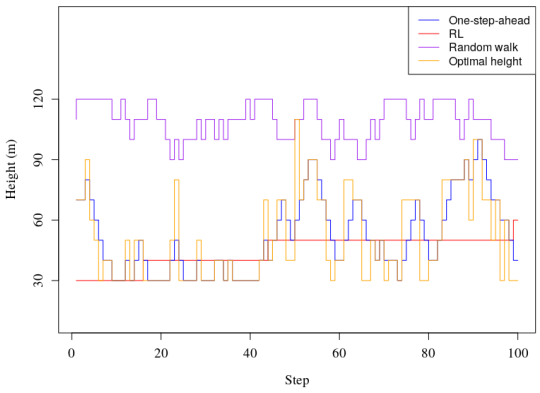
cell connectivity, Section 4.6.1, and GCQ with macro cell connectivity, Section 4.6.2. We start the evaluation with the throughput analysis, followed by the analysis of the height adaptation through the path. We evaluated the model after the training phase in this section. The results shown are related to the last 100 UAV steps.

4.6.1 NWQ analysis

Table 4.5 presents the average throughput of the investigated approaches; for non-deterministic solutions, which means the ones that might change at each run, we present a mean over 100 trials. We inspect the throughput as it is our parameter that we wish to optimise. By construction, the Optimal height at each step leads to the highest throughput. Therefore, we consider the Optimal height at each timestep to be the one with the highest throughout at that timestep. Our proposed approach achieves 37 Mbps with a variance of 1 Mbps, which is the highest throughput. One-step-ahead achieves 35 Mbps, that is the second highest. The approach proposed by Zhang [19] performs similarly to the One-step-ahead solution with 35 Mbps, with the added benefit of not needing a priory knowledge of



(A) Mean throughput in (Mbps) in NWQ area.



(B) Height adaptations for a UAV flight over the last 100 steps in NWQ area.

FIGURE 4.4: Performance of the benchmarks and the proposed RL approach in the NWQ area.

the environment. Nonetheless, our proposed RL approach provides the best throughput and outperforms Zhang [19] and the One-step-ahead benchmarks by 6%, also resulting in lower variation in UAV heights when compared to the One-step-ahead approach. It is worth noting that the solutions that maintained large heights, as Constant at 60 and 120 m, do not perform well when compared to those that maintained lower heights. One possible explanation for this is that at greater heights a UAV might have been experiencing increased interference from cells it was not connected. Another possibility is antenna misalignment: as the small cells are designed for ground users, their antennas are directed towards the ground, which means that the aerial UAV receives signals primarily from antenna side-lobes.

Figure 4.4 evaluates an example run, different than Table 4.5 that evaluates the approaches performance after 100 trials. To generate Table 4.5 we needed to calculate the mean over the throughput mean of each run, losing information of the throughput variation through the path. With the analyses of one single run, we can verify how the throughput and height vary through the path. Figure 4.4a presents box plots for the throughput in Mbps for all approaches obtained across the last 100 steps of one example run. Our RL approach shows a stable value for the obtained throughput, with its first and third quartile being 36 to 41 Mbps (the box denotes that 50 % of the data is in this range), respectively, and with median 40 Mbps. On the other hand, one can observe a considerable interquartile range from 18 to 34 Mbps in the throughput for the Random walk approach, as well as for the approaches that maintain the height Constant at 60 m and 120 m. This more significant variance is likely due to the randomness of the Random walk approach and to the fact that at greater heights of the constant strategies, the coverage from several cells is more unpredictable, as the UAV may be connecting to the side lobes of different antennas. Approaches as One-step-ahead and Zhang[19] have a bigger interquartile when compared to the proposed RL approach, with Zhang [19] being between 32 to 40 Mbps, One-step-ahead between 31 to 41 Mbps, and the RL approach between 36.5 Mbps to 41 Mbps. Although in the One-step-ahead, Zhang [19] and RL happens outliers (in the figure represented as the dots outside the box) that means that at some points of the path, the measured throughput was much lower than most of the path. Interestingly, the Optimal height median throughput is only 6% better than our RL-based approach, despite it unrealistically assuming instant jump from any height to any other height is possible, showing

that the proposed method is close to optimal.

Figure 4.4b shows how the different adaptive strategies adjusting the UAV height at different steps in a single sample run for the last 100 steps. We inspect the individual height adaptation to understand how each of the approaches behave in a real path and have an idea of how many adaptation were needed to achieve their respective throughput. We do not illustrate Zhang [19], Constant at 60 m and Constant at 120 m because their values are constant. We can observe that our proposed RL-based solution maintains the UAV height low all the path, with only 3 changes in the UAV height on the last 100 steps. On the other hand, we can see that the Optimal height at each step changes substantially, indicating that even if one knew in advance at which height the optimal connectivity was obtained, the UAV would not be capable of reaching these heights in every timestep, as the height change from one step to another could be in the order of 90 m. The One-step-ahead approach follows the Optimal height, and also moves constantly trying to achieve the Optimal height approach. In this example, the Random walk approach started the last 100 steps at higher heights and it moved randomly through the steps in a up and down movement, and sometimes, did not move, as expected.

4.6.2 GCQ analysis

Table 4.6 shows the average throughput for the GCQ area over 100 trials. Same as in NWQ, we aim to analyse the throughput as it is the variable that we intent to optimise. The Random walk approach provided a throughput of 50 Mbps, better then the constant approach at 60 with and 120 m that achieve. The constant approaches that lead to the UAV

Approach	Throughput
Zhang [19]	68
Constant at 60 m	41
Constant at 120 m	41
Random walk	50 +- 4
One-step-ahead	68
Optimal height	83
RL	70 +-2

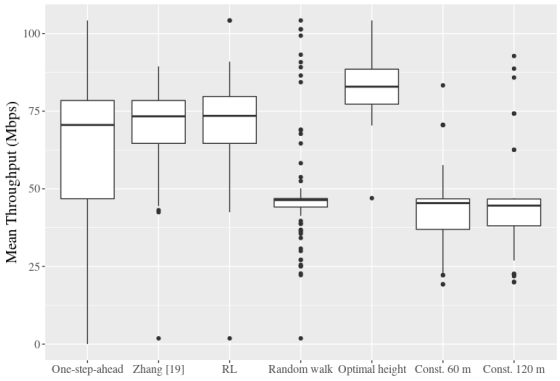
TABLE 4.6: Mean throughput (Mbps) over 100 trials. Based on flight data obtained in the GCQ area.

flying at larger heights result in lower throughput compared to all other approaches, obtaining 41 Mbps, which is only 59% of the throughput achieved by our RL approach. In this scenario, our RL solution also performed better than all benchmarks achieving 70 Mbps in average, while the Zhang [19] approach and One-step-ahead being in second, achieving 68 Mbps. The results of the One-step-ahead approach show that having a priori knowledge of the environment is sometimes not enough to provide the best throughput. As a reference, the Optimal height achieved around 19% better throughput than the proposed RL approach, which showed to be considerate more than in NWQ area. One explanation of the difference in the distance between the Optimal height and the other methods is due to the fact that the optimal approach changed more drastically its height through the path, making it impossible for any other approach to achieve closer to the same throughput as they were limited by "d".

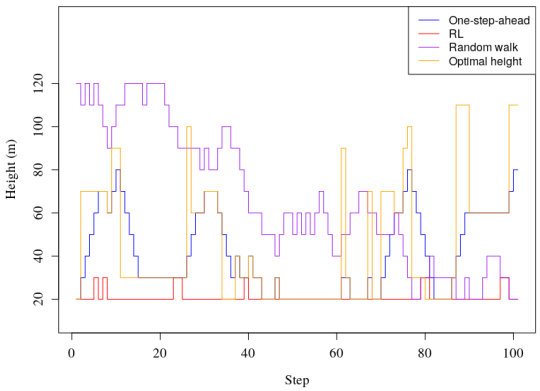
As in the NWQ area, Figure 4.5 evaluates an example run, different than Table 4.6 that evaluates the approaches performance after 100 trials. In Figure 4.5a, we investigate the stability of each of the approaches, with the box plot representing throughput across

last 100 steps. Both, RL and Zhang [19] approaches, achieve median throughput of 74 Mbps, as well as exhibiting low variance. Both achieve the lower quartile at 65 Mbps, but at the third quartile, the RL proposed approach provides 2 Mbps more than Zhang [19], meaning that it provided better throughput for some time in the path. This behaviour is similar to the one in the NWQ area, although the throughput results for the other baseline approaches are significantly different. In particular, the approaches that keep the UAV height Constant at 60 and 120 m show lower variance than for the data set obtained in the NWQ area. Possibly this difference is because the UAV connects to only one macro BS in NWQ area, which leads to greater stability in the throughput. On other hand, One-step-ahead provides high variance through its path, with its median being close the the proposed RL approach, in 70 Mbps, and its first and third quartile been between 47 Mbps and 78 Mbps. The Random walk approach shows a small variance on its quartile, although it also shows many outliers. As the behaviour is random, the outliers showed a significant variation of the throughput. However, on average, it manages to maintain a throughput near its median of 46 Mbps.

Figure 4.5b illustrates how the different strategies adjusted the UAV heights when flying in the GCQ area. As in the NWQ area, we inspect the individual height adaptation to understand how each of the approaches behave in a real path and have an idea of how many adaptation were needed to achieve their respective throughput for the GCQ area. Here, we observe that our proposed solution maintains a low height when flying near a macro cell deployment maintaining its height at 20 m or 30 m at all times. Also, we note that the Optimal height at each step requires significant changes in the UAV height from



(A) Mean throughput in (Mbps) in GCQ area.



(B) Height adaptations for a UAV flight over the last 100 steps in GCQ area.

FIGURE 4.5: Performance of the benchmarks and the proposed RL approach in the GCQ area.

step to step for example. The One-step-ahead approach follows the Optimal height and moves up and down 50 times for these 100 steps. In this example run, the Random walk approach starts at a higher height and keeps moving randomly until move to the lower heights.

4.7 Conclusion

In this chapter, we presented contribution C2 of our thesis, that is a RL-based approach to optimise throughput by adapting height at which a mobile cellular connected UAV should fly in order to improve its QoS. Our primary objective was to increase the UAV's average throughput in order to answer RQ2 "How to adapt the UAV height in order to increase

UAV's QoS?". The environment is complex, with the received channel quality affected by distance attenuation of the signal, building blockage, interference, and the antenna gain of both the BS and the UAV antennas. Changing the height will change all of these factors, some will improve, and some will deteriorate. Because of this complexity, it is not evident if one constant approach will always provide better throughput; the height needs to be adjusted for each individual environment.

We evaluated our approach using real data obtained from a UAV carrying a smartphone measurements in two locations of Dublin city centre [104]. Our proposed solution was shown to be successful in both test environments, providing an improvement of 5% compared to other approaches, including the ones that had access to additional a priori information about the environment.

However, there is a threshold to be considered when using the proposed solution. For example, if the UAV need to inform the local authorities its exact location prior to the flight, a good approach would be the one proposed by Zhang [19], where it maintains the lowest possible height through the flight. However, if the QoS of the connection is mission-critical UAV priority and the UAV can adapt its location during the flight, the UAV could use the proposed RL solution. The code used in this chapter, the one for aligning the real-world dataset and the proposed approach are public available to the community in our GitHub[2].

[2]https://github.com/Erikagpf/DQN-for-UAV-height-adaptation

Chapter 5

Adaptive Height Optimisation for Connected UAVs - A Simulation Investigation

In this chapter we will investigate the height adaptation DQN model in a simulated environment, where we can change the building and BS densities. Here we expand the investigation done with real-world data in Chapter 4 in order to generalise the proposed C2. In this chapter we apply RL to dynamically adapt the UAV altitude in order to improve its QoS, across different environments with different building and base station densities.

In Section 5.1, we introduce the changes in the solution needed to investigate the proposed approach to C2 in the simulation environment, and in Section 5.2 we introduce the system model of the simulated environment. In Section 5.3, we introduce the minor modifications of the proposed solution for the simulation environment. The adaptation was needed because of the difference of the data we could access from the real-world collected

data and the environment. In Section 5.4, we explain the experiment design used to evaluate our proposed solution in the simulation environment. In Section 5.5, we provide a detailed investigation performance of the proposed model, and Section 5.6, concludes the chapter.

5.1 Problem Statement

We consider an urban scenario where a UAV flies while connected to the cellular network, as in Chapter 4. The UAV's initial position (x_1, y_1, z_1) and final position (x_f, y_f, z_f), where x refers to the coordinate of the UAV on the horizontal X axis, and z refers to its height, are known at the beginning of each topology (denoted as different distribution of BS and building). The BS distribution follows a Poisson Point Process (PPP) with $\Phi = \{(x_1, y_1), (x_2, y_2), ...\}$ of intensity BS_{dens}, at a height z_b above ground.

As in Chapter 4, we consider the height range defined by safety regulation, with the minimum allowed height denoted as Z_{min} and the maximum allowed height as Z_{max}. We assume that the UAV starts at Z_{min} in order to avoid any unnecessary movement as in the last chapter. Figure 5.1 shows UAVs horizontal and vertical movements in this simulation. Figure 5.1a illustrates the possible path of the UAV, where d is the maximum distance the UAV can move up or down in each step. It is an representation of a limitation of how much a UAV can move realistic up or down and horizontally in a time-step.

Our main objective is to optimise the z coordinate in order to improve the QoS experienced by the UAV, same as in Chapter 4. The metric used to represent the QoS during the simulation is spectrum efficiency on the downlink. We followed the downlink direction

to follow the same approach we had for the experimental evaluation, but in both cases, the proposed solution could use the representation of the uplink spectrum efficiency. We use spectrum efficiency instead of throughput because of the limitations of the simulator. We use a simulator based on [108], and used in [13], [39], [109], that simulates the physical layer and the UAV flight. In order to simulate the throughput, we would need to implement other layers of the mobile network, as the allocation of the resource blocks, that are not in the scope of this work. As this implementation is not the main objective of our investigation for the validity of our approach, we use the spectrum efficiency, as in other works in the literature [12], [13], [39], [109].

5.2 System Model

In this section, we introduce how we model the height optimisation problem in the simulator environment.

5.2.1 UAV and BS Antennas

The UAV is equipped with one omnidirectional antenna to connect to a serving BS and receive data. The antenna has an omnidirectional radiation pattern, and it has an antenna gain equal to 1. We express the coordinates of the BS which the UAV is associated with as $b_s = \{x_s, y_s\} \in \Phi$ and its horizontal distance to the UAV as r_s.

The BS has a directional antenna with a horizontal and vertical beam-width ω along with a rectangular radiation pattern; The antenna gain is defined as $\eta(\omega) = 0$ outside of the main lobe; and $16\pi/(\omega^2)$ inside of the main lobe.

Spectrum efficiency is the maximum bit rate that can be transmitted per unit of bandwidth and is a measure of the quality of service that can be served by that part of the network. The Shannon–Hartley theorem bounds the maximum achievable rate a user can reach once it establishes a wireless link. As we want to improve user's experience providing reliable connectivity to UAVs, our purpose is to increase spectrum efficiency. We calculate the spectrum efficiency value for the calculated SINR based on Shannon–Hartley theorem. The SINR is a function of the antenna gain and channel model and given as:

$$SINR = \frac{p\eta(\omega)c(x_s^2 + \Delta\gamma^2)^{-\alpha_{t_s}/2}}{I_L + I_N + \sigma^2} \tag{5.1}$$

where p is the BS transmit power, α_{t_s} is the pathloss exponent, $t_s \in \{\mathrm{L}, \mathrm{N}\}$ indicates whether the UAV has LoS or NLoS to its serving BS x_s, c is the near-field pathloss, σ^2 is the noise power, and I_L and I_N are the aggregate interference from LoS and NLoS, respectively.

5.2.2 Horizontal route adaptation

The UAV horizontal route is extended in the simulation and it is defined by an independent approach that focuses on bringing the UAV closer to the BS it is connected. The UAV flies in direction to its final destination but approximates its Y trajectory to get closer to the BS that it is connected by d. At every time step, the UAV connects to the BS with stronger SINR and get closer in the Y coordinates to this BS by d, being maximum of d distant to the straight line between (x_1, y_1) and (x_f, y_f) as illustrated in Figure 5.1b. The focus of our approach is to investigate if the approach is able to adapt the height of the UAV and can

adapt to any underlying routes decision that a UAV might take during its path, showing its independence from the horizontal path decisions.

5.2.3 Building distribution

The buildings distributed in the area might affect the UAV LoS, as they can block the channel between the UAV and the BSs. In order to check if a signal is in LoS or not, we verify if there is a tall enough building between the UAV and BS. If the signal is blocked by a building, NLoS, it causes the signal to be attenuated, which is reflected in the SINR expression in Equation 5.1. We needed to defined this expressions as LoS and NLoS affect the calculation of the SINR. But in real-world implementations there is no need for this information, as the SINR is measured. We use a commonly-adopted model for the urban environment which models the buildings as a square grid with the locations of building centerpoints (x_{bl}, y_{bl}), that was presented in [110] and used in works as [80], [81], [111]. The area occupied by each building, Bl_a, is constant, and the density of buildings, $Build_{dens}$, is denominated by the number of building per square kilometre. The individual building height, h_bl is randomly distributed according to a Poisson distribution, with scale parameter a.

5.2.4 UAV-BS Link

The UAV connects to the BS with the best SINR at all times. Therefore, as the UAV moves through the environment some BSs become stronger and others weaker. When it reaches the point where its serving BS is no longer the BS with strongest signal, it will reconnect to

the new LoS with the strongest signal. We assume that this handover occurs seamlessly, and there is no disconnect or loss of signal quality when it happens.

We assume that the UAV will have access to the SINR measurements from the BS it is connected to and from the 5 neighbours BS with strongest signals, the spectrum efficiency it is achieving with the serving BSs, and its height at all steps. SINR and throughput data is easily obtained by the UAV from its cellular connection, while the height information is obtained via other UAV sensors located on the UAV, as already present in UAVs as the DJI-mini 2 [3].

5.3 Proposed Solution

The algorithm proposed to solve the height optimisation in the simulation environment is similar to the one presented in Section 4.4. The main adaptation we made from the proposed solution presented in Section 4.4 is that we use as input for the model the SINR of neighbours BSs - $SINRn$. We included this information to the model as a normal UE has access to it when doing the measurement report. With this, we expected to have more information of the environment. In Chapter 4, we did not use the neighbours' data as input because we did not have access to this information from the experiment.

In summary, the state space S assumed in this chapter is:

- Height z: which is obtained by UAV sensors and it is relevant for UAV's decision-making process, as in order to know whether to move next and stay within the hard limits.

- Received $SINR$: which is obtained by UE sensors to perform the measurement reports. This value impacts the UE QoS that is what we intent to maximise.
- Received Neighbours $SINRn, R \in [1, 2, 3, 4, 5]$: which is the SINR of the BSs with the strongest signals. It is obtained by UE sensors while performing the measurement reports. These values impacts the UE QoS that is what we intent to maximise.
- 4 last z, $SINR$, $SINRn$, a, r: which will be stored from the previous steps. In order to achieve better optimisation, we extended the state with the four previous z, $SINR$, $SINRn$, action a and reward r, following the lines of the original DQN implementation [56], as well its implementation in UAV connectivity [39].

As it is showed in Algorithm 2, the algorithm and most of the parameters of the designed RL solution is the same as in Algorithm 1. The main changes are the lines 8 and 9 that is the collection of the neighbours SINR data and its use as part of the state information. And the reward function in line 24, that on the simulation is the spectrum efficiency. All other steps are the same as in Algorithm 1. Our objective was to change the lest possible the solution proposed in Chapter 4 and also use values that are easily available to a UE and to a UAV.

5.4 Experiment Design

To evaluate our proposed approach and analyse how the city topology influences the connected UAV, we utilise the simulator developed in R [12]. In this experiment we want to vary the building and BS densities in order to evaluate the generalisation of our model.

Algorithm 2 RL-based algorithm for UAV height optimisation

Input: β; β_{min}; {//Batch parameters}
ϵ; ϵ_{Decay};ϵ_{min}; {//ϵ-greedy parameters}
d; Z_{min}; Z_{max};x;y;{//Coordinates parameters}
$epochs$;var;$Discount$;{//Replay memory parameters}

$DQNModel \leftarrow InitialiseDQLModel(modelparameters)$
$targetDQNModel \leftarrow InitialiseDQLModel(modelparameters)$
$(x_t, y_t, z_t) \leftarrow (x_1, y_1, Z_{min})$
$t \leftarrow 0$
while $(x_t, y_t) \neq (x_f, y_f)$ **do**
  $t \leftarrow t + 1$
  $SINR_t \leftarrow uav.GetCurrentSINR(x_t, y_t, z_t)$
  $SINRn_t \leftarrow Get.SINR.Neighbours(x_t, y_t, z_t)$
  $s_t \leftarrow SINR_t, z_t, SINRn_t, s_t$
  randomNum $\leftarrow RandomNumber(0 - 1)$
  if *randomNum* $> \epsilon$ *and* $t > 4$ **then** {//The model needs 4 last steps in the input}
    $a_t \leftarrow$ *DQNModel.predictMaxValue*(s_t)
  else
    $a_t \leftarrow$ *Random*$(+d, 0, -d)$
  end if
  {//Changing height according to action}
  uav.takeSelectedHeightAction(d,a, Z_{min}, Z_{max})
  $(x_t, y_t) \leftarrow (x_{t+1}, y_{t+1})$
  $r_t \leftarrow uav.GetSpectrumEfficiency(x_t, y_t, z_t)$
  if $\epsilon >$ ϵ_{min} **then**
    $\epsilon \leftarrow \epsilon * \epsilon_{Decay}$
  end if
  $s_{t+1} \leftarrow a_t, r_t, s_t.removeState(s_{-4t})$
  $\beta \leftarrow StoreTransition(\beta, s_t, a_t, r_t, s_{t+1})$
  {//Applying replay memory}
  if $length(\beta) > \beta_{min}$ **then**
    $tempBatch \leftarrow BatchSample(\beta)$
    for *i in* $1 : length(\beta_{min})$ **do**
      $CurrentQValue \leftarrow$ *DQNModel.predict*$(tempBatch.state)$
      $FutureQValue \leftarrow$ *targetDQNModel.predict*$(tempBatch.nextState)$
      $maxFutureQ \leftarrow$ *FutureQValue.maxValue(i)*
      $newQ(i) \leftarrow$ *tempBatch.reward* $+ Discount * maxFutureQ$
      $newQTable \leftarrow CurrentQValue(tempBatch.a(i))$
      $newQTable \leftarrow$ *DQNModel.update*$(tempBatch.state)$
    end for
    $DQNModel.train(tempBatch.state, newQtable, epochs)$
  end if
  if $t\%var$ **then** {//Updating target model}
    $targetDQNModel \leftarrow setWeights(DQNModel.getWeights())$
  end if
end while

Variable	Value
Z_{min}	40 m
Z_{max}	240 m
BS_{dens}	$[1, 2.5, 5]/km^2$
Bs height z_b	$30m$
BS downtilt ϕ	30
$Build_{dens}$	$[1, 2, 3] * 100/km^2$
Building area Bl_a	40 m^2
Building height parameter a	20 m
Simulated area	3 km^2
Horizontal Speed	10 m/s
Timestep	1 s
Distance between x_i and x_f	1000 m
Step	100
d	10 m
Mctrials	100

TABLE 5.1: Simulation parameters.

A Mctrial, or Monte Carlo simulation, is a model used to predict the probability of different outcomes once the simulation environment changes its characteristics randomly. In Chapter 4, we had the equivalent of 2 Mctrial simulations, and we tested the model on that environment 100 times, starting the model from scratch at each time, so it would not overfit. In order to evaluate the model in different environments, for each MCtrial, we randomly generate the BS and buildings distribution, so the UAV always observes a "new environment" at every 100 steps. During the simulation, we do not vary the BS and the building densities at the same time, always letting one of the variables to remain fixed so we can evaluate the impact of each. The chosen value to be the fixed is the mean of the list values, and shown in Table 5.1 with other simulation parameters. A step (or timestep)

is a single movement (action taken) of the UAV in a specific BS and building topology. We assumed the minimum allowed height Z_{min} as 40 m, because of simplification for the simulation, as with the UAV achieving the minimum of 40 m it would above the buildings. We wanted to investigate if a increase of the maximum height Z_{max} allowed for the UAV to fly could be beneficial to its QoS, then we increased from 120 m in Chapter 4 to 240 m in the simulation. The BS density BS_{dens}, BS height z_b, building area Bl_a, building density $Build_{dens}$ and building height are based on the ITU specification [112].

In the beginning of a simulation z_1 is fixed for all approaches, and afterwards it is controlled by the RL algorithm action selection process or the baseline solution algorithm. We choose five different height-selection strategies to benchmark the performance of our approach. These baselines are the ones presented in Section 4.5.1, named: Zhang [19], One-step-ahead, Random walk, Optimal height, Constant at 120 m and Constant at 240 m. The constant height was changed from Chapter 4 as we assume a higher maximum allowed height in the simulation. In Chapter 4, we considered 120 m as the maximum height the UAV could fly, as it was the maximum allowed to fly in that region. However, in a simulation environment, we increase it to investigate if the regulations should consider a higher approach for some BS and building densities.

5.5 Evaluation Results and Analysis

We next evaluate how our RL approach can adapt the UAV heights with the objective to optimise the total spectrum efficiency. We run the same algorithm in 100 different Mctrials, simulating 100 different cities for each BS and building density. The evaluation

always start the model from scratch, so it does not use the trained weights from the last run. Nevertheless, we do it so we do not overfit the model to the data and have a fair evaluation of the learning over an unseen scenario.

The main points that we want to evaluate are how the BS density and building densities influence the optimal height of a connected UAV. In order to assess each of these factors separately, we divide this section in two sections. First, we analyse the spectrum efficiency by BS density and building density. Then, we inspect height changes within each approach.

5.5.1 Spectrum efficiency

In this section we analyse the mean of spectrum efficiency per unit of bandwidth, that is a mean of the spectrum efficiency over an entire episode, for varying BS densities and building densities. We inspect the spectrum efficiency as this is the parameter that we wish to optimise.

Varying BS densities

To demonstrate how the RL solution can have its performance affected by different BS densities. We study in detail three different BS densities $(1, 2.5, 5)/km^2$, denoted as low, medium and high, as illustrated in Figure 5.2.

Figure 5.2a shows the mean spectrum efficiency per approach. As expected, the optimal height provides much better spectrum efficiency, achieving median of 23 bits/s/Hz. This happens because it does not have any movement restriction, being able to move any

distance from step to step. For low BS density, One-step-ahead, Zhang [19] and the proposed RL approach perform similarly, with all archiving median of 20 bits/s/Hz. The constant height at 240 m is the approach with the worse spectrum efficiency, with 15 bits/s/Hz, showing that high heights for low BS density do not perform as good as other approaches do. Constant at 120 m performed slightly worse than the Random walk approach, with median of 18.5 bits/s/Hz and Random walk approach with 19 bits/s/Hz. It is interesting to note that the approaches do not vary much its mean spectrum efficiency, and all have a relatively small first and third quartile of around 2 bits/s/Hz, with exception of Constant at 240 m with 4 bits/s/Hz.

Figure 5.2b shows that our RL approach performs better, 4%, then Zhang [19] for medium BS density, and 26% better than Constant at 120 m, Constant at 240 m and Random walk. It indicates that maintaining higher heights at all times provides worse spectrum efficiency for the medium BS and building densities when compared to the proposed RL approach that adapts the height dynamically to the environment. One-step-ahead showed the best performance compared to the approaches that could only move "d", achieving 14.5 bits/s/Hz, showing that for medium BS density having previous knowledge of the radio characteristics of the environment can improve the UAV QoS.

When investigating the high BS density in Figure 5.2c, Constant at 120 m and Random walk are the worst solutions achieving 3.5 bits/s/Hz, with the Zhang [19] being slightly better than them, showing that maintaining the lowest altitude for all topologies is not the best approach. The proposed RL approach shows performance comparable to Constant at 240 m, with 4% better performance. Therefore, its third quartile is higher, which

means that the RL performed better in more simulations. The One-step-ahead approach showed the best performance with its median achieving 9 bits/s/Hz, showing the previous knowledge of the environment can improve UAVs QoS. However, it is unrealistic to expect to have this knowledge for each set of coordinates in the environment.

We have used Figure 5.2 to analyse how each approach compares to each other, now Figure 5.3 shows the data in a different format to easier show how the BS density impacts all the approaches performance. The figure shows that the general mean spectrum efficiency for low BS density is much better than for medium and high BS density, with solutions archiving near 20 bits/s/Hz. We can also analyse that One-step-ahead and the proposed RL solution are always the best approaches for all densities, showing that an intelligent and adaptable decision can provide a good QoS for all densities. Moreover, the proposed RL solution can adapt its response to the environment on the fly without previous knowledge.

Varying building densities

Figure 5.4 illustrates the spectrum efficiency for low and high building density. In Figure 5.4a, the One-step-ahead provides the best approach achieving median of 15 bits/s/Hz, and the proposed RL approach is the second best with 12.5 bits/s/Hz. We can observe that Zhang [19] approach achieves 11.7 bits/s/Hz, that is 6% worse than the proposed RL solution. The Constant at 240 m performs as well as the Constant at 120 m, and both are worse than all other solutions, which show a deterioration for those heights, implying that the UAV would be most of the time in a poor coverage area. Random walk approach

performed slightly better then the higher constant approaches, showing that the Random walk movement of the UAV is comparable to maintaining high constant values.

Figure 5.4b illustrates the mean spectrum efficiency for high building density. It shows a similar pattern when compared to the low building density, with One-step-ahead being the best approach and the proposed RL solution being slightly better, 2%, than Zhang [19]. We can conclude that since it has no impact, it is providing an indication that building density is not a factor that needs to be taken account when determining UAV's height. It shows that that same approach should work in density urban areas and rural ones.

Figure 5.5 is a macro view of the performance of the solutions over the building densities in order to show how the building density impacts all the approaches performance. The medium density is the same as in Figure 5.3, but we leave it in Figure 5.5 as it makes it easier to analyse the overall performance. As an overall performance, we discovered that the difference in the building density when the UAV is flying above the buildings did not influence the mean spectrum efficiency as the approaches performed similar in all the distributions.

5.5.2 Height variation

While in the previous section we focus our analyses on the spectrum efficiency of each approach, in this section we inspect in more detail underlying height variations that achieve the discussed performance. Results average from 100 Mctrials, although to illustrate the height changes in a single path, we randomly selected two single paths from those 100,

and illustrate them in Figure 5.6. We inspect the individual height adaptation to understand how each of the approaches behave in a real path and have an idea of how many adaptation are required to achieve their respective spectrum efficiency. Example one in Figure 5.6a shows how the RL approach makes adaptations in order to increase its QoS maintaining lower heights. In Figure 5.6b, the RL approach tends to go to a higher altitude, maintaining it for most of the path, changing more to the end where it comperes to the same height as the optimal approach. Figure 5.6 illustrates how much the optimal height changes substantially UAV's altitude, as in Chapter 4, showing that even if one knows the optimal height at each step, it would not be possible to reach it. It is possible to see that the One-step-ahead approach changes often UAV's height. As expected, the Random walk approach moves randomly maintaining a higher height in 5.6a and a lower in 5.6b. The constant approaches take the first few steps to move in d increments from its starting height Z_{min} to the height specified in constant approach, ie 120 or 240 m.

To make a more detailed investigation over the 100 Mctrials, Figure 5.7 illustrates the mean of the heights for different BS and building densities. The constant approaches have no variance on the height after they achieve their constant heights. In Figure 5.7a, the average height of the optimal height approach varies with the BS density, being lower for low BS density, and higher for high BS density. As we can notice, the intelligent approaches, One-step-ahead and the proposed RL solution, adapt their altitude to the one that better serves the BS distribution, also increasing its heights when the BS density increases. The Random walk approach, as it does not consider any information of the environment, it also maintains, in average, the same height in all cases.

When we analyse in Figure 5.7b the height adaptation by the building density, the optimal height is not related with the density. The approaches does not change its mean height considerably during the different building densities. The RL approach varies from 68 m in medium building densities, to 83 m in high building densities.

Observing behaviours for both BS and building densities, we conclude that RL is a competent approach to solve UAV height optimisation. As we can see in Figure 5.7, the RL solution demonstrated to be learning the best height, resulting in a spectral efficiency improvement. We can also conclude that the RL approach does not make changes on its height at all steps, making intelligent changes when needed and avoiding spending extra energy to move its height at all steps.

5.6 Conclusion

In this chapter, we generalised the proposed RL approach to solve UAV height optimisation given different BS and building densities for a moving UAV which is acting as a user of the cellular network. Our main focus was to to analyse performance of the proposed approach while varying BS and building densities by allowing the UAV to vary its height inside a range. We answer RQ2 "How to adapt the UAV height in order to increase UAV's QoS?", and evaluate C2 in various scenarios. We extended the approach presented in Chapter 4 considering multiple BS and building densities, in the UAV horizontal movement, and on the available data the UAV has access to.

In our analysis we concluded that the low densities of BS can provide spectrum efficiency, in general, better then in any other configuration. This happens as there is lower

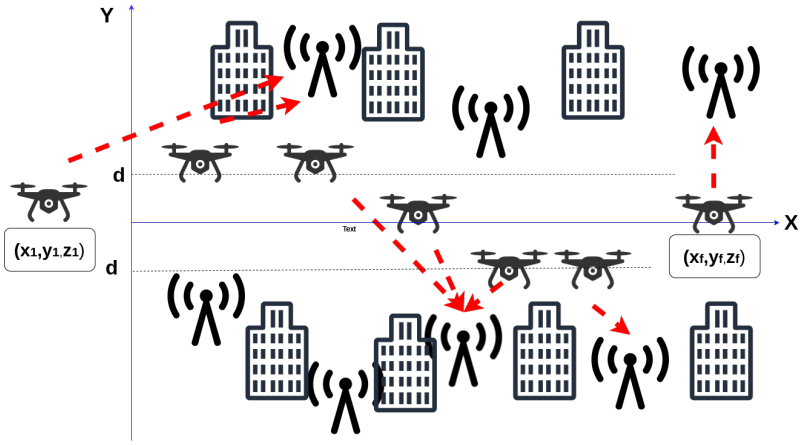
interference from neighbour BSs. We could also notice that the approach proposed in Zhang [19] is a simple approach and usually provides good spectrum efficiency, showing to not be adequate mostly for high BS density. The proposed RL model showed to learn how to adapt its height to improve the UAV spectrum efficiency for all densities.

Therefore, this chapter confirms our conclusion from Chapter 4, that the RL model for height adaptation can provide consistent good QoS in all the situations, learning which is the best behaviour. The results showed that with the proposed RL approach, the UAV could achieve its best throughput without previous information of the environment, showing that the RL is an adaptable solution.

The proposed approach for simulated data discussed in this chapter is public available to the community in our GitHub[1].

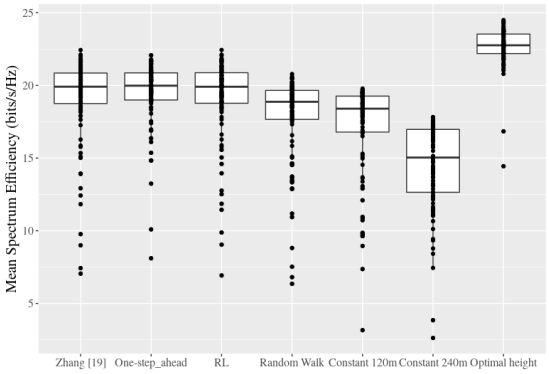
[1] https://github.com/Erikagpf/DQN-for-UAV-height-adaptation

(A) Side view of a UAV connected to the mobile network adjusting its height to maximise its spectrum efficiency.

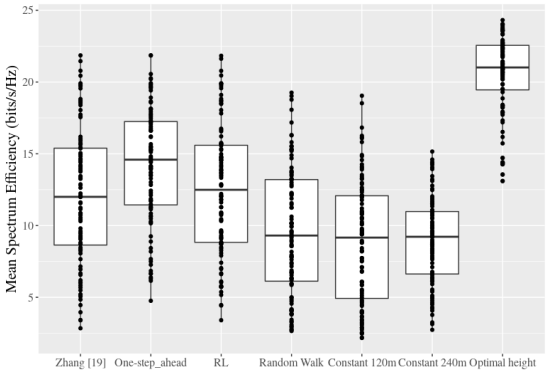


(B) Top view of a UAV connected to the mobile network when it moves closer to the connected BS through the path in an urban area from position (x_1, y_1) to the final position $((x_f, y_f))$. In red shows which BS the UAV is connected at each point of the trajectory.

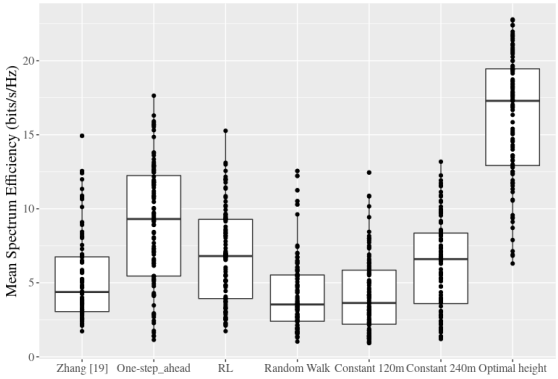
FIGURE 5.1: UAV vertical and horizontal movement assumed in the simulation.



(A) Spectrum efficiency per unit of bandwidth (bits/s/Hz) for low BS density.



(B) Spectrum efficiency per unit of bandwidth (bits/s/Hz) for medium BS density.



(C) Spectrum efficiency per unit of bandwidth (bits/s/Hz) for high BS density.

FIGURE 5.2: Spectrum efficiency per unit of bandwidth (bits/s/Hz) for 3 different BS densities and medium building density.

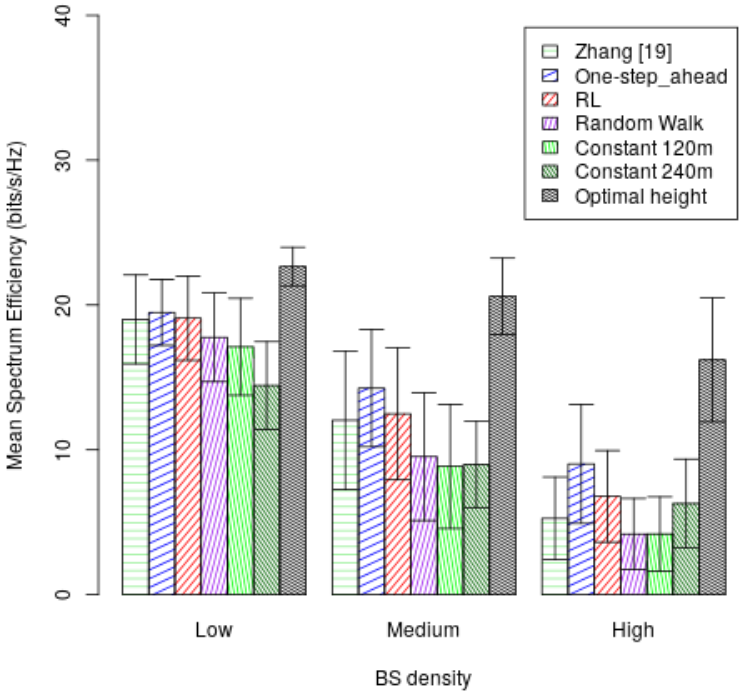
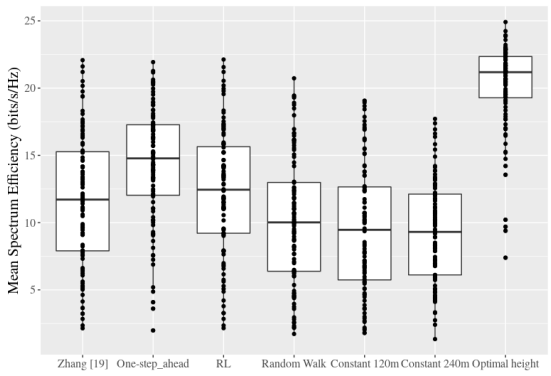
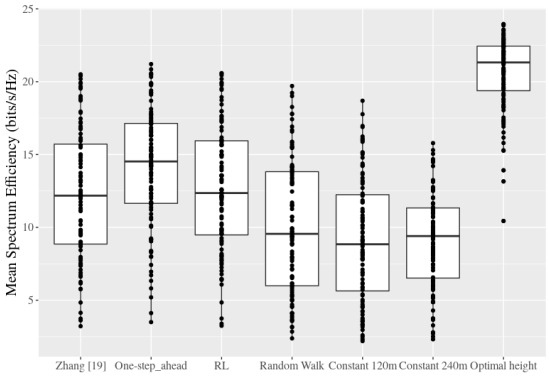


FIGURE 5.3: Average Spectrum efficiency per unit of bandwidth (bits/s/Hz) per BS density (low, medium, and high), and medium building density.



(A) Spectrum efficiency per unit of bandwidth (bits/s/Hz) for low building density



(B) Spectrum efficiency per unit of bandwidth (bits/s/Hz) for high building density.

FIGURE 5.4: Spectrum efficiency per unit of bandwidth (bits/s/Hz) for different building densities with medium BS density.

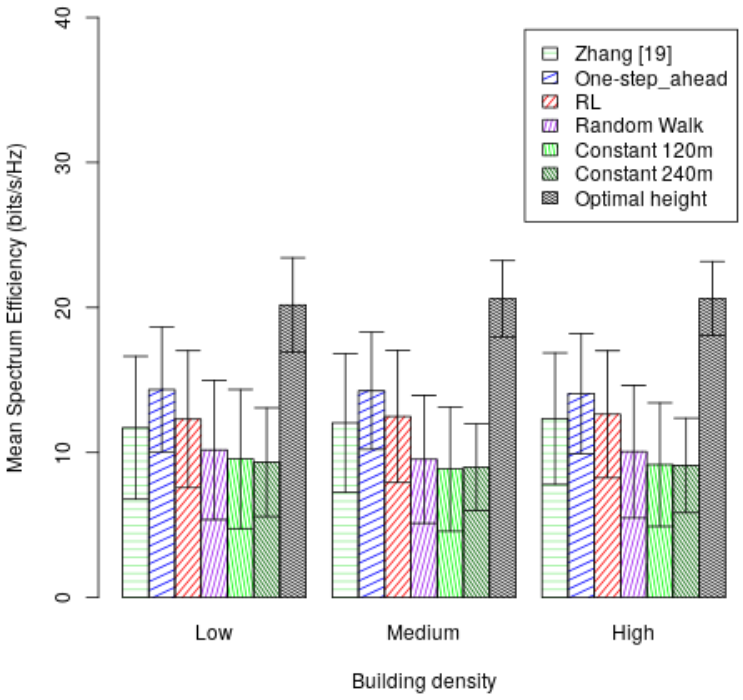
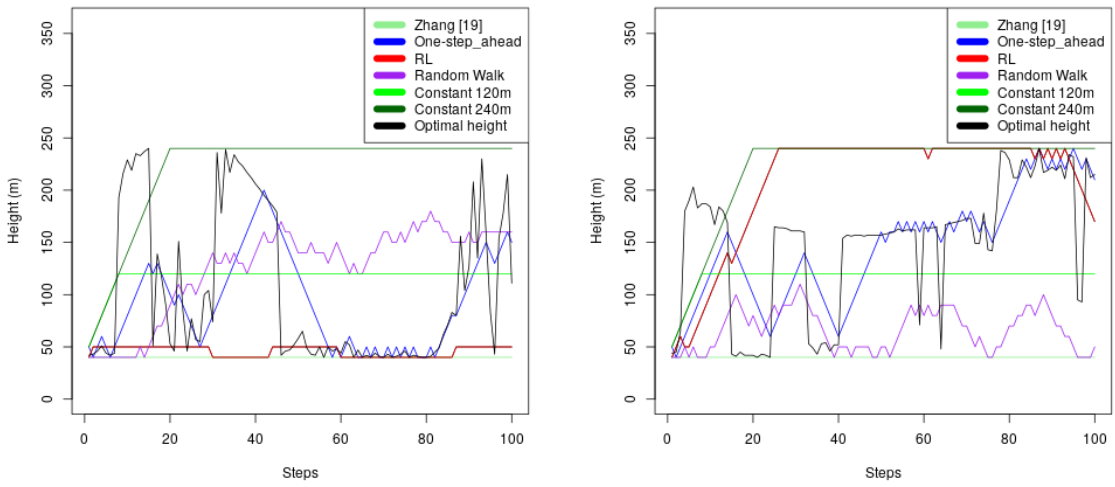


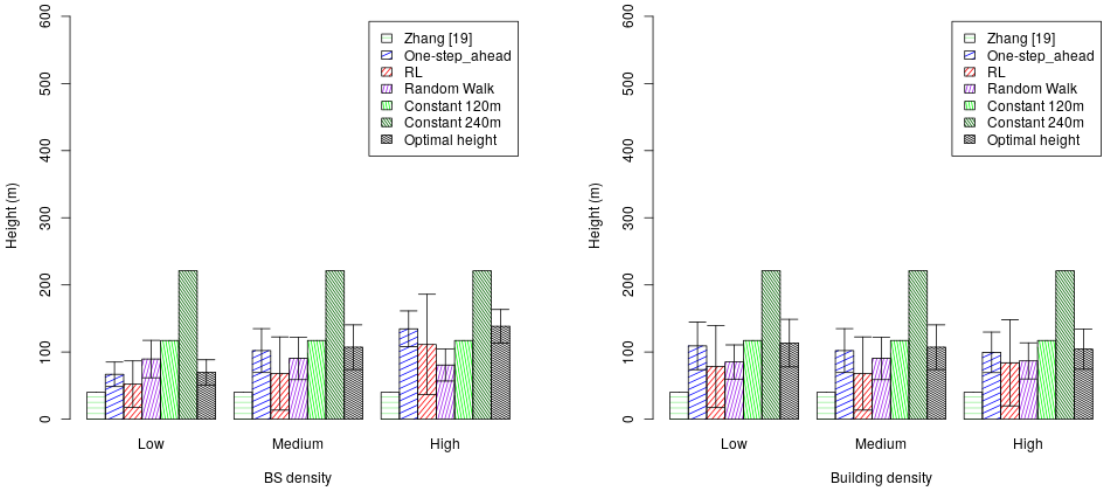
FIGURE 5.5: Spectrum efficiency per unit of bandwidth (bits/Hz) per building density (low, medium, and high), and medium BS density.



(A) Simulation example one.

(B) Simulation example two.

FIGURE 5.6: Height adaptations for each approach during two sample runs with medium BS and building densities.



(A) Mean height per BS density. (B) Height per building density.

FIGURE 5.7: Height analyses for varying BS and building densities.

Chapter 6

Radio Access Technology Characterisation Through Object Detection

Nowadays UAVs implement different RATs for their communication and utilise different bandwidths. Therefore, it can use WiFi or Lora, for example, for its connectivity [2]. With its inclusion as a UE of the mobile network, UAVs will also be able to use LTE and 5G RATs in both licensed and unlicensed spectrum. The utilisation of the unlicensed spectrum for cellular mobile technologies is a new field and needs further investigation [28]. The connected UAV will have this extra challenge to consider in order to improve it's QoS, as the possibility of using more channels increase the efficiency in bits/joule and bits/second throughput [31]. Studies have proven that the network can make a more efficient use of the spectrum when it has access to spectrum characterisation. [113] showed the importance of this awareness for the higher layers stacks.

In this chapter, we propose C3, as an answer to RQ3 "How to better characterise the unlicensed spectrum in order to enable more efficient spectrum usage?". Our proposed approach is an image-based ML technique applied to the scenario of RAT classification in shared spectrum. Our solution employs DNN, performing object detection directly on spectrograms, for the characterisation of different RATs in shared spectrum. We not only classify different RATs but also localise the signals in the frequency and time domains, as well as extract features, including centre frequency, bandwidth, frame and inter-frame duration. Our model works well under different levels of received signal strength, and in the presence of overlapping transmissions by multiple radios. We evaluate the proposed method under real-world collected data and through a synthetic transmitted data over the air.

The remainder of this chapter is structured as follows. In section 6.1 we specify the problem statement. In Section 6.2, we introduce our approach for the characterisation of RATs, composed of a RAT classifier and feature extraction components. In Section 6.3, we present the dataset generator used in the evaluation. In Section 6.4, we evaluate the performance of our classifier and feature extraction under different channel and interference conditions. Finally, in Section 6.5, we present our concluding remarks of this chapter.

6.1 Problem Statement

Spectrum monitoring is necessary for efficient coexistence of UEs in the shared spectrum as presented in Chapter 3. It is also needed for regulators to be able to enforce spectrum policy and identify possible violations [83]. 5G brings additional challenges in monitoring

the spectrum, such as the need to support mission-critical UAV, IoT applications, Industry 4.0, and autonomous vehicles. These different types of UEs have different needs and different requirements from their communications. To support them all, an entity that is monitoring the spectrum can take advantage of not only determine the presence of a UE in a spectrum band, but the type of UE it is, and how they use the spectrum.

Most of the existing works on spectrum monitoring employ the RSSI or energy detection-based methods for detecting the presence of a signal in the channel of interest [83], [114] [84]. However, these approaches may not be effective when multiple RATs coexist in the same band [85]. In such scenarios, the UEs must be capable of discriminating among different UEs and RATs, which can only be achieved through more advanced signal classification algorithms. We extensively investigated state of the art on the literature review in Section 3.3.3, and with Table 3.2, we conclude that still there is a gap whether to provide both, the classification of the transmitted RATs with its transmission characteristics. In order to operate efficiently, i.e., in an interference-aware manner, wireless devices operating in shared spectrum must identify other radios and RATs present in the same band before communicating.

The coexistence between different RATs is one of the challenges in this scenario, as each RAT has its own particularities and may not be suitable for coexistence in the same channel. This problem becomes more challenging when considering the broad range of services and applications envisioned in futures networks [115]. LTE-U must rely on contextual information about the spectrum usage to operate in shared spectrum [86], [116]. These solutions require an effective sensing mechanism, for providing detailed spectrum

usage information in real-time. With more information of the spectrum, it is possible to characterise it and open the possibility of detecting not only the presence of other radios in its surroundings, but also other features.

In this chapter, we propose a solution to detect the presence of other RATs in the band of interest, classify them, and extract features of the transmission, such as centre frequency, bandwidth, frame interval, and frame duration. Our main objective is to classify the RAT transmissions (R) in the spectrum, and to detect features as centre frequency, bandwidth, duty cycle, and frame duration. Examples of possible RATs $r_1, r_2, ..r_n$ could be LTE, Lora, Zigbee, DVB-T, and WiFi. Given a set of measurements M, and a set of RATs R, our objective is to classify each transmissions in the correct category r_c on each m. In addition, for each transmission, we want to provide detailed and accurate information of each feature, specifically: as centre frequency f_c, bandwidth b_w, frame interval FI, and frame duration FD. We use F as a set of features f_c, b_w, FI, FD, then the equation is in f, where $f \in F$. For each of these features, we need to minimise the distance between the estimated features F_e and its true values F_t. We formulate the problem as follows:

$$\begin{aligned} \sum_{t=1}^{\infty} \min \quad & F_t(m_t) - F_e(m_t); \\ \text{where } & F \in f_c, b_w, FI, FD \end{aligned} \tag{6.1}$$

We assume that the UE needs to have a radio capable of doing spectrum sensing in the band it is interested to operate (as described in Section 2.1 normal UEs should be able to do the sensing). The algorithm can run either in the BS or on the UE. In case the UE is restricted on its processing, the proposed approach should run in the BS. The UE can

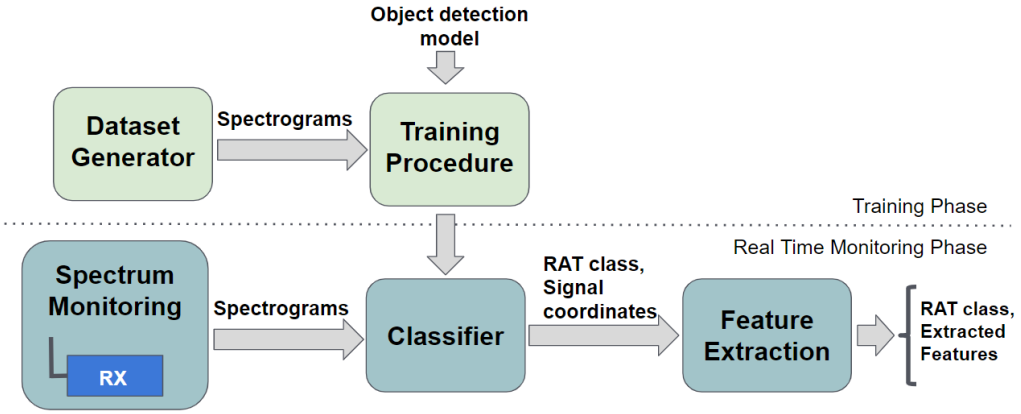


FIGURE 6.1: Overview of the proposed approach for characterising RATs through object detection.

send the sensing measurements to the BS and the BS makes the analyse. The entity that runs the algorithm needs to be capable of generate spectograms and run the algorithm.

6.2 RAT Characterisation

In this section, we describe our solution for RAT characterisation using object detection. The proposed solution is made available to the community in a public GitHub repository[1]. Our approach consists of two main components, an image-based RAT classifier, and post-processing feature extraction, as shown in Figure 6.1 in the real-time monitoring phase. In the following subsections, we describe each of these.

[1]https://github.com/Erikagpf/gr-specmonitor/tree/next

6.2.1 Image-based RAT Classifier

We proposed the desing of a CNN-based classifier for recognising different RATs coexisting in shared spectrum. Our classifier can identify multiple RATs by directly applying OD to spectrograms. To reduce this training time and the need for a big training dataset, we use TL. We use a pre-trained model and re-train its last layer. To meet the requirement to perform classification in real-time, we train the model previously to its use. Even though the training is not in real-time, using a smaller training dataset is advantageous as the labelling is made manually.

We also require a solution that can provide not just the classification of the object, but also its localisation in the image as we use spectograms (as discussed later, we rely on this localisation information for feature extraction). A spectogram is a image that represents the spectrum in frequency (horizontally) and time (vertically).

We employ the well-known OD model You Only Look Once (YOLO) [117] as the starting point for our RAT classifier. YOLO is one of the most efficient solutions in the literature for real-time implementation of OD. This model outputs both the class of the detected objects, as well as their position in the input image. Using weights and architecture from YOLO pre-trained on ImageNet [50], we modify the output layer, which corresponds to the last layer before the output of the model as illustrated in 6.2. During the training process, the output layer is explicitly optimised for the classification of LTE and WiFi waveforms. The architecture we adopted is presented in [118] and it has 19 convolution layers and 5 max-pooling layers. Moreover, our model can easily be extended for supporting more RATs, by retraining it with datasets that include new waveforms.

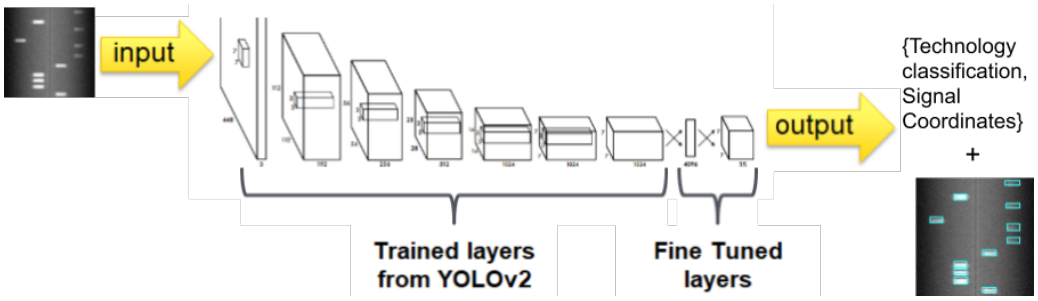
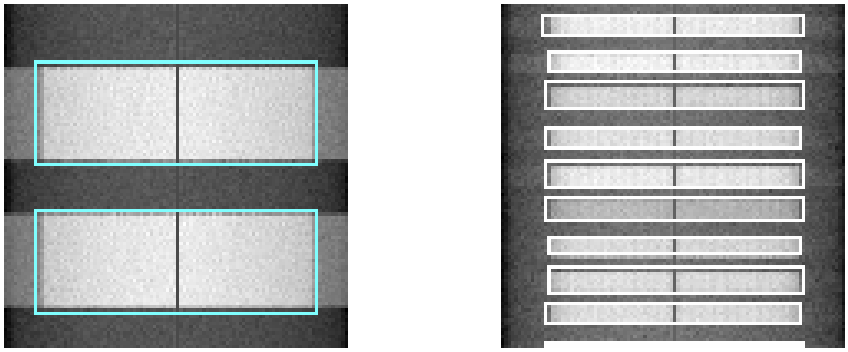


FIGURE 6.2: Proposed approach of RATs classification and detection, applying TL in YOLO2 trained model.

Our model produces the identification of the RAT (i.e., the result of the classification) and the coordinates of each frame detected in the spectrogram image. Figure 6.3 shows examples of LTE and WiFi frames detected, surrounded by bounding boxes: blue for LTE, white for WiFi. The four coordinates of each of these bounding boxes are used by the feature extraction component, discussed next.

The use of a CNN provides real-time capabilities for RAT classification. Usually, the training and testing phases are computationally expensive and take hours or days to be finished, as a vast number of input samples are needed to train and test a model. Fortunately, most of the computational power is required to train the model, but not to use it. For the spectrum sharing scenario, where it is necessary to dynamically assess how the spectrum is being occupied, we need a model that can provide acceptable classification accuracy in real-time. Once our model is trained and validated, it can provide results on the fly, making it suitable for real time applications. Our classifier analyses frames in batches of three frames each, providing three outputs at the same time; this allows us to parallelise the classification task and use multiple cores simultaneously. The batch approach allows



(A) LTE detection. (B) WiFi detection.

FIGURE 6.3: Spectrogram with the bounding boxes created by our ML-based signal classifier. The positions of the bounding boxes represent the detection of the frame, and the colour represents the classification, blue for LTE and white for WiFi.

the use of multiple cores in parallel.

A trade-off that is important to consider is the implication of this design choice on real-time detection and RAT classification: the number of images analysed simultaneously cannot be too large, as otherwise, the model will not operate in real-time. In our implementation, we evaluated the classification speed using an computer with Intel Core i7-6820 HK processor and GeForce GTX 1070 Mobile. With this commercial off-the-shelf Graphics Processing Unit (GPU), we are able to analyse three images in around 0.1 ms with 2 classes and trained with a commercial transmission dataset. We used the library Caffe[2] and the code was developed in C++.

[2] caffe.berkeleyvision.org/

6.2.2 Post-processing Feature Extraction

In this subsection, we detail the post-processing algorithm that we apply on spectrograms for extracting features of different RATs. Once the classification of the RAT is completed, it allows us to obtain additional information about the RATs present in a given channel. We consider a scenario where multiple RATs coexist in the same band. The UAV or other UE senses the channel and generates the spectrograms that represent the inputs to our model. Then, we perform the RAT classification and return the types and locations of the objects, i.e., the different RATs, present in spectrogram images. For the location of the objects on the spectograms to have a meaning, we translate these to time and frequency domains. This requires a mapping between the image parameters and time and frequency values. First, we extract the knowledge that the monitoring radio possesses about the centre frequency (F_C) and bandwidth (B_W) that it is monitoring. The spectrogram corresponds to a band of frequencies [f_1, f_2], collected during a time interval [t_1, t_2]. Let B_W represent the bandwidth of the channel being monitored. Then, we calculate the granularity that each pixel in the image represents in the time and frequency domains, as an increment value in time (I_T) and frequency (I_F), respectively. This mapping depends on the size of the spectrogram ($[X_{min}, X_{max}], [Y_{min}, Y_{max}]$)[3].

The trained model provides the corners of a rectangle that encloses a transmission frame, denoted by coordinates $x_{min}, x_{max}, y_{min}, y_{max}$. Given the coordinates of this rectangle, i.e., the bounding box, as well as the values of each time and frequency increment,

[3]Note that uppercase X and Y refer to the spectrogram, and lowercase x and y refer to the bounding box around a frame.

TABLE 6.1: The mapping between the image position and the parameters of interest in time and frequency domains.

Parameters Time/Frequency	Position Mapping
I_t	$(t_2 - t_1)/(Y_{max} - Y_{min})$
I_f	$(f_2 - f_1)/(X_{max} - X_{min})$
b_w	$(x_{max} - x_{min}) * I_f$
f_c	$f_1 + (I_f * x_{min}) + (b_w/2)$
FD	$(y_{max} - y_{min}) * I_t$
CWT	$(t_2 - t_1) - (frame\_rate * f_{av})$
FI	$CWT/frame\_rate$

we can localise the signals in the spectrum and in time. In order to calculate the bandwidth of the signal (b_w) and its centre frequency (f_c), we use the horizontal coordinates of the corners of the bounding box, translating them into their respective value in frequency. The frame duration FD of the signal is calculated in a similar manner, but now using the vertical coordinates of the corners of the bounding box. To calculate the average frame interval FI, we must first calculate the average time the channel stays without a transmission (CWT), which is the total time represented in a spectrogram subtracted by the time that is occupied by frame transmissions. Then, the frame interval FI is given by CWT divided by the number of transmissions on the spectrogram. We summarise the formulas we use for extracting the features of different RATs in Table 6.1, and illustrate the representation of the relevant values on a spectrogram in Figure 6.4.

After the feature extraction, the complete output for the RAT characterisation are the classes of the identified RATs, as well as their features, as Figure 6.4 illustrates.

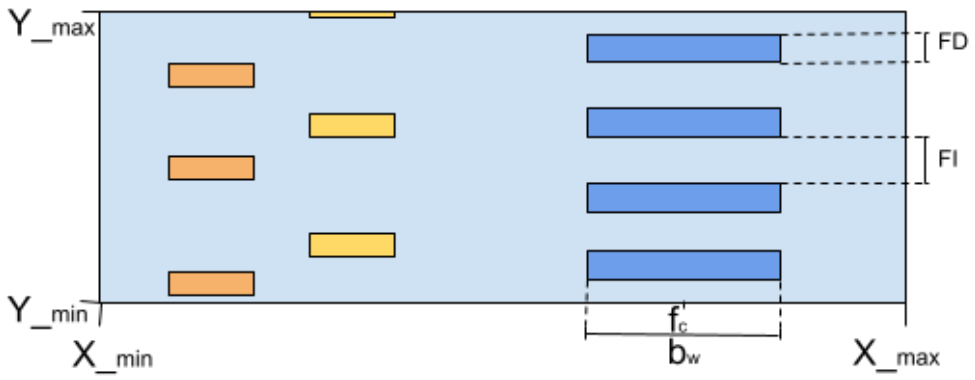


FIGURE 6.4: Parameters representation in a spectrogram.

6.3 Dataset

In order to evaluate the proposed spectrum characterisation approach, we use two different datasets. One was collected in Belgium from real-world operators transmissions, and one was generated and captured by us through over-the-air transmissions.

We need the real-world dataset in order to assess if our model can be used with real-world applications and transmissions. We needed to generate the synthetic dataset to know the exact label of each object and evaluate how good the feature extraction performs. With the synthetic data, we could also perform overlapping transmissions evaluations that would not happen in licensed spectrum transmissions and not overloaded unlicensed bands.

6.3.1 Real-world Dataset

The real-world dataset is available to the community in a GitHub repository[4] and is compound of RSSI measurement of WiFi and LTE collected at various locations of Ghent, Belgium. All the measurements are conducted in an office building of 12x80m. The measurements were collected on different days and different locations inside the building in order to increase the diversity of SINR. Ten traces were collected by each RAT. Specifically, the WiFi signals were collected in an office environment with two access points at 5540 MHz and with an average of 20 associated UEs. The LTE signal was collected from a near BS that operates in Frequency Division Duplexing (FDD) mode at 806 MHz, around the Ghent area of Belgium. A spectrum analyser of model Anritsu MS 2690A is used during the collection of the RSSI. The samples are collected at a 10 MHz of rate for 1 second. The RSSI is calculate as in equation 6.2 for N=200.

$$RSSI = 10 * log_{10}(\frac{1}{N}\sum_{k=1}^{N}(I_k^2 + Q_k^2)) \tag{6.2}$$

where N is the number of IQ samples per RSSI, and IQ the index of IQ samples.

6.3.2 Synthetic Dataset

For the testing and validation of our proposed solution, we have relied on datasets of LTE and WiFi transmissions collected over different locations in Belgium [119]. However, such datasets of commercial transmissions are not sufficient for the complete evaluation of our feature extraction component. To evaluate that component it is necessary to have

[4]https://github.com/ewine-project/Technology-classification-dataset

the ground truth for the parameters of the transmissions (the label of each spectogram), as this evaluation is related to the position of the signal in the spectrum and in time. In the case of commercial transmissions captured over the air, it is not possible to determine precisely the ground truth, and also it is not possible to vary the Signal-to-Noise Ratio (SNR) of the transmission, its centre frequency or its bandwidth, for example. To tackle this issue, in this thesis and the associated publication [100] we use the dataset generator developed with our collaborator Francisco Paisana that is available to the community in a public GitHub repository[5] [120]. The framework creates labelled RF datasets, based on waveform that mimic the transmissions of LTE and WiFi radios.

As one is transmitting and receiving the signals, they have full control and knowledge about parameters a priori so that they can generate the ground truth label of the transmissions. This allows the evaluation of the feature extraction that is essential for validating our solution. The generator relied on SRS LTE [121] for the generation of LTE signals and on a GNURadio implementation [122] for the generation of WiFi signals.

To create a dataset that reflects real-world transmissions, the dataset must be collected over-the-air to produce samples that undergo RF impairments such as phase/frequency offsets, phase noise and amplifier nonlinearities. The dataset generator uses a Software-defined Radio (SDR) to generate waveforms of different signal strength and bandwidth. It automates the collection and labelling of over-the-air samples of the waveforms of different RATs. Figure 6.5 depicts the process of generating, collecting and labelling RF waveforms using the dataset generator. These RF waveform datasets can be used for training and testing of deep CNNs for signal classification and spectrum monitoring.

[5] github.com/frankist/gr-specmonitor

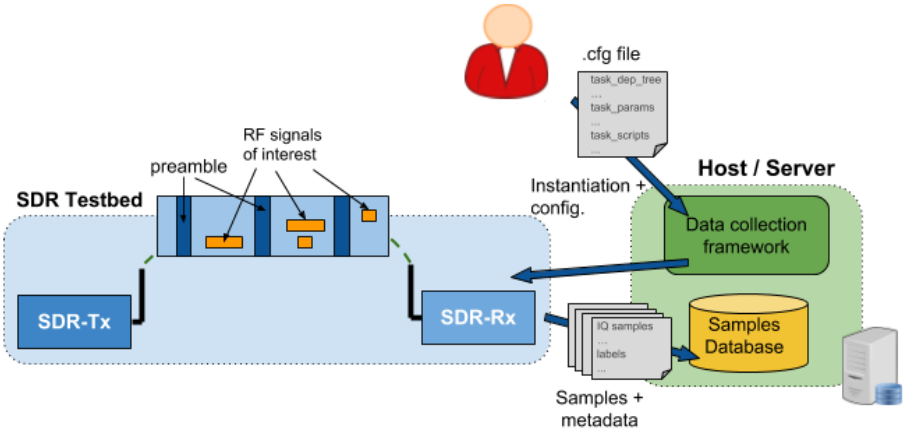


FIGURE 6.5: Generating, collecting, and labelling RAT transmissions using our dataset generator. The experimenter specifies the waveforms and their parameters. Then, our RF dataset generator creates signal traces with all the permutations of parameters, as well as transmitting, collecting, and labelling the signal traces.

It uses a pipeline-based approach for generating traces of RF waveforms with different characteristics. The process is implemented as a graph of individual tasks, e.g., producing a waveform, setting the frame duration, and setting the transmission gain. Each task can be configured and run independently. Each of the task's parameters can be a list of different values, and the task generates respective output files for all the input values. The subsequent task receives a set of different input files from the previous task and performs its operation on all of them. An example of the file used in the pipe-line described is in Appendix 8.1. Such a pipeline-based approach facilitates the extension and inclusion of new tasks, the parallelism of tasks, and resuming from intermediate points.

6.4 Performance Evaluation

In this section, we evaluate our proposed solution for RAT characterisation through OD. In order to evaluate the model accuracy with different SINRs of transmission, the feature extraction, and interference conditions, we need to use the synthetic generated dataset as we have control of the characteristics and have the ground truth of each transmission. We use the real-world dataset to test and confirm that the model is capable of working on real-world transmissions, even when the environment is not controlled.

In the evaluations, we first evaluate the detection and classification performance of our model for different RF waveform under different channel conditions with the synthetic dataset. Next, we assess the feature extraction component of our solution also with the synthetic dataset. Then, we estimate the accuracy of the RAT classifier component using real-world dataset and compare the proposed approach with different classification benchmarks. Finally, we evaluate the mAP of our solution under both datasets.

6.4.1 Detection and Classification Performance

In this section, we evaluate the detection performance and classification accuracy of our model, and demonstrate its robustness in detecting and classifying RF waveforms under different SNR conditions and interference levels. We used the dataset generator described in the previous section to compose a dataset of images, i.e., spectrograms and labels, of two radio access technology classes, LTE and WiFi. This scenario resembles the real-world use cases of coexistence in unlicensed spectrum [123].

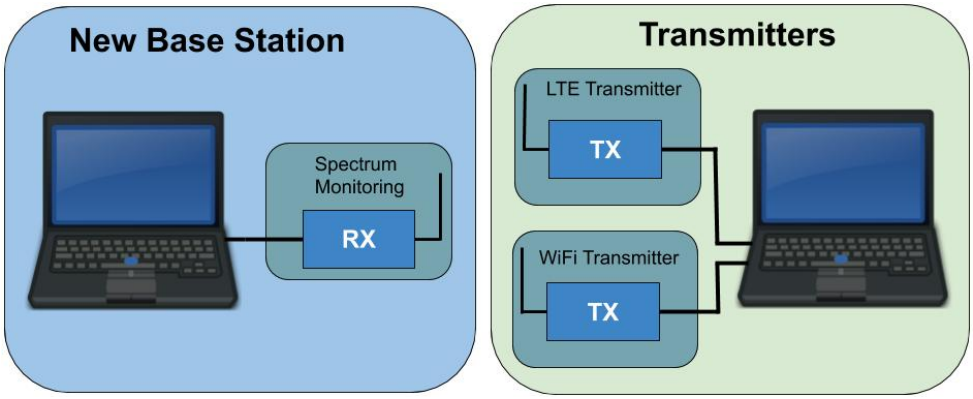


FIGURE 6.6: Experimental setup with three Ettus USRP B210s.

To generate the dataset, we emulate a new device accessing the environment, where LTE and WiFi coexist in shared spectrum, as illustrated in Figure 6.6. The setup consists of one USRP monitoring the spectrum, while a second USRP plays the role of an LTE transmitter in that band and a third USRP acts as a WiFi transmitter. Moreover, our model can be extended, for instance, by increasing the diversity of the RATs included in the training dataset. Extending the training dataset might be useful in a scenario where a technology operating in the unlicensed spectrum might share it with Bluetooth or Zigbee, for example.

Performance of the Classifier Under Different SNRs

In this analysis, we evaluate the classification performance of our model under different SNR conditions. For this evaluation, we generated a dataset with different levels of transmission power, measuring the SNR at the receiver side. We used 400 images to train the model. The dataset generator has a minimum SNR threshold value for synchronisation of

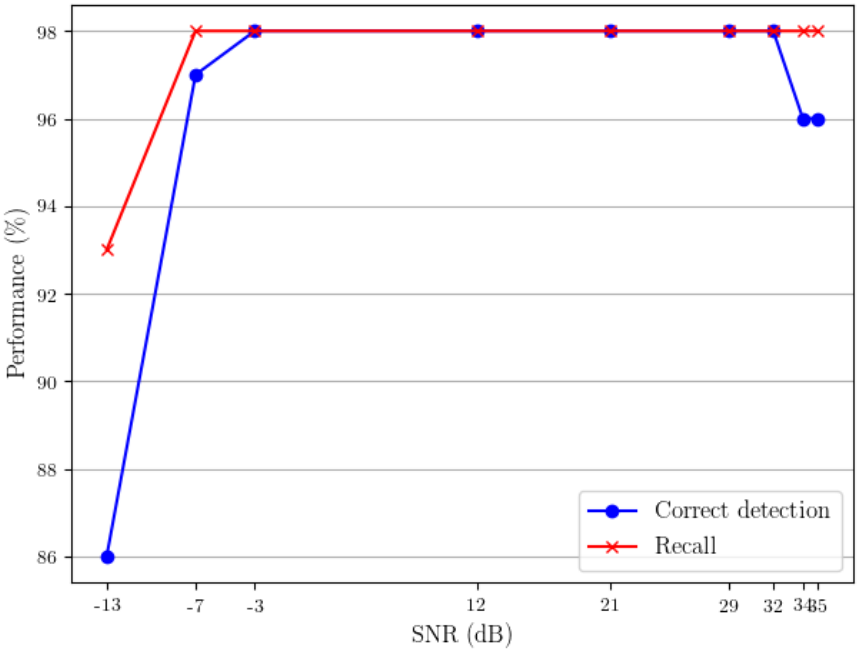
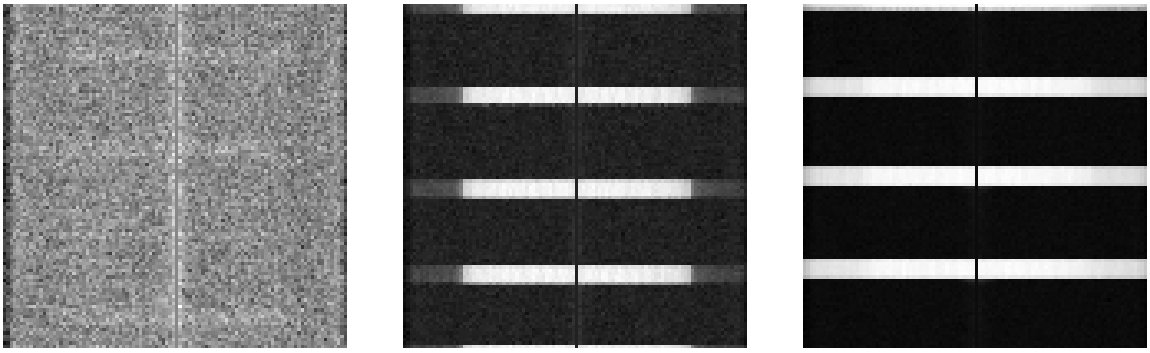


FIGURE 6.7: Percentage of detected objects as a function of SNR.

the preamble over-the-air. The measurements start with an SNR value of -13 dB and go up to 35 dB (these were the values possible to use on the dataset generator).

First, we are interested in assessing the ability of our model to detect the transmitted frames correctly. Figure 6.7 shows the percentage of correctly detected frames, as a function of SNR. The curve marked *correct detection* does not achieve 100% because there are a number of objects that were detected by the ML model but do not correspond to a transmission. The difference in the behaviour in correct detection between the -13 dB SNR and other SNR values is due to our model requiring a minimum signal power in the spectrogram. For example, with SNRs lower than -3 dB the edges of the transmitted frames



(A) WiFi detection under -13dB.

(B) WiFi detection under 12dB.

(C) WiFi detection under 35dB.

FIGURE 6.8: Illustration of WiFi signals under different SNRs.

are not as sharp as in the transmissions produced with higher SNRs, as illustrated in Figure 6.8. Once the frame transmissions have a minimum clarity in the edges, our model achieves 98% of correct detection, with a small percent of incorrect detection (between 2 and 4%).

Next, we are interested in assessing our model's ability to classify the detected frames, using recall as a metric. As introduced in Chapter 2, the recall metric illustrates the percentage of detection that was actually a transmission. The recall of our model for the detected signals is shown in Figure 6.7. The recall varies from 86% for - 13 dB of SNR in the reception, to 98% between -3 to 32 dB. For the highest SNRs, 34 and 35 dBs, we obtained an recall of 96%. It is worth mentioning that when the SNR is very high the leakage of the transmission also increases, which in our evaluation compromised 2% of detection and classification accuracy.

Interfering Transmissions with Varying Degrees of Overlap

In this analysis, we evaluate the ability of our model to detect and classify objects under the effect of cross-technology interference. As we could not find in the literature any work that evaluated their classification approach under interference, we could we do not compare in this evaluation with any benchmark. We consider two signals with the same bandwidth: the desired signal with an SNR of 34 dB, and an interfering signal with an SNR of 29 dB. At this evaluation, the desired signal is an LTE transmission, and the interfering signal is a WiFi transmission, characterising a cross-technology interference. We start this experiment by transmitting the desired and interfering signals adjacent in frequency to each other, then gradually increase the amount of overlap between the two. The model used to test the detection and classification performance of our solution under interfering transmissions with varying degrees of overlap was the one trained with the different SNRs, used in the previous section.

Figure 6.9 shows the results of our experiment, which indicate, as expected, that the detection performance suffers as the percentage of overlap between the two signals increases. In the absence of overlapping interference, i.e., when the transmissions are adjacent, our model achieves 98% of correct detection and 2% of incorrect detection. In the case of 100% of overlap, i.e., both signals are on the same channel, our model's performance drops to 92% of correct detection, while the incorrect detection increases to 7%.

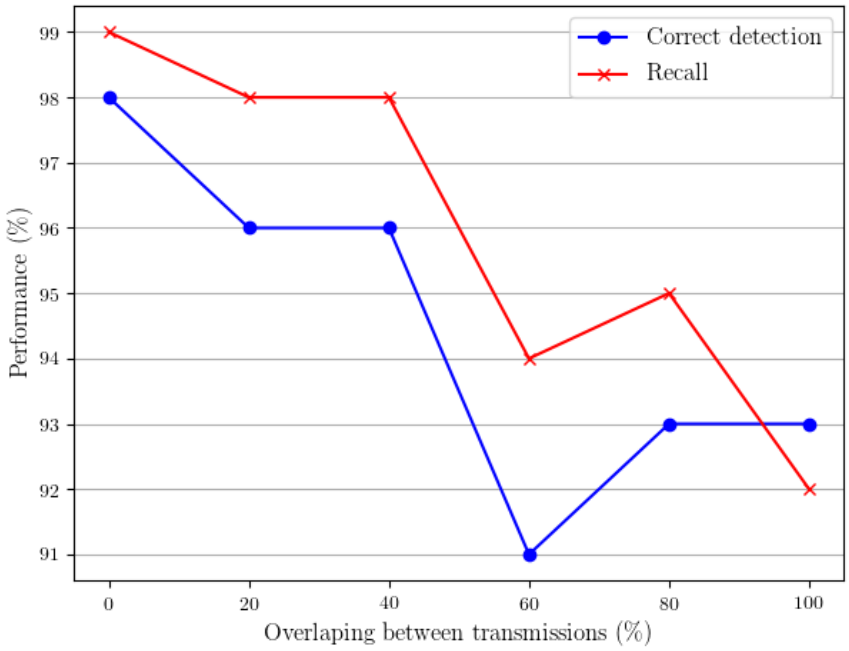


FIGURE 6.9: Correct and incorrect OD as a function of the amount of overlap between two signals in the same band.

6.4.2 Feature Extraction

To evaluate the capabilities of our feature extraction component, we generated several datasets using different combinations of: transmission bandwidths, frame duration, inter-frame duration, and centre frequency. The average SNR of the transmissions in this evaluation is 29 dB. Figure 6.10 illustrates the deviation of the features for different values of transmission characteristics. In our experiments, the value of the I_f is 192.307 KHz, which means that each pixel in the spectrograms accounts for a variation of 192.307 KHz in the frequency domain. For example, if the calculated centre frequency is off by a single pixel,

the computed value will deviate 192.307 KHz from the correct centre frequency. The same applies in the time domain, where each pixel accounts for a variation of $I_t = 28.8\mu$s.

Figure 6.10a illustrates how the calculated bandwidths vary from the ground truth regarding the image size (in percentage). The median (shown by the horizontal line in the middle of the box-plot) varies at most by 2% of the image size. In the evaluation of centre frequency, the median deviation varies at most by 1% of the image size, as shown in Figure 6.10b.

In the time domain, the median values of the frame duration deviate maximum of four percent in the worst case. Figure 6.10c illustrates that when the frame duration to be detected is smaller, the solution tends to have an average error higher than when the frame has a longer duration. It happens because it is harder to identify the precise size of smaller objects. This means that the estimated frame transmission time deviates only 144 μs compared to the ground truth. As depicted in Figure 6.10d, the outcome is that in average the median of the interframe measurements are similar to the frame duration. This is due to the fact that if our model has a high detection percentage of the transmitted frames. In case of the model fail to detect frames, it would understand that the spectrum is empty for that period, increasing the overall inter-frame duration. However, even in the occurrence of misclassifications, our model achieves a median deviation less than 2 percent in all the cases.

Considering the results discussed in this subsection, we can conclude that our model is capable of detecting the signals with high precision. Moreover, if necessary for specific applications, a higher precision can be achieved by using higher-resolution spectrograms,

i.e., smaller I_f and I_t values.

6.4.3 Performance Comparison Using Real-world Dataset

In this section, we evaluate our model using a publicly available dataset of real-world LTE and WiFi transmissions collected in Belgium. This evaluation is crucial because it shows that our model can work with commercial data, and can be used in real-world applications.

First, we investigate the accuracy of our model per number of spectrograms in the training dataset. Then, to demonstrate the ability of our OD model to classify commercial transmissions accurately, we compare our solution to the ones proposed in [97], which used the same publicly available dataset.

Training Dataset Size versus Accuracy

In this subsection, we analyse how the number of the samples (spectrogram images) affects the performance of the proposed model. The number of training data can limit the ML application because usually it needs a considerable amount of data to learn. For example, the work of [97] used more than 12 thousand images for training the CNN solution based on spectrograms. In this section, we assess the difference in the performance of our model, considering the number of training data in the training process.

We repeated the training in an identical setup while only adjusting the number of spectrograms used, i.e. 2, 10, 20, 30, 40, 50, 100, 200, and 400. The training samples equally represent the LTE and WiFi classes. Figure 6.11 illustrates how accuracy varies

by increasing the number of spectrograms. By observing Figure 6.11, we can estimate the number of labelled spectrograms required in training to achieve a certain accuracy. The best accuracy achieved was 96% with 400 spectrograms. Hence, we limited the size of our training dataset to 400 images, as it was enough for our model to achieve a comparable accuracy to the CNN image-based presented in [97], while using a considerably lower number of training images (only 3.23% of the dataset size used in [97]).

Comparison with Other ML Techniques

The public dataset is not labelled for OD, and consequently, we needed manually create bounding boxes using a tool [124] for using this data to train our model. We labelled 600 images, out of which 300 were from LTE transmissions, and were 300 from WiFi transmissions, the images from both RATs were randomly chosen from the complete dataset. From the total of 600 images, 200 were randomly chosen to be used for the test dataset, and the others 400 to train our model.

We compare the OD-based classification solution presented in this chapter against other RAT classification solutions in [97]. These solutions include fully connected neural network (FNN), Random Forest (RForest) [125], a CNN solution based on RSSI, a CNN solution based on IQ samples, and a CNN solution based on spectrograms. The results of this comparison are shown in Figure 6.12. The CNN-based solutions, including the solution presented in this paper, correctly identify the RAT with the accuracy above 95%. The CNNs IQ and image-based solutions marginally achieve better accuracy compared to

the proposed solution. However, our solution provides additional information regarding spectrum usage. This information can enhance the efficient use of the spectrum.

6.4.4 Evaluation of the Bonding Boxes: Real-world and Synthetic Datasets

In this section, we evaluate how precise our bounding boxes are using the mAP metrics. We evaluate the model's mAP performance when trained with the real-world dataset, and when trained with the synthetic dataset.

Figure 6.13a shows the values of APs for both classes, LTE and WiFi, calculated by the model trained with the generated dataset. Figure 6.14a illustrates the values of APs calculated by the model trained with the dataset collected in Belgium. The model created with data collected in Belgium shows worse performance than 6.13. The model used with commercial data cannot find more than 75% of all the transmitted LTE frames and no more than 92% for WiFi frames. However, it maintains the precision of the detected objects above 90% for LTE and near to 99% for WiFi. In the AP graph from the dataset generator model, Figure 6.13, our model detects more than 97% of the LTE frames and 94% of the WiFi frames with high accuracy, achieving approximately 99% detection of the objects.

The model trained with the data generated by us has better performance due to the automatic labelling, being more precise than manual label approaches. There is also the fact that the spectrograms generated by the public dataset were collected by different BSs and under different circumstances, which may have influenced the results as the same did not apply to the generated data in a controlled environment. The mAP of our model

trained with the public dataset is 83.04%, and the mAP of the same model trained with the generated data is 96.17%. To the best of our knowledge, this is the first work that evaluates the mAP of an OD model for RATs classification. It is worth mentioning that when the YOLOv2 is used on the VOC 2007 dataset [118], it achieves 78.6 mAP for images with a resolution of 544x544.

6.5 Conclusion

UAVs already make use of the unlicensed spectrum, although as an individual transmitter with WiFi, Lora, and more, as RATs. As UAVs will be a UE of the mobile network, they will need to provide more detailed measurements to assist the LTE and 5G BSs that work in the unlicensed spectrum. These technologies were not created to share the spectrum and need to be adapted to access the unlicensed bands. The mobile network makes more efficient use of the spectrum when it has access to spectrum characterisation [113].

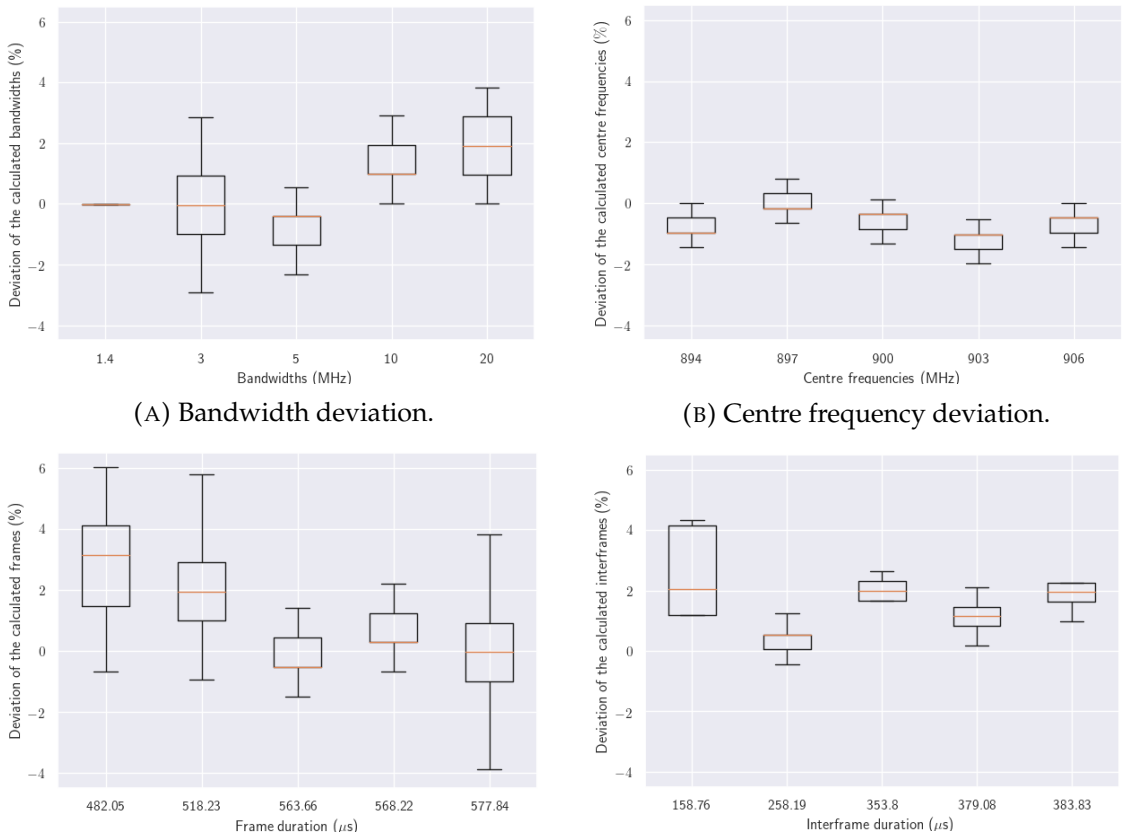
In this chapter, we presented a ML-based classifier for RAT characterisation using OD as our C3, in order to answer RQ3 "How to better characterise the unlicensed spectrum in order to enable more efficient spectrum usage?". This is the first work that evaluates the classification of an OD model in the field of technology classification. Our proposed approach combines the application of OD on spectrograms for classifying different RATs and a feature extraction component for characterising the RATs. Based on spectrogram images, we can extract specific features from the RAT, e.g., inter-frame duration, frame duration, centre frequency, and signal bandwidth. In order to evaluate the classification of our model in real-world applications, we trained and classified spectrograms based on

the public dataset [119] that was collected in different locations of Belgium. To evaluate the feature extraction component of our approach, we developed a prototype implementation of the RAT classifier using OD. We trained our classifier with LTE and WiFi waveforms and showed its efficiency in detecting and classifying different RATs through the application of OD on spectrograms. Furthermore, we evaluated the resilience of our feature extraction component through transmissions over-the-air with different bandwidths and centre frequencies under distinct SNRs and with overlapping transmissions.

Our RAT classifier using OD combined with a feature extraction algorithm can be useful in spectrum monitoring applications as connected UAVs in the unlicensed spectrum, for facilitating the characterisation of different RATs in a shared spectrum. However, there still need improvements to be done in the generation of the labelled data from commercial transmissions. For instance, the process of manually labelling data is time consuming and error-prone. Also, there is the need for further evaluation of models with different image sizes to check if independent of the size; the model always lose just 1 or 2 pixels when it creates the bounding boxes. Another further investigation can be done in the power of the interference and the transmitted signal once the transmissions are overlapping.

We have made available to the community in a public GitHub repository the labelled dataset from commercial transmissions[6]. Our implementation is based on widely used frameworks, such as GNU Radio, for digital signal processing, and YOLO, for real-time OD. This facilitates the use of our proposed approach by the community and enables further potential applications related to spectrum sensing.

[6]https://github.com/Erikagpf/WiFi-LTE-commercial-data–labelled-for-OD



(A) Bandwidth deviation.

(B) Centre frequency deviation.

(C) Frame duration deviation.

(D) Inter-frame duration deviation.

FIGURE 6.10: Feature extraction deviation evaluation in time and frequency domain.

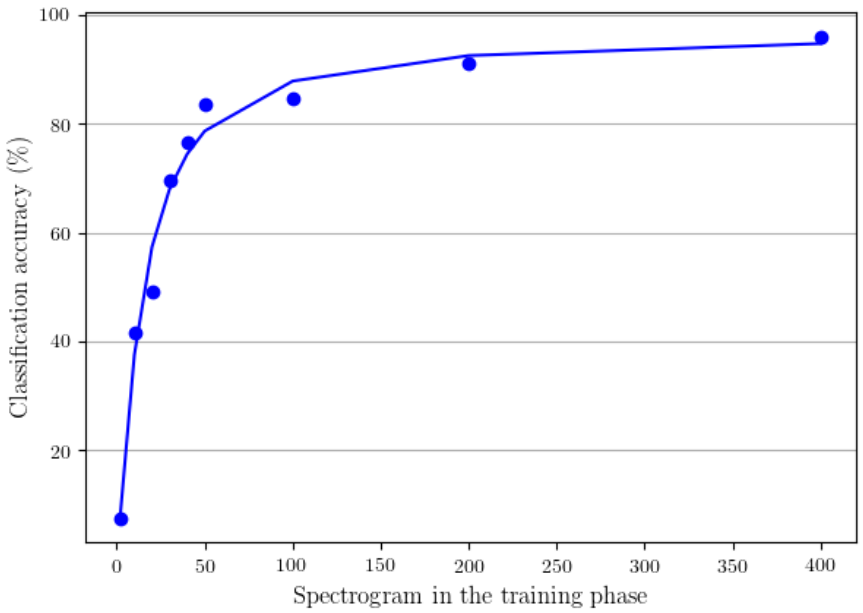


FIGURE 6.11: Number of spectrogram in the training phase versus accuracy of the model.

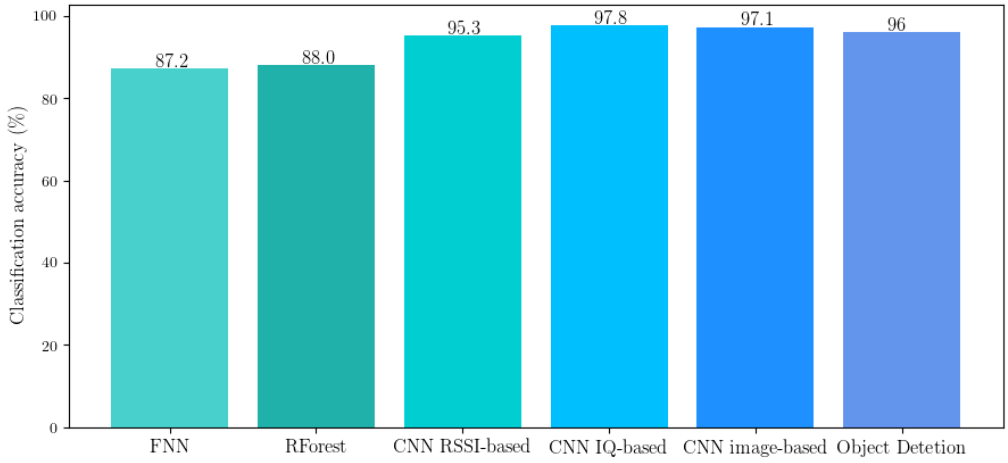
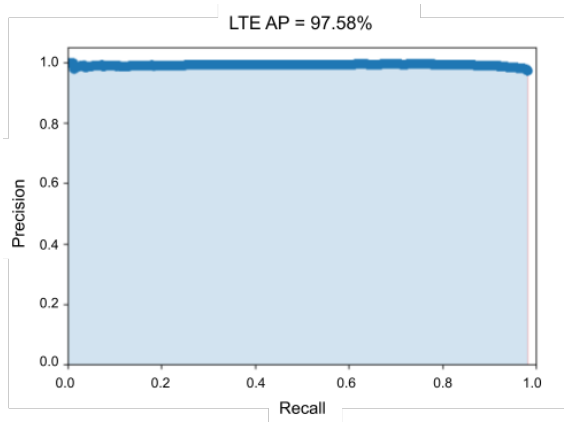
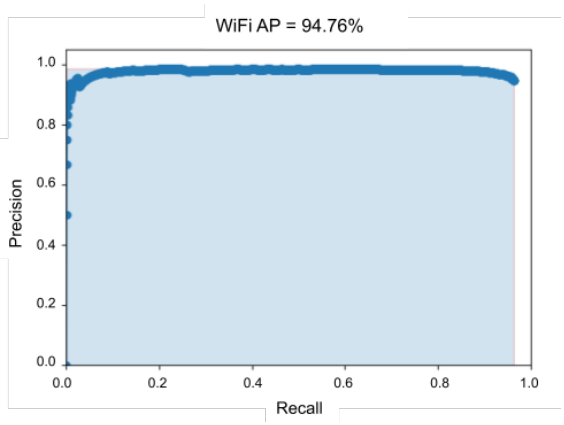


FIGURE 6.12: Classification accuracy of different ML solutions.

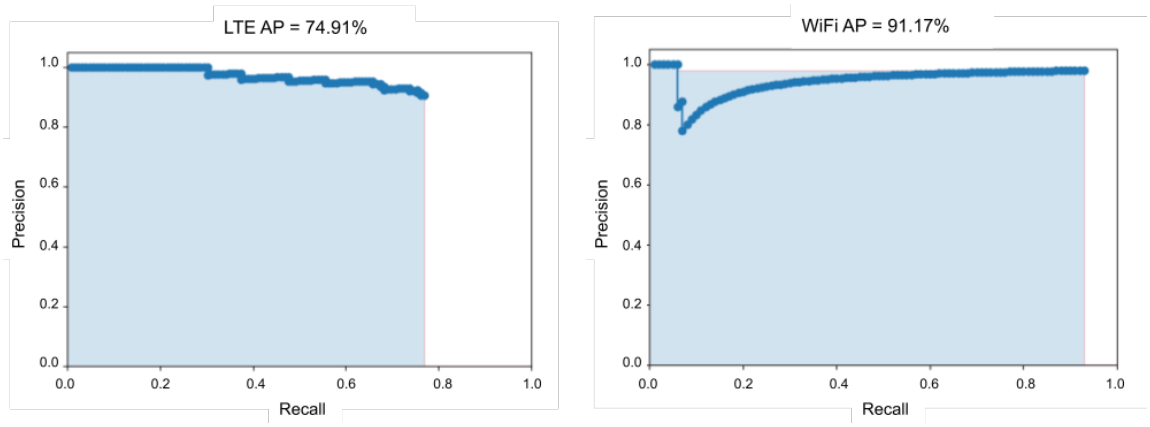


(A) LTE AP from our model trained with the generated dataset.



(B) WiFi AP from our model trained with the generated data.

FIGURE 6.13: AP of the ML-based signal classifier trained with the generated dataset.



(A) LTE AP from our model trained with the dataset collected in Belgium.

(B) WiFi AP from our model trained with the dataset collected in Belgium.

FIGURE 6.14: AP of the ML-based signal classifier trained with the generated dataset.

Chapter 7

Conclusions and Open Challenges

This thesis has investigated improving connected UAV's QoS in mobile cellular networks. This chapter summarises the contributions, discusses the tradeoffs, and introduces future works in the investigated area. Section 7.1 provides the summary of the contributions of this thesis, with a deeper discussion on tradeoffs and limitations of its applications. In Section 7.2, we present how the work could be extended to surpass its limitations. In Section 7.3, we present the future trends in the connected UAV research area.

7.1 Thesis Summary

In this thesis, we presented different ways a connected UAV can improve its QoS. Chapter 1 introduced the connected UAV, how its connectivity occurs, and its requirements. This chapter also presented the scenario of interest, which we assume in our experiment. We presented the research questions we address in this thesis, our contributions designed

to provide an answer to the research questions raised, the outline of the thesis and the publications related to the thesis.

We then introduced background concepts on UAV connectivity, 5G mobility management, UAV in the unlicensed spectrum, and machine learning concepts in Chapter 2. These concepts are introduced so the reader can follow the challenges the thesis intends to solve, the proposed solutions and state of the art in the investigated areas. First, in Section 2.1, we introduced the architecture involved in the UE mobility, responsibilities, and the events that may occur depending on the sensing of the spectrum. It was needed to understand the presented challenges in Chapter 3, and the dynamics of spectrum sensing. Then, in Section 2.2, we introduced how UAV communications arise nowadays and how the research community is looking at UAVs as being a potentially demanding UE for future networks. In this section, we also introduced the concept of cognitive radios and the steps that should be done by these radios before using the unlicensed spectrum—one of them being spectrum sensing, which develops to be our Chapter 6. We then introduced the ML techniques for UAV integration, where we present the concepts of supervised and reinforcement learning.

In Chapter 3, we addressed RQ1 "Which are the main challenges a connected UAV may encounter if deployed in a typical modern-day cellular network?". Our contribution to answer the question is presented in this Chapter, divided by air coverage, PCI challenges and handover challenges. To derive these challenges, we analysed 3GPP technical reports, state of the art on UAV communications, and even gNBs specifications to understand what could change in the case a connected UAV flies as a UE of the mobile network

that is not prepared for it. We then investigated challenges related to the air coverage, which included PCI challenge and handover challenges. They happen mainly because of high altitude, as the UAV can sense a significant number of BSs, as explained in the scenario of interest in Chapter 1. To support our claims, we used the data collected in Dublin city centre by a UAV that was carrying a mobile phone. We could observe in Section 3.2.4, that as the UAV increased its height, it started sensing further cells as the strongest one, and the most potent cell change varied over the heights, showing a behaviour different than expected for GUEs. We then inspected possible solutions on the network infrastructure that could ease the proposed challenges. However, in this thesis, we decided to investigate what the UAV could do to help with its integration into the network. We decided to go in this direction, as any change to the network could take years or not even be implemented by all the operators. In addition, UAVs are a flexible technology which are capable of leveraging their mobility and on-board intelligence to address these problems in a more efficient manner. Then, we presented the state of the art in areas where a UAV could facilitate its integration on the network. We start with UAV handover optimisation, where the UAV can support the handover decisions depending on environmental conditions. We then presented the state of the art in UAV movement optimisation in Section 3.3.2, focusing on two works that considered UAV connectivity to decide its trajectory. These works did not dynamically optimise the height of the UAV; they assumed a constant height for its experiments and focused on the horizontal path adaptation. One of the presented works [19] is used later in Chapter 4 and 5 as a benchmark to compare our proposed approach. In Section 3.3.3, we review state of the art in spectrum characterisation,

examining modulation classification and RAT classification. This area was investigated for works using different strategies. A few would calculate based on the signal characteristics of the RAT or the modulation, and the most recent ones would apply various techniques of ML. In order to use ML, most works needed an extensive amount of labelled data, however, they could achieve a reliable performance above 93%. A common factor between the approaches is that they only provided the classification and no other characteristics of the transmissions.

Chapter 4 and Chapter 5 proposed the answer to RQ2 "How to adapt the UAV height in order to increase UAV's QoS?". These chapters answer this question by presenting and evaluating a reinforcement learning-based solution to UAV height optimisation. The proposed solution adapts the UAV's height based on an RL model that optimises the height of the UAV dynamically. The main objective of the solution is to increase the QoS in the long term. We proposed a height adaptation approach independent of the horizontal approach because we considered that depending on UAV applications, it would not be able to adapt its horizontal path, for example: organ delivery or even security surveillance needs the horizontal path to be defined for its application. The independent approach is also interesting if the UAV adapts its horizontal path following other optimisations, such as the shortest path. To evaluate our proposed approach, we evaluated it in an experimental environment in Chapter 4, with data from two different locations of Dublin city centre. In one location the UAV is connected to macro-cells and in other location the UAV is connected to small cells. We defined the action as the possibilities of move up, move down, and do not move. The reward is the throughput, and the model's state space is the

actual SINR, height and the same for the last four steps, with their respective achieved reward. In order to generalise the results encountered, we also evaluated our approach in a simulation environment in Chapter 5. We varied the density of BSs and buildings for 100 Mctrials to each density and evaluate how these changes impact the solution performance. In Chapter 5, we had access to information that would be used for handover events by a standard UE, like its neighbours SINRs. As this information express' more details about the environment characteristics, we add this information to represent the environment on the state space. The other change we did on the model used in the simulation was the use of spectrum efficiency as a reward and not the direct throughput; it happened because of a limitation of underlying simulator, however it does not affect validity of results as spectrum efficiency is direct related to throughout. Either in the experimental investigation or the simulation, the RL proposed approach provides better throughput when compared to the feasible solutions. Although, we perceived that in both cases, the improvement did not pass more than 6% when compared to the benchmarks. It proves that it is possible to dynamically improve UAV's QoS adapting its height, although that improvement depends on the UAV freedom of movement and how critical is its connectivity. Maintaining the lower altitude, as proposed by Zhang [19], might be a interesting approach if the UAV has to inform its height previous to the flight.

The last contribution chapter is Chapter 6, that addressed research question RQ3, "How to better characterise the unlicensed spectrum in order to support UAV and network decisions?". We answered this question with a spectrum characterisation model, that provided detailed information of the spectrum. As a UE of the mobile network,

UAVs can keep using the unlicensed spectrum for their communications with RATs as LTE and 5G. As UAVs encounter a non-planned air coverage, it is likely to find poor coverage areas and sense further transmissions due to its LoS. Usually, UEs sense the licensed spectrum in order to find a BS that provides good QoS. Although, the sensed information is used only to perform handovers, not to decide when to transmit. RATs need to have some mechanism to share it fairly in the unlicensed spectrum, which did not happen for mobile cellular networks before. To make efficient use of the spectrum, we proposed in a RAT characterisation through OD. We proposed to apply OD on spectrograms in order to classify different RAT transmissions on a bandwidth. We base our proposed solution on a pre-trained YOLO model, which we transfer to our problem using TL techniques and fine-tune the softmax layer so it could learn the RAT classes. The use of TL was effective to train our model; we needed only 400 spectrograms when using real-world transmissions. The model itself outputs the class of the detected object and its location. Based on this information, we calculate the bandwidth of each transmission, its centre frequency, the duration of each frame, and an average interval of its transmissions, which provided a good characterisation of the spectrum. Our results showed that the classification model provided high accuracy, 96%, but not above all the CNN approaches, that goes between 95.3% and 97.8%. However, as we evaluated three frames in a row for real-time transmissions, the chance a transmission is detected if it appears in more than one frame is even higher. We also evaluate the overlapping transmissions, where two transmissions are happening at the same bandwidth. In this analysis, our model still managed to classify

the transmissions with 93% accuracy, even with 100% overlap. While evaluating the features of the transmissions, as the bandwidth, centre frequency, frame duration and frame interval, we concluded that the output of our solution would vary around 2 pixels from the exact location, making the feature extraction component depending on the size of the spectrogram.

7.2 Directions for Improvement

This section introduces possible future directions in the work presented in this thesis. We divide this section in the future trends to the proposed RL approach for dynamic height optimisation and a section for the proposed OD model for spectrum characterisation.

7.2.1 Proposed RL Approach Improvements for Dynamic Height Optimisation

There are a number of directions in which the work presented in this thesis can be further extended and evaluated. In the height optimisation RL approach, the model did not explicitly address the impact of height changes on energy usage, however it could be extended to consider the energy the UAV has available and how critical it is to improve its connection before taking a height adaptation decision. Energy-saving factor can improve our solution and help it be available to real-world applications, where energy saving is a concern to UAVs [17], [18].

As mentioned in Chapter 1, one of the advantages of having a height optimisation model separated from the horizontal optimisation is the possibility of using it with different horizontal path approaches. It would be an interesting direction to implement the proposed RL height adaptation with other concerns adapting its horizontal path. In Chapters 4 and 5, we adapted the height while the UAV was moving straight in the horizontal direction and when it was slightly adapting its horizontal route to be closer to the serving BS. It could be interesting to investigate this approach with other ways to optimise the path. For example, in [18], authors are concerned about UAVs energy re-charging to decide its horizontal path and do not consider the UAV connectivity in its implementation. Therefore, our RL model could be used with [18] proposed approach to achieve the best of both optimisations, maintaining the UAV charged and with a reliable connection.

The height adaptation approach should also be investigated in a longer experimental data-set, where the UAV fly over a long path and observes how it behaves. We also consider that using simulations to import real-world 3D maps could be an excellent strategy to test before using it in the real world.

When we investigated the UAV height adaptation, we only considered the UAV a unique UE of the network as we had no access to other flying UAVs or any data of GUEs that were connected to the network during the real-world data collection. In the simulation environment, we tried to replicate the same conditions as in the experiment, which limited the simulation to also excluding the GUEs and UAVs. There should be an investigation of the performance of the proposed RL approach when co-existing with other UAVs and GUEs.

During our investigation we optimised the sum of the total throughput through the path. There might be applications that the average throughput, in every step, should be optimised. These application would need an average throughput being the maximum possible. Even though these metrics are strongly related, it could influence the behaviour of our model. To prove if it influences or not, it is necessary for further investigation. The change in Equation 4.1 would be to instead of calculating the max, we calculate the average in the sum.

7.2.2 Proposed OD Approach for Spectrum Characterisation

The spectrum characterisation proposed approach needs to be investigated in a more complex scenario, with more RATs on the spectrum. One could also evaluate the proposed approach varying the strength of the interference signal. The main issue with the proposed method is that it requires labelled data to train the model. Even though it proved to learn with relatively small training data (400 spectrograms, 200 for each class), it is still necessary to create these 200 labelled data by hand.

An extension of the proposed spectrum characterisation could be to detect which applications or UEs are using the spectrum. Then, based on the applications or UEs types, know their transmission characteristics. For example, it would be valuable to know that the UE sharing the spectrum with the sensing UE is a sensor that transmits every 5 seconds. This next step on the characterisation could even help the sensing UE predict its future transmission success.

We have only investigated the spectrum characterisation, not how it is used in the decision making. The full integration with the decision making should be investigated for different types of UEs. For example, the type of UE that is transmitting will influence the decision of when to transmit, besides the type of transmissions that are already on the spectrum.

7.3 Future Trends in the Connected UAV Field

As presented in Chapter 3, there are open challenges for the network operators that should be investigated. Solving these issues might take a while, as it has to be implemented in all networks. However, in the long term, the operators should be able to deal with all the possible issues its UEs might bring and do not depend only on the UE's abilities to avoid it.

A presented challenge in Chapter 3 was the coverage planning on the air. It is necessary to update the tools used in planning new sites to include air coverage. How these tools deal with different heights is a big challenge that needs to be investigated. It is also essential to aerial flight tests to confirm that the new and old sites can provide reliable communication to UAVs as it is done for GUEs.

Considering the issues that unplanned coverage brings, PCI collision and confusion are some of the most dangerous problems for the network itself, as it needs to reconfigure the BSs in order to solve it, which makes the BS turning off for a while and interfering with all the BS operations. Therefore, it is crucial to find out how to avoid these issues in the sky, even when the UAV senses further BSs. It might be that detailed planning of the

values, considering not only the GUEs but also the UAVs is enough to solve this issue. However, it needs investigation in different scenarios for macro and small cells.

Another challenge that should be investigated is how to avoid ANR possible issues, as block-listing and white-listing for sites that are only sensed by the UAV. Maintaining a separated NR for UAVs would prevent these issues on the network, although there is a need for a more detailed investigation on how this implementation needs to be done and if it is worth doing.

Finally, the network usually decides mobility management for all its UEs, and the threshold used for GUEs is proven not to be optimised for UAVs due to UAVs having more frequent disconnection time and unnecessary handovers, as presented in Chapter 3. Therefore, for future networks, the threshold to start a handover event will need to be optimised for UE's type and application.

7.4 Real-world Application in Future Networks

This thesis investigated how to improve connected UAV's QoS in order to accelerate its introduction as a UE of the future mobile networks. UAV's are intended to be actual users of the network from 5G advance implementations, as presented in Release-18 package approved at the December 2021 RAN plenary meeting [126]. The 3GPP research groups are about to define and design the UAV requirements to be UEs of the network. The content presented in this thesis could be of use while defining these requirements.

The regulators should further investigate the presented challenges presented in Chapter 3 as some of them can impact all the network and generate interference. It is important

that these challenges are addressed to avoid a decrease on the network performance when the UAV is added as a UE.

The presented height adaptation should not be regulated by 3GPP, but actually be seen as a possible implementation for an operator or a UAV manufacture as an approach to improve the UAVs QOS. Before being available on the connected UAV, it needs further investigation on the energy spent for the dynamic height adaptation.

The unlicensed spectrum will be investigated by the regulators for V2X communication. They have expressed the intention to research the use of the spectrum in devices onboard aerial vehicles, including UAVs. The proposed approach for spectrum characterisation can be used as an operator approach to better use the unlicensed spectrum. It should not be regulated by 3GPP. The main adaptation needed to the proposed work to be used in real-world scenarios at the time of the 5G advance, is to train the model with a greater number of RATs in order to cover all the RATs encountered on the future unlicensed scenario. It is also important to use data collected by a UAV during training, so the model will be trained to the exact same scenario the UAV will encounter.

Chapter 8

Appendix

Appendix A

LISTING 8.1: Parameters passed to Dataset Generator

```
#!/usr/bin/env python

import numpy as np
from specmonitor.labeling_framework import random_generator

num_sections = 1
section_size = 3000000
toffset_range = [50]
frequency_offset = 0
skip_samps = 0
wf_gen_samps = section_size*num_sections + toffset_range[-1] + skip_samps + 50
n_repeats = 10
sample_rate = 23.04e6
```

```
ch1_interval = (30000,50000) #1.5-2.5ms
ch2_interval = (1000000,1500000) # 50-100ms
N_fft_avg = 150

tags = ['wifi','lte']

spectrogram_representation = {
    'format_type':'spectrogram',
    'boxlabel':'waveform',
    'fftsize':256,
    'cancel_DC_offset':True,
    'dB':True
}
Tx_params = {
    'frequency_offset': frequency_offset,
    'time_offset': toffset_range,
    'section_size': section_size,
    'num_sections': num_sections,
    'soft_gain': 1.0,
    'noise_voltage':0
}

stage_params = {
    'wifi':
    {
        'waveform':
        {
```

```
            'waveform': 'wifi',
            'number_samples': wf_gen_samps,
            'sample_rate': sample_rate,
            'encoding': 0,
            'pdu_length': 1500,
            'pad_interval': [('uniform',ch1_interval)],
            'signal_representation': [spectrogram_representation],
            'frame_mag2': 0.99,
            'runs': range(n_repeats)
        },
        'Tx': Tx_params
    },
    'lte':
    {
        'waveform':
        {
            'waveform': 'lte_dl',
            'n_samples': wf_gen_samps,
            'sample_rate': sample_rate,
            'n_prbs': 100,
            'pad_interval': random_generator('randint',ch2_interval),
            'signal_representation':[spectrogram_representation],
            'n_offset_samples': [('uniform',(0,500000))],
            'runs': range(n_repeats)
        },
        'Tx': Tx_params
    }
```

```
}
```

Bibliography

[1] *30+ Smartphone Usage Statistics for the UK [2021]*, https://cybercrew.uk/blog/smartphone-usage-statistics-uk/, Accessed: 2022-01-11.

[2] Shukla, Deepshikha, "Controlling drones and uavs: Advancements in wireless technologies," *Electronics For You*, pp. 70–71, 2018.

[3] *Da-Jiang Innovations - DJI*, https://www.dji.com/, Accessed: 2022-01-11.

[4] Stöcker, Claudia and Bennett, Rohan and Nex, Francesco and Gerke, Markus and Zevenbergen, Jaap, "Review of the current state of UAV regulations," *Remote sensing*, vol. 9, no. 5, p. 459, 2017.

[5] Galkin, Boris, "Consumer and Commercial Drones: How a technological revolution is impacting Irish society," *L&RS Spotlight*, 2021.

[6] Stanczak, Jedrzej and Kovacs, Istvan Z and Koziol, Dawid and Wigard, Jeroen and Amorim, Rafhael and Nguyen, Huan, "Mobility challenges for unmanned aerial vehicles connected to cellular LTE networks," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, IEEE, 2018.

[7] Azari, M Mahdi and Rosas, Fernando and Pollin, Sofie, "Cellular connectivity for UAVs: Network modeling, performance analysis, and design guidelines," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3366–3381, 2019.

[8] Mozaffari, Mohammad and Saad, Walid and Bennis, Mehdi and Nam, Young-Han and Debbah, Mérouane, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys and Tutorials*, 2019.

[9] Mozaffari, Mohammad and Saad, Walid and Bennis, Mehdi and Debbah, Mérouane, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, 2016.

[10] Azari, Mohammad Mahdi and Rosas, Fernando and Pollin, Sofie, "Reshaping cellular networks for the sky: Major factors and feasibility," in *2018 IEEE International Conference on Communications (ICC)*, IEEE, 2018, pp. 1–7.

[11] Azari, Mohammad Mahdi and Rosas, Fernando and Chiumento, Alessandro and Pollin, Sofie, "Coexistence of terrestrial and aerial users in cellular networks," in *2017 IEEE Globecom Workshops (GC Wkshps)*, IEEE, 2017, pp. 1–6.

[12] Galkin, Boris and Kibilda, Jacek and DaSilva, Luiz A, "Backhaul for low-altitude UAVs in urban environments," in *2018 IEEE International Conference on Communications (ICC)*, IEEE, 2018, pp. 1–6.

[13] Galkin, Boris and Amer, Ramy and Fonseca, Erika and DaSilva, Luiz A., "Intelligent Base Station Association for UAV Cellular Users: A Supervised Learning Approach," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020, 383–388.

[14] Challita, Ursula and Saad, Walid and Bettstetter, Christian, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2125–2140, 2019.

[15] Huang, Yi-Feng and Chen, Hsiao-Hwa, "Applications of Intelligent Radio Technologies in Unlicensed Cellular Networks-A Survey," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 15, no. 7, pp. 2668–2717, 2021.

[16] Cheng, Peng and Chen, Zhuo and Ding, Ming and Li, Yonghui and Vucetic, Branka and Niyato, Dusit, "Spectrum intelligent radio: Technology, development, and future trends," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 12–18, 2020.

[17] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 3, 2002, 1936–1941 vol.3.

[18] Natalizio, Enrico and Zema, Nicola Roberto and Di Puglia Pugliese, Luigi and Guerriero, Francesca, "Download and Fly: An Online Solution for the UAV 3D Trajectory Planning Problem in Smart Cities," in *Proceedings of the 9th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, 2019, pp. 49–56.

[19] S. Zhang and Y. Zeng and R. Zhang, "Cellular-Enabled UAV Communication: A Connectivity-Constrained Trajectory Optimization Perspective," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2580–2604, 2019.

[20] Saleem, Yasir and Rehmani, Mubashir Husain and Zeadally, Sherali, "Integration of cognitive radio technology with unmanned aerial vehicles: issues, opportunities, and future research challenges," *Journal of Network and Computer Applications*, vol. 50, pp. 15–31, 2015.

[21] Eappen, Geoffrey and Shankar, T and Nilavalan, Rajagopal, "Advanced squirrel algorithm-trained neural network for efficient spectrum sensing in cognitive radio-based air traffic control application," *IET Communications*, 2021.

[22] "TS 138.331: Technical Specification. 5G;NR;Radio Resource Control (RRC);Protocol specification; (3GPP TS 38.331 version 15.3.0 Release 15)," 3GPP, Tech. Rep., 2020.

[23] "Unmanned aircraft systems," Qualcomm, Tech. Rep., 2017.

[24] Bilen, Tugce and Canberk, Berk and Chowdhury, Kaushik R, "Handover management in software-defined ultra-dense 5G networks," *IEEE Network*, vol. 31, no. 4, 2017.

[25] A. Dahlen and A. Johansson and F. Gunnarsson and J. Moe and T. Rimhagen and H. Kallin, "Evaluations of LTE Automatic Neighbour Relations," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, 2011.

[26] Ramachandra, Pradeepa and Gunnarsson, Fredrik and Zetterberg, Kristina and Moosavi, Reza and Amirijoo, Mehdi and Engström, Stefan and Tidestav, Claes and Ramos, Edgar, " On automatic establishment of relations in 5G radio networks," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, 2016.

[27] "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on NR-based access to unlicensed spectrum (Release 16)," 3GPP, Tech. Rep., 2018.

[28] Dias Santana, Guilherme Marcel and Cristo, Rogers Silva de and Lucas Jaquie Castelo Branco, Kalinka Regina, "Integrating Cognitive Radio with Unmanned Aerial Vehicles: An Overview," *Sensors*, vol. 21, no. 3, p. 830, 2021.

[29] Yucek, Tevfik and Arslan, Huseyin, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[30] Qiu, Robert C and Zhang, Changchun and Hu, Zhen and Wicks, Michael C, "Towards A Large-Scale Cognitive Radio Network Testbed: Spectrum Sensing, System Architecture, and Distributed Sensing.," *Journal in Communications*, vol. 7, no. 7, pp. 552–566, 2012.

[31] Li, Liying and Zhou, Xiangwei and Xu, Hongbing and Li, Geoffrey Ye and Wang, Dandan and Soong, Anthony, "Energy-efficient transmission in cognitive radio networks," in *2010 7th IEEE consumer communications and networking conference*, IEEE, 2010, pp. 1–5.

[32] Mahesh, Batta, "Machine Learning Algorithms-A Review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[33] Hämäläinen, Seppo and Sanneck, Henning and Sartori, Cinzia, *LTE self-organising networks (SON): network management automation for operational efficiency*. John Wiley and Sons, 2012.

[34] Challita, Ursula and Ryden, Henrik and Tullberg, Hugo, "When machine learning meets wireless cellular networks: Deployment, challenges, and applications," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 12–18, 2020.

[35] Letaief, Khaled B and Chen, Wei and Shi, Yuanming and Zhang, Jun and Zhang, Ying-Jun Angela, "The roadmap to 6G: AI empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[36] Kato, Nei and Mao, Bomin and Tang, Fengxiao and Kawamoto, Yuichi and Liu, Jiajia, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 96–103, 2020.

[37] Gui, Guan and Liu, Miao and Tang, Fengxiao and Kato, Nei and Adachi, Fumiyuki, "6G: Opening new horizons for integration of comfort, security, and intelligence," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 126–132, 2020.

[38] Cui, Laizhong and Yang, Shu and Chen, Fei and Ming, Zhong and Lu, Nan and Qin, Jing, "A survey on application of machine learning for Internet of Things," *International Journal of Machine Learning and Cybernetics*, vol. 9, no. 8, pp. 1399–1417, 2018.

[39] Galkin, Boris and Fonseca, Erika and Amer, Ramy and A. DaSilva, Luiz and Dusparic, Ivana, "REQIBA: Regression and Deep Q-Learning for Intelligent UAV Cellular User to Base Station Association," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 5–20, 2022.

[40] Clemente, Diogo and Soares, Gabriela and Fernandes, Daniel and Cortesao, Rodrigo and Sebastiao, Pedro and Ferreira, Lucio S, "Traffic forecast in mobile networks: Classification system using machine learning," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, IEEE, 2019, pp. 1–5.

[41] Fadlullah, Zubair Md and Tang, Fengxiao and Mao, Bomin and Kato, Nei and Akashi, Osamu and Inoue, Takeru and Mizutani, Kimihiro, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017.

[42] Tang, Fengxiao and Mao, Bomin and Kato, Nei and Gui, Guan, "Comprehensive Survey on Machine Learning in Vehicular Network: Technology, Applications and Challenges," *IEEE Communications Surveys and Tutorials*, 2021.

[43] Schmidhuber, Jürgen, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, 85–117, 2015.

[44] LeCun, Yann and Bengio, Yoshua and Hinton, Geoffrey, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[45] Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[46] Ghosh, Anirudha and Sufian, Abu and Sultana, Farhana and Chakrabarti, Amlan and De, Debashis, "Fundamental concepts of convolutional neural network," in *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, Springer, 2020, 519–567.

[47] Géron, Aurélien, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[48] Whiteway, Matthew R and Schaffer, Evan S and Wu, Anqi and Buchanan, E Kelly and Onder, Omer F and Mishra, Neeli and Paninski, Liam, "Semi-supervised sequence modeling for improved behavioral segmentation," *bioRxiv*, 2021.

[49] Howard, Jeremy and Ruder, Sebastian, "Universal language model fine-tuning for text classification," *preprint arXiv:1801.06146*, 2018.

[50] J. Deng *et al.*, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2009.

[51] Zou, Zhengxia and Shi, Zhenwei and Guo, Yuhong and Ye, Jieping, "Object detection in 20 years: A survey," *preprint arXiv:1905.05055*, 2019.

[52] Viola, Paul and Jones, Michael and others, "Robust real-time object detection," *International journal of computer vision*, vol. 4, no. 34-47, 4, 2001.

[53] Cruz-Mota, Javier and Bogdanova, Iva and Paquier, Benoît and Bierlaire, Michel and Thiran, Jean-Philippe, "Scale invariant feature transform on the sphere: Theory and applications," *International Journal of Computer Vision*, vol. 98, no. 2, 217–241, 2012.

[54] Dalal, Navneet and Triggs, Bill, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, 886–893.

[55] *Visual Object Classes Challenge 2012 (VOC2012)*, `http://host.robots.ox.ac.uk/pascal/VOC/voc2012/`, Accessed: 2022-01-11.

[56] Sutton, Richard S and Barto, Andrew G, *Reinforcement learning: An introduction*. MIT press, 2018.

[57] Sutton, Richard S and Barto, Andrew G and Williams, Ronald J, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 19–22, 1992.

[58] Watkins, Christopher JCH and Dayan, Peter, "Q-learning," *Machine learning*, vol. 8, no. 3-4, 279–292, 1992.

[59] Schaul, Tom and Quan, John and Antonoglou, Ioannis and Silver, David, "Prioritized experience replay," *preprint arXiv:1511.05952*, 2015.

[60] Murphy, Kevin P, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[61] Masters, Dominic *et al.*, "Revisiting small batch training for deep neural networks," *preprint arXiv:1804.07612*, 2018, Accessed: 2022-01-11.

[62] 3rd Generation Partnership Project, "3GPP TR 36777: Enhanced LTE Support for Aerial Vehicles," 3rd Generation Partnership Project, Tech. Rep., 2018.

[63] ——, "3GPP TS 22.125: Remote Identification of Unmanned Aerial Systems," 3rd Generation Partnership Project, Tech. Rep., 2021.

[64] ——, "3GPP TR 23.754: Study on supporting Unmanned Aerial Systems Connectivity, Identification, and Tracking," 3rd Generation Partnership Project, Tech. Rep., 2021.

[65] ——, "3GPP TR 23.755: Study on application layer support for Unmanned Aerial System (UAS)," 3rd Generation Partnership Project, Tech. Rep., 2021.

[66] ——, "3GPP TR 22.829: Enhancement for Unmaned Aerial Vehicles (UAVs)," 3rd Generation Partnership Project, Tech. Rep., 2019.

[67] Fakhreddine, Aymen and Bettstetter, Christian and Hayat, Samira and Muzaffar, Raheeb and Emini, Driton, "Handover challenges for cellular-connected drones," in *Proceedings of the 5th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, 2019.

[68] Euler, Sebastian and Maattanen, Helka-Liina and Lin, Xingqin and Zou, Zhenhua and Bergström, Mattias and Sedin, Jonas, "Mobility support for cellular connected unmanned aerial vehicles: Performance and analysis," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, IEEE, 2019.

[69] *DenseAir Dataset Sample*, `https://github.com/galkinb/DenseAirDatasetSample`, Accessed: 2022-01-18.

[70] Amer, Ramy and Saad, Walid and Marchetti, Nicola, "Mobility in the sky: Performance and mobility analysis for cellular-connected UAVs," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3229–3246, 2020.

[71] Azari, Amin and Ghavimi, Fayezeh and Ozger, Mustafa and Jantti, Riku and Cavdar, Cicek, "Machine Learning Assisted Handover and Resource Management for Cellular Connected Drones," *IEEE Vehicular Technology Conference (VTC Spring)*, 2020.

[72] Chen, Yun and Lin, Xingqin and Khan, Talha Ahmed and Mozaffari, Mohammad, "A Deep Reinforcement Learning Approach to Efficient Drone Mobility Support," *Arxiv e-prints*, 2020.

[73] Du, Jun and Jiang, Chunxiao and Wang, Jian and Ren, Yong and Debbah, Merouane, "Machine Learning for 6G Wireless Networks: Carrying Forward Enhanced Bandwidth, Massive Access, and Ultrareliable/Low-Latency Service," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 122–134, 2020.

[74] Chowdhury, Md Moin Uddin and Saad, Walid and Guvenc, Ismail, "Mobility Management for Cellular-Connected UAVs: A Learning-Based Approach," *IEEE International Conference on Communications (ICC)*, 2020. arXiv: 2002.01546.

[75] Ibrahim, Ehab Ahmed and Badran, Ehab F, "An optimized Doppler-based LTE measurement procedure for A4 handover triggering event in high speed train networks," in *2017 8th International Conference on Information Technology (ICIT)*, IEEE, 2017, 380–387.

[76] Jaeger, Herbert, "Echo state network," *Scholarpedia*, vol. 2, no. 9, 2330, 2007.

[77] Al-Hourani, Akram and Kandeepan, Sithamparanathan and Lardner, Simon, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[78] Ravi, Vishnu Vardhan Chetlur and Dhillon, Harpreet S, "Downlink coverage probability in a finite network of unmanned aerial vehicle (UAV) base stations," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, IEEE, 2016, pp. 1–5.

[79] Chetlur, Vishnu Vardhan and Dhillon, Harpreet S, "Downlink coverage analysis for a finite 3-D wireless network of unmanned aerial vehicles," *IEEE Transactions on Communications*, vol. 65, no. 10, pp. 4543–4558, 2017.

[80] R. I. Bor-Yaliniz and A. El-Keyi and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–5.

[81] E. Kalantari and H. Yanikomeroglu and A. 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.

[82] X. Liu and Y. Liu and Y. Chen, "Reinforcement Learning in Multiple-UAV Networks: Deployment and Movement Design," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8036–8049, 2019.

[83] J. Peha, "Sharing Spectrum Through Spectrum Policy Reform and Cognitive Radio," *Proceedings of the IEEE*, 2009.

[84] X. Ying *et al.*, "Incentivizing Crowdsourcing for Radio Environment Mapping with Statistical Interpolation," in *International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, IEEE, 2015.

[85] A. Selim *et al.*, "Spectrum Monitoring for Radar Bands Using Deep Convolutional Neural Networks," in *Global Communications Conference (GLOBECOM)*, IEEE, 2017.

[86] "Review of Regulatory Requirements for Unlicensed Spectrum," 3GPP, Tech. Rep., 2014.

[87] F. Paisana *et al.*, "An Alternative Implementation of a Cyclostationary Detector," in *International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2012.

[88] B. Ramkumar, "Automatic Modulation Classification for Cognitive Radios using Cyclic Feature Detection," *IEEE Circuits and Systems Magazine*, 2009.

[89] L. Han *et al.*, "Low Complexity Automatic Modulation Classification Based on Order-Statistics," *IEEE Transactions on Wireless Communications*, 2017.

[90] Z. Wu *et al.*, "Robust Automatic Modulation Classification under Varying Noise Conditions," *IEEE Access*, vol. 5, 2017.

[91] M. Aslam *et al.*, "Automatic Modulation Classification Using Combination of Genetic Programming and KNN," *IEEE Transactions on Wireless Communications*, vol. 11, no. 8, 2012.

[92] T. O'Shea *et al.*, "Over-the-Air Deep Learning Based Radio Signal Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, 2018.

[93] K. Zhang *et al.*, "A Dictionary Learning Based Automatic Modulation Classification Method," *IEEE Access*, vol. 6, 2018.

[94] F. Paisana *et al.*, "Context-Aware Cognitive Radio Using Deep Learning," in *IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, IEEE, 2017.

[95] T. O'Shea *et al.*, "Learning robust general radio signal detection using computer vision methods," in *IEEE 51st Asilomar Conference on Signals, Systems, and Computers*, IEEE, 2017.

[96] S. Rajendran *et al.*, "Deep Learning Models for Wireless Signal Classification With Distributed Low-Cost Spectrum Sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, 2018.

[97] J. Fontaine *et al.*, "Towards Low-Complexity Wireless Technology Classification Across Multiple Environments," *Ad Hoc Networks*, 2019.

[98] Jan, Sana Ullah and Vu, Van-Hiep and Koo, Insoo, "Throughput maximization using an SVM for multi-class hypothesis-based spectrum sensing in cognitive radio," *Applied Sciences*, vol. 8, no. 3, p. 421, 2018.

[99] M. Kulin *et al.*, "End-to-End Learning from Spectrum Data: A Deep Learning Approach for Wireless Signal Identification in Spectrum Monitoring Applications," *IEEE Access*, vol. 6, 2018.

[100] Fonseca, Erika and Santos, Joao F and Paisana, Francisco and DaSilva, Luiz A, "Radio Access Technology characterisation through object detection," *Computer Communications*, vol. 168, pp. 12–19, 2021.

[101] Doyle, Linda E and Sutton, Paul D and Nolan, Keith E and Lotze, Jorg and Ozgul, Baris and Rondeau, Thomas W and Fahmy, Suhaib A and Lahlou, Hicham and DaSilva, Luiz A., "Experiences from the Iris Testbed in Dynamic Spectrum Access And Cognitive Radio Experimentation," in *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, IEEE, 2010, pp. 1–8.

[102] "3rd Generation Partnership Project Technical Specification Group Radio Access Network," 3GPP, Tech. Rep., 2017.

[103] Miao, Guowang and Zander, Jens and Sung, Ki Won and Slimane, Slimane Ben, *Fundamentals of mobile data networks*. Cambridge University Press, 2016.

[104] Galkin, Boris and Fonseca, Erika and Lee, Gavin and Duff, Conor and Kelly, Marvin, "Experimental Evaluation of a UAV User QoS from a Two-Tier 3.6GHz Spectrum Network," in *IEEE ICC Workshops*, 2021.

[105] "View on 5G Architecture," 5GPPP Architecture Working Group, Tech. Rep., 2018.

[106] *Dense Air*, `https://denseair.net/`, Accessed: 2022-01-28.

[107] Cui, Jingjing and Ding, Zhiguo and Deng, Yansha and Nallanathan, Arumugam and Hanzo, Lajos, "Adaptive UAV-trajectory optimisation under quality of service constraints: A model-free solution," *IEEE Access*, vol. 8, pp. 112 253–112 265, 2020.

[108] *REQIBA*, `https://github.com/galkinb/REQIBA`, Accessed: 2022-01-11.

[109] Amer, Ramy and Saad, Walid and Galkin, Boris and Marchetti, Nicola, "Performance Analysis of Mobile Cellular-Connected Drones under Practical Antenna Configurations," in *2020 IEEE International Conference on Communications (ICC)*, IEEE, 2020.

[110] "Recommendation P.1410-5 "Propagation Data and Prediction Methods Required for the Design of Terrestrial Broadband Radio Access Systems Operating in a Frequency Range From 3 to 60 GHz"," ITU-R, Tech. Rep., 2012.

[111] Galkin, Boris and Kibilda, Jacek and DaSilva, Luiz A, "Coverage analysis for low-altitude UAV networks in urban environments," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, IEEE, 2017, pp. 1–6.

[112] Series, P, "Propagation data and prediction methods required for the design of terrestrial broadband radio access systems operating in a frequency range from 3 to 60 GHz," *Recommendation ITU-R*, 1410–1415, 2012.

[113] A. Bäuml, C. Felber, and W. P. Nitzold, "An Experimental Study of Multi-RAT Systems," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, IEEE, 2020, pp. 1–2.

[114] Miranda, João Paulo and Galkin, Boris and Abreu, Giuseppe and DaSilva, Luiz, "Experimental assessment of eigenvalue-based detection for cognitive radio," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, IEEE, 2014, pp. 157–160.

[115] 5G America, "5G Americas White Paper – Network Slicing for 5G and Beyond," 5G America, Tech. Rep., 2016.

[116] "Qualcomm Research LTE in Unlicensed Spectrum: Harmonious Coexistence with WiFi," Qualcomm, Tech. Rep., 2014, Accessed: 2022-01-11.

[117] J. Redmon *et al.*, "You Only Look Once: Unified, Real-Time Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2016.

[118] ——, "YOLO9000: Better, Faster, Stronger," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017.

[119] *Technology-classification-dataset*, `https://github.com/ewine-project/Technology-classification-dataset`, Accessed: 2022-01-11.

[120] *Dataset Generator of RF Waveforms*, `https://github.com/frankist/gr-specmonitor`, Accessed: 2022-01-11.

[121] *Software Radio Systems*, `https://www.softwareradiosystems.com/tag/srslte/`, Accessed: 2022-01-11.

[122] *IEEE 802.11 a/g/p transceiver for GNU Radio*, `https://github.com/bastibl/gr-ieee802-11`, Accessed: 2022-01-11.

[123] A. A. Alabdel Abass *et al.*, "WiFi/LTE-U Coexistence: An Evolutionary Game Approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 1, 2019.

[124] *LabelImg*, `https://github.com/tzutalin/labelImg`, Accessed: 2022-01-11.

[125] Svetnik, Vladimir *et al.*, "Random forest: a classification and regression tool for compound classification and QSAR modeling," *Journal of chemical information and computer sciences*, 2003.

[126] *30+ Smartphone Usage Statistics for the UK [2021]*, Setting off the 5G Advanced evolution, Accessed: 2022-09-19.