Cognitive Networking for Next Generation of Cellular Communication Systems

Metin Öztürk

Submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy

James Watt School of Engineering
College of Science and Engineering
University of Glasgow

May 2020

Abstract

This thesis presents a comprehensive study of cognitive networking for cellular networks with contributions that enable them to be more dynamic, agile, and efficient. To achieve this, machine learning (ML) algorithms, a subset of artificial intelligence, are employed to bring such cognition to cellular networks. More specifically, three major branches of ML, namely supervised, unsupervised, and reinforcement learning (RL), are utilised for various purposes: unsupervised learning is used for data clustering, while supervised learning is employed for predictions on future behaviours of networks/users. RL, on the other hand, is utilised for optimisation purposes due to its inherent characteristics of adaptability and requiring minimal knowledge of the environment.

Energy optimisation, capacity enhancement, and spectrum access are identified as primary design challenges for cellular networks given that they are envisioned to play crucial roles for 5G and beyond due to the increased demand in the number of connected devices as well as data rates. Each design challenge and its corresponding proposed solution are discussed thoroughly in separate chapters.

Regarding energy optimisation, a user-side energy consumption is investigated by considering Internet of things (IoT) networks. An RL based intelligent model, which jointly optimises the wireless connection type and data processing entity, is proposed. In particular, a Q-learning algorithm is developed, through which the energy consumption of an IoT device is minimised while keeping the requirement of the applications—in terms of response time and security—satisfied. The proposed methodology manages to result in 0% normalised joint cost—where all the considered metrics are combined—while the benchmarks performed 54.84% on average. Next, the energy consumption of radio access networks (RANs) is targeted, and a traffic-aware cell switching algorithm is designed to reduce the energy consumption of a RAN without compromising on the user quality-of-service (QoS). The proposed technique employs a SARSA algorithm with value function approximation, since the conventional RL methods struggle with solving problems with huge state spaces. The results reveal that up to 52% gain on the total energy consumption is achieved with the proposed technique, and the gain is observed to reduce when the scenario becomes more realistic.

On the other hand, capacity enhancement is studied from two different perspectives, namely mobility management and unmanned aerial vehicle (UAV) assistance. Towards that end, a predictive handover (HO) mechanism is designed for mobility management in cellular networks by identifying two major issues of Markov chains based HO predictions. First, revisits—which are defined as a situation whereby a user visits the same cell more than once within the same day—are diagnosed as causing similar transition probabilities, which in turn increases the likelihood of making incorrect predictions. This problem is addressed with a structural change; i.e., rather than storing 2-D transition matrix, it is proposed to store 3-D one that also includes HO orders. The obtained results show that 3-D transition matrix is capable of reducing the HO signalling cost by up to 25.37%, which is observed to drop with increasing randomness level in the data set. Second, making a HO prediction with insufficient criteria is identified as another issue with the conventional Markov chains based predictors. Thus, a prediction confidence level is derived, such that there should be a lower bound to perform HO predictions, which are not always advantageous owing to the HO signalling cost incurred from incorrect predictions. The outcomes of the simulations confirm that the derived confidence level mechanism helps in improving the prediction accuracy by up to 8.23%.

Furthermore, still considering capacity enhancement, a UAV assisted cellular networking is considered, and an unsupervised learning-based UAV positioning algorithm is presented. A comprehensive analysis is conducted on the impacts of the overlapping footprints of multiple UAVs, which are controlled by their altitudes. The developed k-means clustering based UAV positioning approach is shown to reduce the number of users in outage by up to 80.47% when compared to the benchmark symmetric deployment.

Lastly, a QoS-aware dynamic spectrum access approach is developed in order to tackle challenges related to spectrum access, wherein all the aforementioned types of ML methods are employed. More specifically, by leveraging future traffic load predictions of radio access technologies (RATs) and Q-learning algorithm, a novel proactive spectrum sensing technique is introduced. As such, two different sensing strategies are developed; the first one focuses solely on sensing latency reduction, while the second one jointly optimises sensing latency and user requirements. In particular, the proposed Q-learning algorithm takes the future load predictions of the RATs and the requirements of secondary users—in terms of mobility and bandwidth—as inputs and directs the users to the spectrum of the optimum RAT to perform sensing. The strategy to be employed can be selected based on the needs of the applications, such that if the latency is the only concern, the first strategy should be selected due to the fact that the second strategy is computationally more demanding. However, by employing the second strategy, sensing latency is reduced while satisfying other user requirements. The simulation results demonstrate that, compared to random sensing, the first strategy decays the sensing latency by 85.25%, while the second strategy enhances the full-satisfaction rate, where both mobility and bandwidth requirements of the user are simultaneously satisfied, by 95.7%.

Therefore, as it can be observed, three key design challenges of the next generation of cellular networks are identified and addressed via the concept of cognitive networking, providing a utilitarian tool for mobile network operators to plug into their systems. The proposed solutions can be generalised to various network scenarios owing to the sophisticated ML implementations, which renders the solutions both practical and sustainable.

University of Glasgow

College of Science & Engineering

Statement of Originality

Name: Metin Öztürk
Registration Number:

I certify that the thesis presented here for examination for a PhD degree of the University of Glasgow is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it) and that the thesis has not been edited by a third party beyond what is permitted by the University's PGR Code of Practice.

The copyright of this thesis rests with the author. No quotation from it is permitted without full acknowledgement.

I declare that the thesis does not include work forming part of a thesis presented successfully for another degree.

I declare that this thesis has been produced in accordance with the University of Glasgow's Code of Good Practice in Research.

I acknowledge that if any issues are raised regarding good research practice based on review of the thesis, the examination may be postponed pending the outcome of any investigation of the issues.

Signature: ....
Date: ....

List of Publications

Journals

1. **Ozturk, M.**, Abubakar, A., Rais, R. N. B., Jaber, M., Hussain, S. and Imran, M. A. (2020) Context-Aware Connectivity and Processing Optimization for IoT Networks. *IEEE Internet of Things Journal* (Under review).

2. Abubakar, A., **Ozturk, M.**, Rais, R. N. B., Hussain, S. and Imran, M. A. (2020) Load-Aware Cell Switching in Ultra-Dense Networks: An Artificial Neural Networks Approach. *IEEE Networking Letters* (Under review).

3. Mollel, M., Abubakar, A., **Ozturk, M.**, Kaijage, S., Kisangiri, M., Zoha, A., Imran, M. A. and Abbasi, Q. H. (2019) Intelligent Handover Decision Scheme Using Double Deep Reinforcement Learning. *Physical Communication* (Under review).

4. **Ozturk, M.**, Akram, M., Hussain, S. and Imran, M. A. (2019) Novel QoS-aware proactive spectrum access techniques for cognitive radio using machine learning. *IEEE Access*, 7, pp. 70811-70827, May 2019.

5. **Ozturk, M.**, Gogate, M., Onireti, O., Adeel, A., Hussain, A. and Imran, M. A. (2019) A novel deep learning driven low-cost mobility prediction approach for 5G cellular networks: The case of the Control/Data Separation Architecture (CDSA). *Neurocomputing*, page 479-489, September 2019.

6. **Ozturk, M.**, Imran, M. and Jaber, M. (2018) Energy-aware smart connectivity for IoT networks: enabling smart ports. *Wireless Communications and Mobile Computing*, June, 2018.

Book Chapters

1. **Ozturk, M.**, Abubakar, A., Hussain, S., Abbasi, Q. H. and Imran, M. A. (2019) Cognitive Radio Spectrum Sensing: From Conventional Approaches to Machine Learning-based Predictive Techniques. Submitted to *Flexible and Cognitive Radio Access Technologies for 5G and Beyond*, IET (Under review).

2. **Ozturk, M.**, Valente Klaine, P., Hussain, S. and Imran, M. A. (2019) Predictive Mobility Management in Cellular Networks. Submitted to *AI for Emerging Verticals: Human-Robot Computing, Sensing and Networking*, IET (Under review).

3. Imran, M. A., **Ozturk, M.**, Abubakar, A. I., Valente Klaine, P., Hussain, S. and Abbasi, Q. H. (2019) Mobility prediction based resource management. In: Tafazolli, R., Wang, C.-L. and Chatzimisios, P.(eds.) *Wiley 5G REF: the Essential 5G Reference Online.* Wiley. ISBN 9781119471509 (Accepted for publication).

4. Imran, M. A., Turkmen, A., **Ozturk, M.**, Nadas, J. P. B. and Abbasi, Q. H. (2019) Seamless Indoor/Outdoor Coverage in 5G. Submitted to In: Tafazolli, R., Wang, C.-L. and Chatzimisios, P. (eds.) *Wiley 5GREF: the Essential 5G Reference Online.* Wiley. ISBN 9781119471509 (Accepted for publication).

5. **Ozturk, M.**, Jaber, M. and Imran, M. A. (2019) Life-span extension for sensor networks in the industry. In: Imran, M. A., Hussain, S. and Abbasi, Q. H. (eds.) *Wireless Automation as an Enabler for the Next Industrial Revolution.* Wiley-IEEE Press, pp. 19-45. ISBN 9781119552611.

Conference Proceedings

1. Asad, S. M., **Ozturk, M.**, Rais, R. N. B., Zoha, A., Hussain, S., Abbasi, Q. H. and Imran, M. A. (2019) Reinforcement Learning Driven Energy Efficient Mobile Communication and Applications. In: 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Ajman, United Arab Emirates, 10-12 Dec 2019, ISBN 9781728153414.

2. **Ozturk, M.**, Nadas, J. P.B., Klaine, P. H.V., Hussain, S. and Imran, M. A. (2019) Clustering Based UAV Base Station Positioning for Enhanced Network Capacity. In: International Conference on Advances in the Emerging Computing Technologies (AECT 2019), Medina, Saudi Arabia, 08-10 Dec 2019 (Accepted for publication).

3. Abubakar, A., **Ozturk, M.**, Hussain, S. and Imran, M. (2019) Q-learning Assisted Energy-Aware Traffic Offloading and Cell Switching in Heterogeneous Networks. In: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, 11-13 Sep 2019, ISBN 9781728110165.

4. Sambo, Y. A., Valastro, G. C., Patané, G. M. M., **Ozturk, M.**, Hussain, S., Imran, M. A. and Panno, D. (2019) Motion Sensor-based Small Cell Sleep Scheduling for 5G Networks. In: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, 11-13 Sep 2019, ISBN 9781728110165.

5. Patané, G. M. M., Valastro, G. C., Sambo, Y. A., **Ozturk, M.**, Hussain, S., Imran, M. A., Panno, D.(2019) Flexible SDN/NFV-based testbed for 5G mobile networks, IEEE/ACM DS-RT 2019, Cosenza, Italy, 7-9 Oct. 2019, ISBN 9781728129235.

6. **Ozturk, M.**, Abubakar, A., Ul Hassan, N., Hussain, S., Imran, M. and Yuen, C. (2019) Spectrum Cost Optimization for Cognitive Radio Transmission over TV White Spaces Using Artificial Neural Networks. In: 4th International Conference on UK - China Emerging Technologies (UCET 2019), Glasgow, UK, 21-22 Aug 2019, ISBN 9781728127972.

7. Turkmen, A., Mollel, M. S., **Ozturk, M.**, Sun, Y., Zhang, L., Ghannam, R. and Imran, M. A. (2019) Coverage Analysis for Indoor-Outdoor Coexistence for Millimetre-Wave Communication. In: 4th International Conference on UK - China Emerging Technologies (UCET 2019), Glasgow, UK, 21-22 Aug 2019, ISBN 9781728127972.

8. Mollel, M., **Ozturk, M**., Kaijage, S., Kisangiri, M., Onireti, O., Imran, M. A. and Abbasi, Q. H. (2019) Handover Management in Dense Networks with Coverage Prediction from Sparse Networks. In: IEEE Wireless Communications and Networking Conference (IEEE WCNC 2019), Marrakech, Morocco, 15-18 April 2019, ISBN 9781728109220.

9. **Ozturk, M.**, Valente Klaine, P. and Imran, M. A. (2018) Introducing a Novel Minimum Accuracy Concept for Predictive Mobility Management Schemes. In: IEEE International Conference on Communications (ICC 2018), Kansas City, MO, USA, 20-24 May 2018, ISBN 9781538643280.

10. **Ozturk, M.**, Valente Klaine, P. and Imran, M. A. (2018) 3D Transition Matrix Solution for a Path Dependency Problem of Markov Chains-Based Prediction in Cellular Networks. In: IEEE VTC 201 7BackNets Workshop, Toronto, Canada, 24-27 Sept 2017, ISBN 9781509059355.

11. **Ozturk, M.**, Valente Klaine, P. and Imran, M. A. (2018) Improvement on the Performance of Predictive Handover Management by Setting a Thresh-

old. In: IEEE VTC 2017 BackNets Workshop, Toronto, Canada, 24-27 Sept 2017, ISBN 9781509059355.

Contents

List of Tables

List of Figures

List of Acronyms

1G	First Generation of Cellular Networks
2G	Second Generation of Cellular Networks
3G	Third Generation of Cellular Networks
3GPP	3G Partnership Project
4G	Fourth Generation of Cellular Networks
5G	Fifth Generation of Cellular Networks
5G NR	5G New Radio
AC	Alternating Current
ACK	Acknowledgement
ADC	Analog to Digital Convertor
AI	Artifical Intelligence
ANN	Artificial Neural Networks
AR	Autoregressive
AWGN	Additive White Gaussian Noise
BP	Backpropagation
BS	Base station
CAPEX	Capital Expenditure
CBS	Control Base Station
CD	Cyclostationary Detection
CDMA	Code Division Multiple Access
CDR	Call Detail Record
CDSA	Control/Data Seperated Architecture
CNN	Convolutional Neural Network
CP	Correct Prediction
CR	Cognitive Radio
CRBS	Cognitive Radio Base Station
CRN	Cognitive Radio Network
CRS	Cognitive Radio System
DBS	Data Base Station
DC	Direct Current

DTMC	Discrete-time Markov Chains
EARTH	Energy Aware Radio and Network Technologies
EM	Electromagnetic
eMBB	Enhanced Mobile Broadband
eNodeB	Evolved Node B
eSIM	Electronic Subscriber Identity Module
ESN	Echo State Network
FCC	Federal Communications Commission
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
FR	Frequency Range
GSM	Global System for Mobile Communications
HBRA	High-Battery Requirement-Aware
HetNet	Heteregeneous Network
HL	Hidden Layer
HMM	Hidden Markov Model
HO	Handover
IANN	Improved ANN
IIoT	Industrial Internet of Things
IMT-2000	International Mobile Telephone by the Year 2000
IoT	Internet of Things
IP	Internet Protocol
IQ	In-phase Quadrature
IRS	Intelligent Reflecting Surface
ITU	International Telecommunoication Union
ITU-R	International Telecommunoication Union-Radio Communication Sector
ITU-T	International Telecommunoication Union-Telecommunication Standardisation Sector
KPI	Key Performance Indicator
LBRA	Low-Battery Requirement-Aware
LBRU	Low-Battery Requirement-Unaware
LOS	Line of Sight
LSTM	Long-Short Term Memory
LTE	Long Term Evolution
M2M	Machine-to-Machine
MAC	Medium Access Control
MC	Macro Cell

MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLP	Multi-Layer Perceptron
mMTC	Massive Machine Type Communications
mmWave	Millimetre Wave
MSE	Mean Squared Error
MTC	Machine Type Communication
NB	Narrowband
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OPEX	Operational Expenditure
PLMN	Public Land Mobile Network
PST	Probability Suffix Tree
PU	Primary User
QoE	Quality-of-Experience
QWBPS	Q-Learning Enabled Wideband Predictive Sensing
RAN	Radio Access Network
RAT	Radio Access Technology
RB	Resource Block
RL	Reinforcement Learning
RLF	Radio Link Failure
RNN	Recursive Nural Network
RRC	Radio Resource Control
RRH	Remote Radio Head
RSRP	Reference Signal Received Power
RSSI	Received Signal Strength Indicator
SARSA	State-Action-Reward-State-Action
SC	Small Cell
SINR	Signal-to-Interference-plus-Noise Ratio
SISO	Single-Input Single-Output
SMS	Short Message Service
SN	Sequence Number
SNR	Signal-to-Noise-Ratio
SU	Secondary User
TDMA	Time Division Multiple Access
TTT	Time-to-Trigger
TV	Television
UAV	Unmanned Aerial Vehicle

UDN	Ultra-Dense Network
UE	User equipment
UMTS	Universal Mobile Telecommunications System
URLLC	Ultra-Reliable Low Latency Communications
VFA	Value Function Approximation
VoIP	Voice over Internet Protocol
WB	Wideband
WBPS	Wideband Predictive Sensing
WCDMA	Wideband Code Division Multiple Access
WSN	Wireless Sensor Network

Acknowledgements

Only my name may appear on the cover page of this thesis, but indeed there are many others behind the scenes, without whom this thesis would not have been possible. Therefore, I would like to thank all of them for their continuous and unlimited support, help, and understanding.

First, I would like to express my deepest gratitude to my supervisor, Prof. Muhammad Ali Imran, for his never-ending support, motivation, and valuable guidance. Whenever I felt stuck, demotivated, or stressed, his mentorship lent me a helping hand, and I always left his office happier, more motivated, and more hopeful. I also would like to thank him for that he continually shared his immense knowledge and experience with me in order for me to be a good academic and researcher.

Secondly, I would like to convey my most sincere thanks to my co-supervisor, Dr. Sajjad Hussain, for the exceptional support, encouragement, and patience. I have learnt a lot from him through our stimulating discussions, and his feedbacks greatly helped me improve myself. Moreover, he always valued my opinions and ideas as well as being sincere and understanding, for which I am really grateful.

I also thank my scholarship provider, Republic of Turkey Ministry of National Education, for providing this great opportunity, without which this story would have never even started. As such, I would like to thank my guarantors, Bünyamin Öztürk—my elder brother—and Dr. Abdullah Yıldızbaşı, as well for facilitating the scholarship process and for believing in me. I am also thankful to my colleagues and friends; I strongly believe that my Ph.D. adventure has been made a lot easier and smoother with their solidarity and support.

If there was one single person, who deserves all the credit for this thesis, I would definitely point my beloved wife, Belgin Öztürk. She always supported me throughout my Ph.D., and always stood with me at the hardest times. She, with her incredible understanding and patience, sacrificed her own interest, desires, and career for me. Without her love, companion, and understanding, I would never have been able to succeed.

Since 18 August 2018, I have been having another moral support, because it is the day my lovely daughter Meva Nil Öztürk was born; she has been boosting my energy and hope with her presence, helping me in dealing with the difficult times during my Ph.D.

Last but not the least, I am extremely grateful and appreciative to my dear family; especially my father, Ahmet Öztürk, and my mother, Nazik Öztürk. From the very beginning of my life, they have been motivating me to pursue further studies, for which they dedicated their whole life. They sacrificed everything including their personal happiness and needs, so that I could be successful. Their support, help, mentorship, understanding, patience, and compassion have always been limitless. I would also like to thank my sister Sebile Çeker for her continuous support and motivation; she helped me pave my way towards this direction in my childhood.

Chapter 1

Introduction

1.1 Cellular Communication Networks

A cellular system is a type of wireless communication networks, where base stations (BSs) are deployed over wide geographical areas to connect user equipments (UEs) with each other. The name *cellular* comes from the fact that the considered geographical areas are split into *cells*, at which BSs are deployed in order to provide the required connectivity for all the UEs [1, 2]. As such, each BS, which is equipped with at least one antenna to produce the wireless electromagnetic (EM) radiation, constitutes a cell and serves the UEs located under its coverage area. However, only a small portion of the EM spectrum is allocated for cellular communications by the International Telecommunication Union (ITU), thereby BSs need to use their radio resources as efficiently as possible in order to provide service for users in a more efficient manner [1, 3].

Since the early 1980s, when the first generation of cellular networks (1G) was introduced, the cellular communication has evolved significantly. 1G utilised frequency division multiple access (FDMA) technology and it was entirely analog, meaning that it modulated voice signals to higher frequency channels, instead of converting them to digital bit streams [1, 2]. Therefore, due to the analog nature of 1G communications, it had limited capabilities, which led to its disappearance after the introduction of the second generation of cellular networks (2G), which was digital based. 2G encodes an analog voice signal to a digital signal, which is advantageous in not only mitigating errors in transmissions, but also more efficient in terms of radio resource management [1, 2]. The first 2G system was the Global System for Mobile Communications (GSM), which was developed in 1991, and it utilised time division multiple access (TDMA) method. One of the main differences from its predecessor is the introduction of short message service (SMS). In the further versions of 2G; starting from 2.5G, packet switching was also included in the system, which helped to reach 144 kbps data rate [4].

The development of international mobile telephone by the year 2000 (IMT-2000) by ITU in the early 1990s paved the way for the first commercial release of the third generation of cellular networks (3G) in the early 2000s. Two major standards, called Universal Mobile Telecommunications System (UMTS)—also referred as Wideband Code Division Multiple Access (WCDMA)—and Code Division Multiple Access (CDMA)2000, were developed with diverse characteristics. Even though a 2 Mbps minimum downlink data rate was expected from 3G at the beginning, its further versions, such as 3.5G and 3.75G, a peak data rate of up to several Mbps was achieved, enabling some new applications, such as video conferencing. However, the ever-increasing data demands, the growing number of connected users, and the need for higher data rates made 3G networks outdated due to the nature of the underlying CDMA technology[1] being employed in 3G [6]. Moreover, the operational cost of 3G was also high [7], making it challenging for mobile network operators to maintain their networks. These kind of limitations, combined with improvements in the technology, led to the emergence of the fourth generation of cellular networks (4G). As such, in 2009, the ITU-Radio communication sector (ITU-R) released the requirements for 4G, which stated that the peak data rate for low mobility users should be of 1 Gbps, while it is 100 Mbps for high mobility users [8]. Circuit switching was completely abandoned with 4G, and all-internet protocol (IP) based communication system was adopted. Moreover, the 3G Partnership Project (3GPP) developed a standard for 4G networks, regarded as Long Term Evolution (LTE), which considered orthogonal frequency division multiple access (OFDMA) and multiple-input multiple-output (MIMO) technologies in order to increase the peak data rate.

Nevertheless, with the introduction of emerging concepts, including the Internet of things (IoT) and machine type communications (MTC), 4G would be no more sufficient due to its inherent human-type traffic based design [9]. Furthermore, new applications, such as augmented reality and remote surgery, are quite demanding in terms of bandwidth and latency. In remote surgery, for example, a reliable communication link with minimal latency is required, since any fault and/or delay in the communication would result in undesired consequences; e.g., incorrect movements of a robotic arm or inadequate precision for surgeons. Therefore, it is clear that these new technologies and applications necessitate a better cellular communication network that can provide not only higher data

[1] There is a design challenge with this modulation scheme called near-far problem [1, 5]. Since more power is received from a user, who is located nearby the BS, compared to the far user, the demodulation of the signal from the far user becomes more difficult owing to the fact that the near user increases the noise floor of the demodulator at the receiver [1].

rates, but also lower latency, and improved reliability. This, by its turn, paved the way for the development of the fifth generation of cellular networks (5G). Because of its promising offers and high expectations, 5G has been perceived as not only an evolution for the cellular networks but also a revolution [10–12]. In addition, it is not an imagination any more, since it has already loomed on the horizon with its first release, which is envisioned for 2020[2].

In terms of standardisation, 3GPP has been working on it, and has developed the concept of 5G New Radio (NR) and published the first set of standards on its Release 15 in 2018. Moreover, in order to provide the adequate requirements for each application, 5G has seen its use-cases split into three main categories, namely: enhanced mobile broadband (eMBB), massive MTC (mMTC), and ultra-reliable and low latency communications (URLLC) [13]. Nonetheless, each scenario comes with stringent requirements including 20 Gbps downlink (DL) and 10 Mbps uplink (UL) target peak data rates, and 0.5 ms DL and 0.5 ms UL latencies. In this regard, in order to meet these requirements and address near future challenges, several new technologies have been proposed. For cellular networks capacity enhancement, for example, millimetre wave (mmWave) communications, massive MIMO, and network densification are some of the possible solutions listed in [14], in which it is mentioned that all these solutions should be considered together in order to satisfy the 5G requirements of data rates.

As such, mmWave frequencies have been included in 5G NR [15] as Frequency Range (FR)-2 along with traditional sub-6 GHz frequency bands. Nevertheless, albeit huge bandwidth availability, mmWave communication comes with its inherent problems, since it is more prone to suffer from severe penetration losses, owing to propagation characteristics of higher frequencies [16–18]. This subsequently reduces the coverage area of mmWave BSs, leading to more BS deployments to compensate the coverage holes [19]. Provided that the antenna size is proportional to the wavelength of the transmitted signal, the use of mmWave frequencies facilitates the deployment of multiple antennas, which in turn puts MIMO-based communications at the heart of 5G.

The primary benefits of MIMO over a traditional single-input single-output (SISO) systems, where UE and its serving BS have only one transmitting and receiving antennas, are discussed in [20] and [21] with the main takeaways of better capacity, enhanced reliability, improved energy efficiency, etc. Better capacity is achieved by the reduced interference due to better directivity of beams, which also helps in better frequency reuse that subsequently boosts the spec-

[2] 5G is already made available in some countries, such as United Kingdom, Finland, and United States, to name a few.

tral efficiency. Moreover, reliability comes from diversity, where the receiver gets multiple copies of the transmitted signal, and energy efficiency is achieved from less energy consumption per antenna owing to the directivity of the beam [21]. Despite all these benefits theoretically expanding with an increasing number of antennas [21], there are some issues, such as channel reciprocity, pilot contamination, and orthogonality of channel responses, that limit the amount of antennas to be deployed [22].

Network densification is another candidate to enhance network capacity, which subsequently improves the spectral efficiency [23–25]. The idea of network densification is that small cells (SCs), which are compact and low-power BSs, are deployed under the coverage area of conventional macro cells (MCs) in order to provide higher data rates to the users by bringing the BSs closer. Owing to the low transmit power of SCs [23, 24], their coverage areas are much smaller when compared to the MC, enabling frequency reuse and improving the network capacity. While, on one hand, it is better to increase the number of SCs in order to obtain more capacity, on the other hand, it will make the effect of interference more significant, which would eventually undermine the obtained capacity gain [14].

1.2 Motivation

With the advances in wireless communications and proliferations of new technologies, the demand for mobile communication networks has been perpetually increasing. This, in turn, inflates not only the number of subscribers but also the data volume requested by users. The report published by Ericsson [26] reveals interesting findings. The number of global subscriptions, for example, increased by 44 million during the first quarter of 2019, making the overall global number 7.9 billion, which is estimated to hit 8.8 billion by 2024. Moreover, the global mobile data traffic volume grew by 82% between first quarters of 2018 and 2019.

In addition, the transition between generations of mobile networks is also quite dynamic, with the market share of older generations taken by their successors. Another report published by GSM Association [27] suggests that while the rate of global 2G connections was 29% in 2018, it is estimated to collapse to 5% by 2025. Similarly, the share of global 3G connections is estimated to shrink from 28% to 20% between 2018 and 2025, whereas the global share of 4G and 5G connections are predicted to hit 59% and 15% by 2025, respectively.

On the other hand, the emergence of IoT, which is regarded as a paradigm shift in communication technology [28, 29], has been a rising issue, since it is dedicated to connect various devices, including home appliances, industrial machineries, and vehicles, to the Internet. The Telecommunication standardisation sector of International Telecommunication Union (ITU-T) defines IoT as "*a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies*" [30]. The key points in this definition are the interconnection and interoperation, since they mean that the "things" are supposed to communicate and operate with each other, minimising or eliminating human intervention completely. Therefore, due to minimal human interaction, processes; e.g., production, management, etc., can be made more efficient, dynamic, and agile, which in turn accelerates the proliferation of IoT in various domains including industry and routine life [28, 29, 31]. Despite IoT already being part of wireless communication networks, it is projected to be an integral part in the near future. The same report from Ericsson [26] forecasts that the number of global IoT connections will reach 22.3 billion by 2024, which represents a 17% increase when compared to 2018, with 4.1 billion of the connections being cellular IoT. Therefore, it is evident that a special care should be taken in order to combat the dramatic upsurge in the number of connections.

Another alarming aspect for mobile communication systems is the breeding of new technologies, including augmented reality, remote surgery, 4K video streaming, and tactile internet, to name a few. Given the stringent bandwidth and latency requirements of these type of applications, the current situation becomes even more challenging. For the case of augmented reality, for instance, bandwidth, end-to-end latency, and reliability play crucial roles, since for this type of applications, in addition to being intolerant to delays and faults in the data transmission, a large bandwidth—in order to stream all possible angles—is required [32, 33]. Having all these said, with the ever-increasing number of connections and data demand as well as advances in technology that brings demanding applications, the near future is projected to be quite exigent for mobile communications, making the legacy networks, such as 3G and 4G, outdated.

As mentioned earlier, even though mmWave communication offers huge amount of bandwidth, which subsequently increases the data rate significantly, it has its own issues due to the nature of higher frequency bands. Compared to the frequencies employed in legacy 4G networks, mmWave frequencies are more vulnerable to attenuations, since its penetration properties are much lower, putting the line of sight (LOS) at the heart of mmWave networks [16]. Therefore, mmWave communication has smaller coverage areas than legacy networks, which is around 200 m, and also depends on the environment [17]. Given that there is a considerable amount of studies available in the literature that discusses the use of terahertz band (between 0.1 and 10 THz) for wireless communications, the coverage areas would be even more reduced (around 10 to 20 m) [34].

Network densification is reported in [17] as another enabling concept to address the above-mentioned problems in mobile communication networks. According to this concept, networks are densified with the deployments of SCs, which in turn result in so called ultra-dense networks (UDNs). Owing to the fact that SCs are mostly low power devices, their coverage areas are also limited, an thus they are deployed in bulk which paves the way for more spectral efficiency, as the same frequencies can be reused within the area-of-interest. In addition, mmWave communications and SCs concepts can be combined [35], where the SCs use the mmWave frequencies, in order to benefit from both higher available bandwidth and enhanced spectral efficiency.

In this regard, this thesis identifies energy optimisation, capacity enhancement, and dynamic spectrum access as key design challenges for the next generations of cellular communication networks. Using the summary in Fig. 1.1, the main rationales behind considering these challenges are as follows:

- they are identified by accounting the demands from the next generations of the cellular networks;
- they are quite inter-connected to each other, such that a candidate solution for one results in additional issues for the others:
    - mmWave communications and network densification are enablers for capacity enhancement, however, they would cause a significant increase in terms of both the energy consumption of cellular networks and the frequency of handovers (HOs) for users. This is due to fact that the footprints of BSs get smaller with increasing carrier frequency, leading to more BS deployments, which in turn increases the network power consumption and the number of HOs experienced by users.
    - a better spectrum access opens up a new dimension for capacity enhancement by exploiting the existing capacity in a more efficient way, thereby i) more users may be accommodated, ii) there becomes more availability for bandwidth-hungry applications.
    - the efficiency of dynamic spectrum access is provided by allowing unlicensed users to use the EM frequency spectrum opportunistically via spectrum sensing. Given that each sensing attempt is energy and time consuming, the proposed spectrum access technique is directly interrelated with energy optimisation.

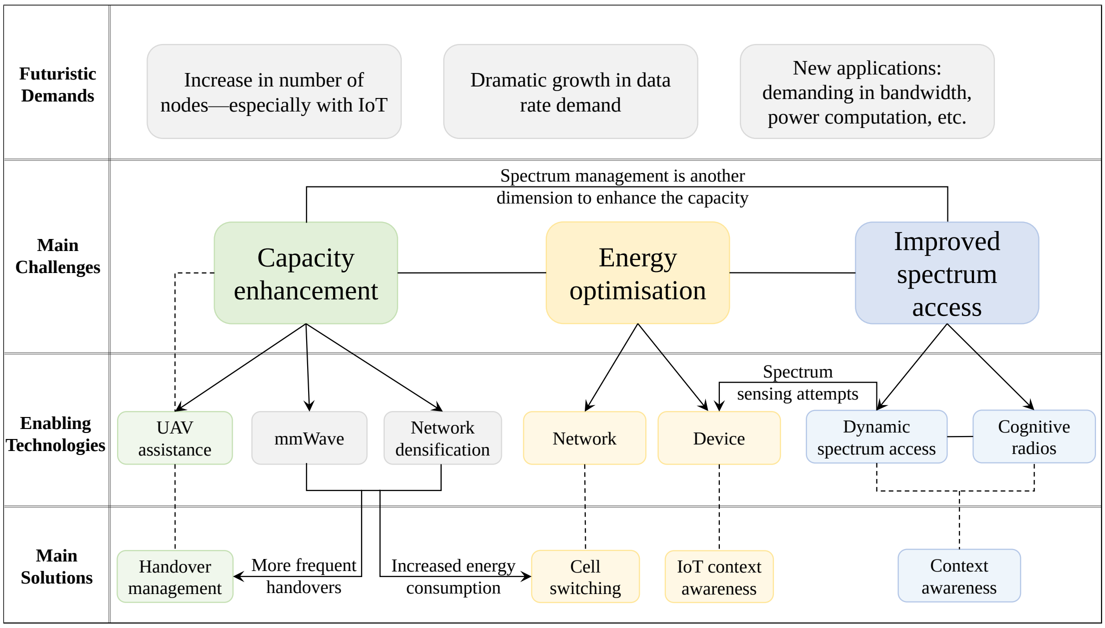


Figure 1.1: The summary of the thesis with identified key design challenges and their inter-connection.

Therefore, it is considered as appropriate to originate this thesis around these three design challenges, as the strong inter-connection between them creates the underlying story. Besides, machine-centric solutions, where intelligence is introduced to cellular communication networks, are lying at the core of this thesis due to following reasons:

- the volume of data to be generated from cellular networks is expected to soar owing to more BS deployments and the inclusion of IoT devices [28, 29, 31, 36].
- cellular networks have been becoming more complex with each new generation, and they will be even more complex with 5G [4, 14, 17, 37, 38], making fixed and static solutions less reliable.
- dynamic environments make the optimisation processes harder, thereby learning from generated data and gaining experience from previous actions will be vital in order to make the networks more informed and proactive.

1.2.1 Need for Energy Optimisation

The energy consumption of cellular networks is escalating greatly as a result of the exponential increase in data demand, thereby forcing mobile network operators to continually expand their networks by deploying more BSs to accommodate this data crunch. In addition, the provision of local support for IoT in 5G to meet the growing demand for applications, including smart health, smart homes, and smart cities, would result in the massive deployment of IoT devices [39], which leads to a further increase in the energy consumption of cellular networks.

Network densification, comprising the ultra-dense deployment of SCs alongside MCs has been identified as one of the enablers for achieving the one thousand fold capacity increase in 5G networks [40]. SCs are low power nodes which are deployed to boost the capacity at hotspot zones—the busiest areas of a deployment. However, the massive deployment of SCs would also result in a dramatic increase in energy consumption. Moreover, the utilisation of mmWave spectrum also results in more BS deployments due to the reduced coverage areas of mmWave BSs.

Although the increasing amount of IoT devices also contributes to the increase in energy consumption of the network, the lifetime of these devices are another important aspect to consider from the energy perspective. Provided that IoT devices are mostly battery operated, their energy consumption becomes a crucial aspect in order to keep the system operational. Therefore, prolonging the battery life of IoT devices is vital, since it is reported in [39] that low-power operation is essential for the majority of the applications. In this regard, sensing, data processing, and communication of IoT devices are supposed to be optimised in a way that they are performed efficiently with a reduced amount of energy consumption.

In addition to that, increasing the energy consumption of cellular networks also causes two severe adverse effects:

- an increase in green house gas emission, harming the environment;
- an increase in the network operating cost, due to higher energy bills charged to mobile network operators [41].

Therefore, energy optimisation needs to be given more consideration in 5G networks so that the increase in network capacity due to network densification does not incur on a large increase in energy consumption, thereby making it cost-effective and, most importantly, environmentally friendly.

1.2.2 Need for Network and User Capacity Enhancement

As mentioned earlier in this chapter, some of the emerging applications are very demanding in bandwidth, and there is an increasing number of users who request these type of applications. According to [42], 1 terabyte of annual data is going to be downloaded by an average user in 2020. Obviously, this will create a data crunch, and the legacy cellular networks would not be able to deal with this huge amount of traffic [38]. Furthermore, accounting the dramatic upsurge in the number of cellular connections forecasted with the IoT networks, it is quite self-evident that the burden of the cellular networks will be a lot higher. Hence, this needs to be addressed efficiently, and 5G has already been envisioned to tackle this data crunch with state-of-the-art concepts including mmWave, massive MIMO, and network densification.

Enhancing the network capacity by providing plenty of spectral resources is not the only strength of mmWave, as it is also linked to the other two aforementioned 5G enablers:

- it paves the way for massive MIMO technology, since the size of antennas become smaller due to the carrier frequency (the higher the carrier frequency, the smaller the wavelength is, and so is the antenna size).

- owing to the reduced coverage areas of the mmWave BSs, the network densification is needed in order to compensate the coverage holes.

Therefore, with the utilisation of mmWave communication in cellular networks, the gain will not only come from the abundant bandwidth resources, but also from massive MIMO and network densification.

Another aspect that is worth mentioning is the user capacity. Although cellular networks would obtain a significant gain from mmWave communications, the issue of mobility management immediately emerges. Mobility is an integral part of all types of wireless communication networks, and it is even more critical for 5G networks as a consequence of more heterogeneity in the networks [40]. HO refers to the situation, where a UE changes its serving BS while being in an active mode [43]. For this to happen, normally, the UE performs measurements on signal quality received from the surrounding BSs, and reference signal received power (RSRP) is used as a key performance indicator (KPI). Then, the UE switches its connection from the serving to the target BS once certain criteria are met. However, the UE experiences service interruptions during HO, since no data is transmitted before the HO is executed successfully. Given the reduced coverage areas of mmWave frequencies and SCs, the HO management becomes more critical, because the number of HOs increases [44], so is the service interruptions. This would undermine the gain obtained from the enabling concepts, such as mmWave, network densification, and massive MIMO. In other words, the service interruptions boost with growing number of HOs, which in turn damages user quality of experience (QoE). Considering that the average throughput is inversely proportional to the number of HOs and the time spent for each HO [45], the more HO leads to a lower average throughput. Besides, if not managed properly, this greater number of HOs increases the probability of HO failures, where the HO is not executed successfully due to various reasons, including: radio link failure (RLF) during time-to-trigger (TTT) and HO preparation [46]. Therefore, an accurate HO management scheme should focus on reducing the number of HOs and/or time spent for each HO in order to improve average throughput and avoid HO failures.

1.2.3 Need for Dynamic Spectrum Access

Having discussed the imminent need for energy optimisation and capacity enhancement regarding the increasing number of connections and data demand, there is still room for improvement in terms of spectrum efficiency. Especially considering the inflation in the number of devices trying to access the spectrum, which will be more with the IoT inclusion, the situation becomes even more challenging from the resource perspective. It is quite intuitive that the mmWave frequency band would relive the given burden to some extent, but additional solutions would be required to manage it properly for scenarios with a large number of users/devices [14]. Put it another way, even though the aforementioned network capacity enhancement concepts are beneficial to alleviate the problem, the need for higher data rates and thus more bandwidth is increasing, making the issue not completely solved.

It has been a truism that some portions of the frequency spectrum, such as the one assigned to television (TV) broadcasting, are underutilised [47–49], while cellular networks have been suffering from overutilisation. As such, the spectrum portions that are currently underutilised can be opened up for cellular communication networks. This idea is developed under the concept of cognitive radio (CR), or dynamic spectrum access, which has been proposed in 1999 [50] in order to use the EM frequency spectrum more efficiently. To mitigate the congestion in cellular networks, two different types of users have been introduced in the CR concept: primary users (PUs) and secondary users (SUs). The former is a licensed user who always has priority to access the spectrum, while the latter is unlicensed and can use the spectrum opportunistically. Given that SUs utilise the vacant portions of the spectrum, this process not only improves the spectrum efficiency, but also eases the congestion in cellular networks, especially in ultra-dense scenarios. CR has already been identified as an important approach to mitigate the aforementioned demanding applications and load burden of 5G networks [29].

A special case mentioned in [29] states that the deployment of new infrastructure to rural areas would harm the business of cellular network operators, since their capital expenditure (CAPEX) would increase with a limited profit. This occurs due to the limited number of residents in those areas, which also hinders the deployment of IoT networks. On the other hand, by using the unlicensed spectrum, CR technology can help in providing the spectrum access to the IoT devices using the TV white spaces, making network management in rural areas more cost-effective.

1.2.4 Need for Machine-Centric Solutions in Optimisation Process

Machine learning (ML), a part of artificial intelligence (AI), is a set of algorithms aiming to learn from given data without any explicit coding [51], meaning that it tries to understand the patterns hidden in the data set. There are three main categories in ML, namely supervised learning, unsupervised learning, and reinforcement learning (RL). In supervised learning, the given data set consists of both inputs and outputs so that the algorithm searches for a model between the two, whereas in unsupervised learning, no output information is provided and the algorithm tries to find out the hidden patterns in the data set [52, 53]. RL, on the other hand, is reward-oriented, meaning that its primary focus is to obtain the best action in a given environment that maximises the reward. As such, a virtual agent takes various actions and evaluates the resulting reward, which makes it learn from previous experiences and take more informed actions in the future [52, 54, 55]. Therefore, supervised learning is mostly used for classification and future predictions, unsupervised learning is a good candidate for clustering applications, while RL is often employed for optimisation or online learning purposes [53].

With its proven capabilities and advances in computing technologies, ML has already been applied in various domains including finance [56], agriculture [57], healthcare [58], etc. In addition, ML has also been a research focus for cellular communication networks, especially in the concept of self-organising networks (SON) with its broad range of applications [3, 52], such as mobility management [43, 59], dynamic spectrum access [60], and energy efficiency [61].

The summary of why ML is attracting more interest and why it is advantageous is as follows:

- **ML can deal with complex scenarios [37, 51, 62]:** 5G is expected to encompass various communications types and scenarios with distinctive requirements. While higher reliability and lower latency are the focuses for

URLLC applications, eMMB scenario would be looking for enormous radio resources. Besides, heterogeneous networks (HetNets), where different types of SCs like femto and pico cell are deployed along with traditional MCs, are forecast to be an integral part of 5G networks [38]. Thus, the network will be more complex with different type of BSs and devices; e.g., IoT devices, mobile user handsets, or vehicles, which makes the network optimisation a challenging task. ML algorithms are capable of providing reliable solutions for these kind of complex issues, which require more non-linearity as well as less human-intervention [37].

- **ML is more efficient in utilising huge amount of data [52,53]:** With the increasing number of network components including BSs and UEs, there will be a huge amount of data generated, particularly from IoT devices [39]. The data generated would be very beneficial in order to make the networks more efficient in terms of various aspects, such as cost, energy, resource, etc., thereby ML is required to be able to obtain the structure and hidden patters in the data. In other words, with the inclusion of ML, the collected data is going to be more meaningful and better utilised. Moreover, since there will different data sources with different data generation frequencies, big data analytics would also be needed in the process to deal with all these issues [52].

- **ML is able to cope with dynamic environments with its agility and adaptability [52, 53]:** Cellular communication networks are quite dynamic, which makes the network optimisation more challenging. It is not only wireless channel characteristics that change rapidly, but many other network parameters. The traffic loads of a BS, for example, show different patterns for different times and dates [63], and thus the optimisation approaches also have to be dynamic and adaptive. ML algorithms provide efficient optimisation processes in a sense that they do not output a fixed model, instead they can update their models according to the variation in network conditions. When they are implemented in an online fashion, in which ML algorithms keep training themselves with the newly generated data, they are able to adapt and so is the network optimisation. Hence, they provide dynamism, adaptability, and agility to the cellular networks.

- **Decentralised solutions can be provided with ML [53, 62]:** Decentralised (or distributed) solutions are one of the characteristics of SON, which is considered as a promising enabler for 5G networks [3]. In 5G, centralised approaches would be infeasible due to the amount of data collected and generated, which makes the wireless network management both time consuming and costly. As such, distributed decision making becomes a crucial paradigm, where local entities, such as BSs and even UEs, perform decisions and data processing with light-weight local ML implementations.

Towards that end, as illustrated in Fig. 1.2, the concept of cognitive networking is adopted in thesis, and ML is an important tool orchestrating the main characteristics of cognitive networking, such that:

- ML helps in learning from the generated data through continuous network monitoring;
- the experience is obtained with all the ML techniques implemented in this thesis, but RL has a special place in this given that it interacts with the environment and learns the consequences of the actions taken;
- ML is capable of adapting changes in the environment, such that when the network conditions change, instead of starting the optimisation process from scratch, ML algorithms can simply update the model they created for previous conditions;
- ML is a good tool to take the advantage of the large amount of data generated by cellular networks—through sensing/monitoring.

1.3 Objectives

The primary objective of this thesis is to provide intelligent, dynamic, agile, and scalable solutions for the problems within cellular communication networks. For this reason, the concept of cognitive networking is considered as a generic methodology, as it offers great potentials including learning from data, gaining experience, and adapting to changing conditions. ML is identified as a strong and applicable tool that can enhance the capabilities of cellular networks by reducing inefficiencies and redundancies in their processes. Various types of ML algorithms in its diverse branches, such as supervised, unsupervised, and RL, are employed in this thesis for different purposes, since each use-case and problem may require unique approaches.

In particular, three key design challenges in cellular communication networks, namely energy optimisation, network capacity enhancement, and spectrum access, are identified. First, as also mentioned in Sections 1.2.1, energy consumption in cellular networks has been a growing issue, and is expected to become even more complicated for 5G networks due to the increasing number of both BSs and

UEs—especially with the inclusion of IoT devices. For this reason, this thesis aims at minimising the energy consumption of cellular networks. Second, as discussed in Section 1.2.2, network capacity enhancement is one of the main design challenges for cellular networks—especially for 5G—owing to the ever-increasing demand in data rates and stringent requirements of new applications. Thus, this thesis focuses on increasing network capacity not only by abolishing redundant steps in mobile network management but also by opening up new dimensions, such as UAV assistance. Third, as also explained in Section 1.2.3, spectrum access is an important point to consider while designing cellular communication networks. Therefore, the objective of this thesis is to make CR networks more efficient, which will assist cellular networks in providing more spectrum to their users.

However, before examining all these developed methods for the three main categories, one of the main objectives of this thesis is to present a state-of-the-art review on each category thoroughly along with some brief background information. This helps in understanding the current state of the corresponding research activities, and how this thesis fills in the research gaps existing in the literature.

Based on that, the objectives of this thesis can be summarised as follows:

- Identify energy optimisation, network capacity enhancement, and spectrum access as important design challenges for future cellular networks including 5G;
- Provide an extensive state-of-the-art in each of the three categories in addition to brief introductions;
- Develop RL based methods separately for network and UE to decrease their energy consumption without compromising on user requirements or network throughput;
- Eliminate some HO steps by performing them in advance with the help of predictive HO management that employs Markov chains as a predictor;
- Investigate the effects of positioning approaches for UAV deployments in UDNs, and develop a k-means algorithm based UAV positioning technique;
- Improve the spectrum sensing phase of CR process in order to solve the spectrum access issues, and analyse the impacts of ML implementations including unsupervised, supervised, and RL.
- Discuss future trends on the areas of energy optimisation, capacity enhancement, and spectrum access.

1.4 Research Contributions

The primary contribution arising from this thesis is to introduce cognitive networking[3] in next generations of cellular communication networks. The generic framework is shown in Fig. 1.2, where the concept of cognitive networking is presented with its distinctive characteristics, such as continuous monitoring, learning from data, obtaining experience, taking informed actions, and adapting to the changes in the environment. Towards that end, after realising the demands from cellular networks, energy optimisation, capacity enhancement, and spectrum access are identified as key design challenges, followed by developing corresponding solutions separately for each design challenge with the concept of cognitive networking. In that regard, combatting the identified challenges through network intelligence—included in cognitive networking concept—constitutes the main idea behind this thesis, since the developed solutions will be made adaptive and dynamic owing to learning and experience gaining.

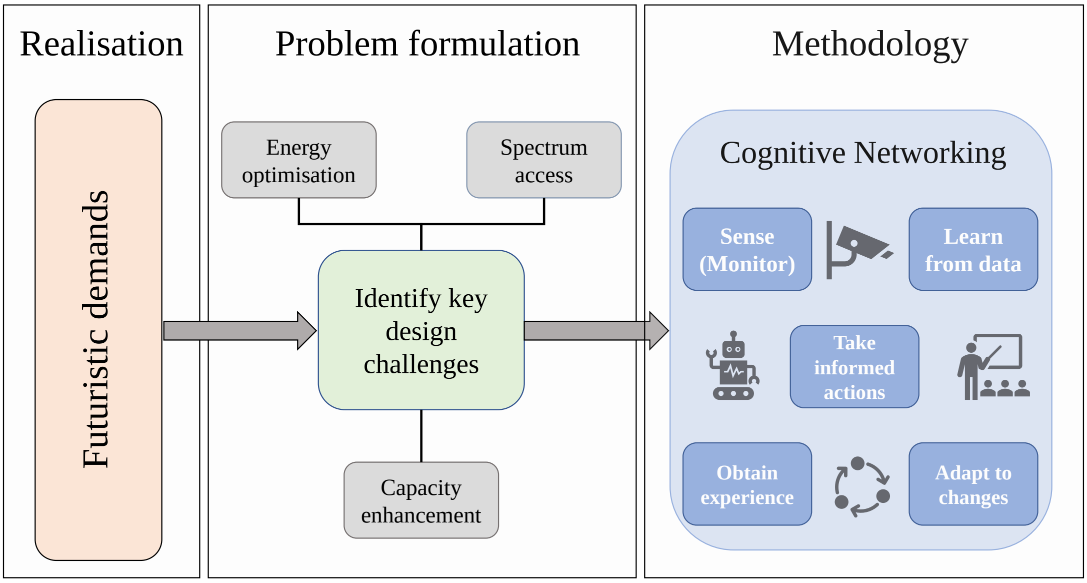


Figure 1.2: The overview of the research contributions provided in the thesis.

More specific research contributions for each design challenge are elaborated in the following paragraphs:

- **Energy Optimisation**
  - **From device perspective:** a context-aware framework that jointly optimises the connectivity and computational speed of IoT networks is presented in order to deliver the qualities required by each vertical.

[3] The concept of cognitive networking will be discussed thoroughly in Chapter 2.

Based on a smart port application, energy consumption, security, and response time are identified as essential quality features and a wireless realisation of IoT connectivity is considered using short-range and long-range technologies. An RL technique is proposed, which demonstrates a significant reduction in energy consumption while meeting the quality requirements of all related applications.

  - **From network perspective:** an RL based cell switching algorithm is developed to minimise the energy consumption in UDNs without compromising the QoS experienced by the users. In this regard, the proposed algorithm can intelligently learn which SCs to turn off at any given time based on the traffic load of the SCs and the MC. To validate the idea, the open call detail record (CDR) data set from the city of Milan, Italy is used, and the algorithm is tested against typical operational benchmark solutions. With the obtained results, it is demonstrated exactly when and how the proposed algorithm can provide energy savings, and moreover how this happens without reducing QoS of users. Most importantly, it is shown that the solution has a very similar performance to the exhaustive search, with the advantage of being scalable and less complex.

- **Network Capacity Enhancement**

  - **Predictive HO management:** two main issues with Markov chains based HO prediction are identified. First, revisits—which are defined as a situation whereby a user visits the same cell more than once within the same day—are found to be affecting the prediction performance significantly. This is addressed by incorporating HO orders into the Markov chains based prediction algorithm as an additional contextual information, which is enabled by storing a 3-D transition matrix. Second, the prediction criterion of holding the highest transition probability is found to be insufficient, especially for the cases with higher numbers of possible transition states. In this regard, an analytical model that analyses and determines the required confidence level for predictive methods is derived in terms of HO signalling cost, such that the predictive HO procedure is triggered only if the derived confidence level is met. The results indicate that the proposed solutions improve the prediction performance of Markov chains based HO predictors by advancing their confidence level via the additional contextual information and prediction criterion.

  - **UAV positioning:** a solution is developed to determine the optimal 3-D position of multiple UAVs in a capacity enhancement use-case, or in other words, when the ground network cannot cope with the user traffic demand. For this scenario, real CDR data from the city of Milan, provided by Telecom Italia is utilised to simulate an event. Based on that, a solution based on k-means clustering algorithm, an ML technique, is proposed to position multiple UAVs, and it is compared with two other baseline methods. Furthermore, the impacts of overlapping footprints of the deployed UAVs—controlled by their altitudes—on the received signal-to-interference-plus-noise ratio (SINR) are thoroughly analysed. Results demonstrate that, at certain scenarios, the proposed solution is able to significantly outperform other methods in terms of users covered and QoS.

- **Dynamic Spectrum Access**
  A comprehensive and novel solution is proposed to decrease the sensing latency and to make CR networks aware of unlicensed user requirements. As such, a proactive process with a novel QoS-based optimisation phase is proposed, paving the way for redesigning the spectrum sensing step. Initially, future traffic loads of the different radio access technologies (RATs), occupying different bands of the spectrum, are predicted using ANNs. Based on these predictions, two strategies are proposed. In the first one, which solely focuses on latency, a virtual wideband (WB) sensing approach is developed, where predicted relative traffic loads in WB are exploited to enable narrowband (NB) sensing. The second one, based on Q-learning, focuses not only on minimising the sensing latency but also on satisfying other user requirements. The results reveal that the sensing latency is dramatically reduced while significantly enhancing the full-satisfaction, which happens when all the user requirements are satisfied simultaneously.

In addition to these main research contributions, I have also performed some other works included in the List of Publications. A conventional Q-learning algorithm is employed in (Conference Proceedings 3, published in [63]) to decide which SCs to turn off under the assumption of control/data separated architecture (CDSA). Similarly, in (Conference Proceedings 1, published in [64]), an RL based cell switching is developed with the focus of mitigating CO_2 emissions. In (Conference Proceedings 4, published in [65]), a motion sensor based cell switching algorithm is developed and experimentally tested in an LTE test-bed. The main idea behind the algorithm is to detect the indoor user presence to schedule the switching for indoor SCs. Coverage analysis is carried out in (Conference Proceedings 7, published in [66]) for mmWave communication network scenario, where indoor and outdoor mmWave BSs coexist. A new HO management algorithm is presented in (Conference Proceedings 8, published in [44]), where the HO parameters are gathered from a sparse network in order to utilise when the network is densified with the objective of reducing the number of HOs. In (Journal 3), a deep RL based HO skipping approach is introduced—wherein the RL algorithm is trained for a user trajectory to decide either to HO or not to a candidate BS by considering the QoS and number of HOs. Lastly, a spectrum cost optimisation for CR networks, which considers leasing cost, latency, and QoS, is investigated in (Conference Proceedings 6, published in [67]) using ANN.

1.5 Thesis Organisation

The rest of this thesis is organised as follows. A background information on ML along with literature reviews for energy optimisation, network capacity enhancement, and dynamic spectrum access are provided in Chapter 2. After that, a research gap analysis is presented for each of the aforementioned design categories followed by discussions on how this thesis fills in the identified research gaps. Chapter 3 focuses on energy optimisation in cellular networks, which is divided into two parts; one for device-side energy optimisation and another for network-side energy optimisation. Starting from the device-side, a smart-port scenario, which includes connected IoT devices, is demonstrated followed by discussing the system modelling and the proposed methodology. After analysing the obtained results, the Chapter continues with the network-side energy consumption framework, where an RL based cell switching approach is introduced after a succinct introduction and literature review. After completing the discussion on the system model, performance evaluation, and result analysis, the Chapter is concluded with a brief summary.

Network capacity enhancement is examined in Chapter 4, which is also divided in two sections. In the first section, the predictive HO management that uses Markov chains as a predictor is introduced and its scenario and performance evaluation are thoroughly discussed. The UAV deployment is studied in the second part of Chapter 4, where the k-means algorithm based UAV positioning is proposed and extensively evaluated with SINR calculations for various cases. After comparing the proposed method with benchmark techniques using the obtained results, Chapter 4 is summarised by highlighting the main aspects of network capacity enhancement performed in this thesis. Chapter 5 elaborates the proposed dynamic spectrum access model. It starts with a problem definition followed by a brief literature review. Then, the system modelling for the CR spectrum sensing is presented. After that, the methodology, which comprises the proposed k-means algorithm based clustering, ANN based data traffic prediction, and Q-learning based decision making, is detailed. Before summarising the chapter with concluding remarks, the proposed method is evaluated thoroughly in terms of sensing latency and user requirement satisfaction. Lastly, in Chapter 6, drawn conclusions from each research chapters are provided. Moreover, future trends are identified individually for each design challenge addressed in this thesis (e.g., energy optimisation, capacity enhancement, and spectrum access) followed by mentioning existing challenges, open issues, and future research directions.

Chapter 2

Background and Literature Review

In this chapter, the literature regarding each of the aforementioned three key design challenges (energy optimisation, capacity enhancement, and dynamic spectrum access) will be thoroughly reviewed. Moreover, Table 2.1 is also presented in order to provide a summary of the literature review performed in this chapter. Note that Section 2.3.1 is a brief reproduction of the book chapter in [68].

2.1 Cognitive Networking for Cellular Communication

Thomas, et al. suggested a standard definition for cognitive networking in [69] as *"a cognitive network has a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals."*, which matches with the cognitive loop introduced by Mitola III: observe, orient, plan, decide, act, and learn [70]. Therefore, it is quite self-explanatory from the definition in [69] and the cognitive circle in [70] that a network is supposed to sense (observe/perceive) the given environment in order to understand the current circumstances, which is followed by an orientation via evaluating/analysing the environmental information obtained. The plan phase can be bypassed based on the urgency of the sensed information, but otherwise, alternative plans are prepared if it is a regular information. Lastly, one of the alternative plans is chosen and the agent takes an action by executing the plan. The sensed information from the environment of interest also feeds the learning phase in order to make more informed decisions in the future with the help of experience.

In this regard, the idea of cognitive networking aims at bringing network automation to wireless communication systems through introducing self-control in order to make the systems more dynamic, agile, and efficient. In other words, since the primary goal is to minimise the human intervention from the communication networks, the response time to any changes and/or faults in the systems is shortened, improving the dynamism and agility of the networks. Moreover, provided that human interventions render the management of communication systems more time consuming and costly, reducing it would result in utilising the resources more efficiently. On the other hand, this is also beneficial for cellular network operators, as their operational expenditures (OPEX) and CAPEX are reduced, which in turn make their businesses more sustainable and profitable [3,52].

It is reported in [71] that cognitive networking has characteristics of self-configuration, self-optimisation, self-healing, and self-protecting, the first three of which are in line with the phases of SON discussed in [3]. Since self-protecting could somehow be included in self-optimisation and self-healing, the focus of this thesis will be the first there characteristics, which are common between cognitive networking [71] and SON [3] concepts.

2.1.1 Self-Configuration

From the network automation point of view, self-configuration is the process of configuring the initial parameters of BSs, which include IP settings, antenna radiation patterns, and neighbour lists, etc., whenever there is new BS deployments and/or changes in the network [3]. This process has a crucial importance when it comes to time and cost efficiency, since human intervention—which is the case in conventional networks—incurs monetary and time costs to cellular network operators. Besides, since engineers or technicians more prone to commit mistakes and errors due to the increased complexity of future cellular networks [3], these costly configuration processes might be repeated multiple times, amplifying the costs and undermining the sustainable business of operators.

2.1.2 Self-Optimisation

After the initial configuration, a continuous monitoring of the system parameters is required to keep the communication system efficient and error-free. At this point, the concept of self-optimisation plays a vital role by providing autonomous network optimisation without—or minimal—external intervention.

Table 2.1: The summary of the literature review

<table>
<tr><td colspan="12"><b>ENERGY OPTIMISATION</b></td></tr>
<tr><td colspan="8">Life-time Extension for Wireless Sensors</td><td colspan="4" rowspan="3">Cell Switching</td></tr>
<tr><td colspan="5">Energy Harvesting</td><td colspan="3">Energy conservation</td></tr>
<tr><td>Solar</td><td>Wind</td><td>RF</td><td>Piezoelectric</td><td>Thermal</td><td>Duty-cycling</td><td>Data-driven</td><td>Mobility-based</td></tr>
<tr><td>[72–74]</td><td>[75]</td><td>[76]</td><td>[77]</td><td>[78]</td><td>[79]</td><td>[80]</td><td>[81]</td><td colspan="4">[65, 82–85]</td></tr>
<tr><td colspan="12"><b>CAPACITY ENHANCEMENT</b></td></tr>
<tr><td colspan="6">Predictive Mobility Management</td><td colspan="6">UAV Assistance</td></tr>
<tr><td colspan="6">[86–90]</td><td colspan="6">[91–94]</td></tr>
<tr><td colspan="12"><b>DYNAMIC SPECTRUM ACCESS</b></td></tr>
<tr><td colspan="4">Narrowband Spectrum Sensing</td><td colspan="7">Wideband Sensing</td><td>Predictive Sensing</td></tr>
<tr><td rowspan="2">Matched Filter</td><td rowspan="2">Energy detector</td><td rowspan="2">Cyclostationary detection</td><td rowspan="2">Covariance-based detection</td><td colspan="4">Nyquist-based</td><td colspan="2">Sub-Nyquist-based</td><td rowspan="3">[95–101]</td></tr>
<tr><td colspan="2">Wavelength transform</td><td colspan="2">Multiband</td><td>Filter-bank</td><td>Compressive sensing</td><td>Multichannel</td></tr>
<tr><td>[102]</td><td>[103]</td><td>[104]</td><td>[105]</td><td colspan="2">[106]</td><td colspan="2">[107]</td><td>[108]</td><td>[109]</td><td>[110]</td></tr>
</table>

For example, HO management, energy efficiency, radio resource management, and load balancing are some of the domains that necessitate self-optimisation [3, 52]. Given that wireless communication networks are quite dynamic due to environmental (e.g., weather, building constructions, etc.) and behavioural (e.g., different behaviours of users at different date and time) changes, the optimisation is a never-sleeping process. However, the important point here is that while time consuming conventional approaches would render the network optimisation task highly inefficient, it could be made a lot easier and very efficient via automatic action sets.

2.1.3 Self-Healing

Self-healing—offering a proactive fault detection in cellular networks alongside an autonomous repair for the identified issues—is a critical phase in SON. In conventional network management, the detection of failures and outages in the systems mostly rely on customer complaints [3,111], such that once a complaint is received, engineers or technicians need to visit the corresponding site to identify and fix the problem. On one hand, from the users' perspective, they might be left out of service or with a service with reduced quality for a while, since both detection and fixation take time. From the operator's point of view, on the other hand, this could harm their businesses, as the cite visits by skilled human resources is costly, and the unsatisfied users might switch to another operator. Therefore, by utilising self-healing, cellular networks become capable of autonomously detecting and fixing the issues immediately after they occurred, which subsequently mitigate the challenge by decreasing: 1) the dissatisfaction of the users through quick and accurate actions, and 2) OPEX of the operators via eliminating the human expert intervention [3, 111].

2.2 Machine Learning

The definition of ML in Cambridge Dictionary[1] is as follows: *"the process of computers changing the way they carry out tasks by learning from new data, without a human being needing to give instructions in the form of a program."*. There are two important takeaways from this description: 1) learning from new data, and 2) without a human being needing to give instructions. In fact, the latter puts a condition for the former; the machine is supposed to learn from the given data, meaning that it should develop a model that reveals the patterns in the

[1] Cambridge Online Dictionary: https://dictionary.cambridge.org, accessed on 25/10/2019.

data, however, this learning process has to be instruction-free. In conventional algorithm development, a set of instructions would be normally needed to explicitly program a task so that the algorithm does not take unpredictable actions. In other words, all the possible scenarios are supposed to be considered and the algorithm should be programmed accordingly in order to prevent any type of strange behaviour. In contrast, in ML, a data set is given to the algorithm, but it is not explicitly programmed with the instructions; instead, it is expected to understand the underlying patters/dynamics of the data set [112, 113].

Another interesting point worth mentioning regarding the definition of ML is *"changing the way they carry out tasks."* This part of the definition refers to how the learning is processed: when the data set is given to an ML algorithm, it starts processing each sample and develop a model, which can represent or approximate the data. However, the learning is not performed in a way that the algorithm takes all the samples at once and develops a single model at the end. The process rather takes samples and develops an initial model, which is then kept updated when new data is input to the algorithm. As such, with each new data inclusion, the algorithm evaluates the model it developed and changes it according to the outcomes of the evaluation, and this process is regarded as training [37, 52, 112, 113].

As already discussed in Section 1.2.4, ML has proven capabilities in numerous domains from finance [56] to healthcare [58]. In addition, it has been widely used in wireless communication networks for optimisation purposes through various design parameters, such as mobility management [43, 59] and energy efficiency [61]. The reason behind this popularity is that ML offers a dynamic and effective management, which are crucial characteristics especially for 5G owing to the dramatically increasing number of devices and data traffic that makes the conventional approaches no more applicable.

The main three subcategories in ML, namely: supervised learning, unsupervised learning, and RL, will be individually elaborated in the following paragraphs.

2.2.1 Supervised Learning

Supervised learning is a type of ML, where the data set consists of input and outputs—also referred to as labels. More specifically, a supervised learning algorithm is given inputs along with outputs in the training phase, and it is expected to develop a model that gives the relationship between them [112]. Then, in the testing phase the algorithm is only given new input data—which it has never seen during its training—and predicts the associated output. The performance of the algorithm is evaluated by measuring how accurately it predicts; i.e., the actual and predicted outputs are compared[2].

There are two major types of supervised learning: classification and regression. In classification, the outputs, which are given to the algorithms along with the inputs, consist of multiple classes, where each input is matched to an output class. Therefore, in the training phase, the algorithm tries to find the correlation between the inputs and the output classes, so that when it is given a new data, it can determine which class it is in. Object recognition is one of the typical examples for this type of learning [112], where the algorithm is trained with photos of different objects, such as clothes, mugs, and books, and their accurate labels. For regression, on the other hand, the predicted output is continuous values instead of classes (discrete values). One good example could be determining the price of a laptop computer: while brand, processor speed, memory size, etc. are possible input features, outputs can be the prices of the computers. Hence, the algorithm can be trained with the given inputs and outputs in order to predict the price of a new computer when it is given the aforementioned input features.

Even though there are numerous supervised learning algorithms—such as support vector machines, decision tree, and k-nearest neighbour—artificial neural networks (ANN) has been widely used due to its strong capabilities including non-linearity, adaptivity, evidential response, etc. [114]. Thus, ANN is mainly used as a supervised learning algorithm in this thesis, and it will be discussed in detail in the following paragraphs.

Artificial Neural Networks

ANN is an ML algorithm that tries to mimic a biological brain, such that it includes artificial neurons—which are inspired by the neurons in a biological brain—in its structure [51, 112, 114]. Moreover, there are connections between the artificial neurons, imitating the synapses that conveys signals among the neurons. Fig. 2.1 demonstrates a typical four-layered ANN, which is composed of an input layer, two consecutive hidden layers (HLs), and an output layer. Note that having two HLs in the network is not a must, instead the number of HLs depends on the given data and problem.

Fig. 2.1 also reveals that, apart from the input layer, each layer is connected with the preceding one. Moreover, each layer has multiple neurons, which are connected to the neurons in the subsequent layer. Then, all the neurons from the

[2] The prediction accuracy can be measured during both training and testing phases, but they would serve for different purposes. For example, the case of high training accuracy with low testing accuracy is a sign of over-fitting.

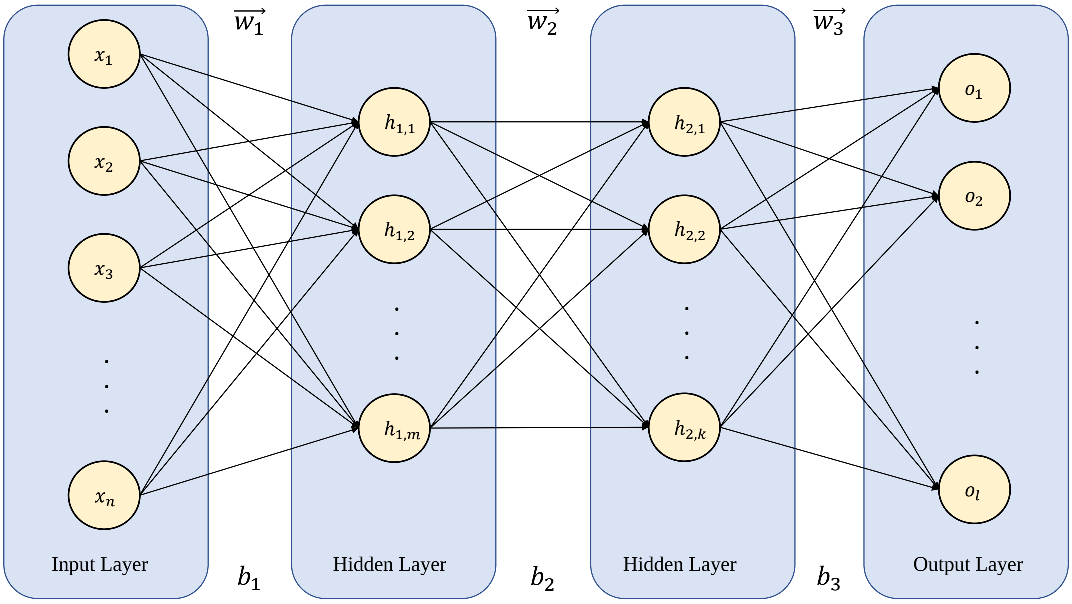


Figure 2.1: A typical ANN structure with two HLs.

preceding layer determines the values of the neurons in the subsequent layer. For $h_{1,1}$ neuron in the first HL, for example, its value is defined as [114]

$$h_{1,1} = \iota\bigg(\sum_{i=1}^{m} x_i w_{1,i} + b_1\bigg), \tag{2.1}$$

where $\iota(\cdot)$ is the activation function, b_1 is the bias value between the input and the first HLs, and $w_{1,i}$ is the ith value of weight vector between the input and first HL, $\overrightarrow{w_1}$. The rule given in (2.1) can be generalised to any neuron in any layer that has a preceding layer. While the activation function, $\iota(\cdot)$, is used to limit the output of a neuron, the bias, b, is used to adjust the input to the activation function, which in turn changes the resulting output [114, 115].

There are two main types of artificial neurons based on the activation function employed: perceptron and sigmoid [115]. The former is a primitive type of neuron, whose inputs and output are both binary, whereas the latter is more advanced type of activation function—which is used in many modern applications—and it accepts not only binary values as an input but also the values in between. Similarly, unlike perceptron, the sigmoid type of neuron can produce output values between 0 and 1 [114, 115].

In the forward pass, the information is propagated from the input layer towards the output layer; however, back-propagation (BP), which goes from the output layer towards the input layer, is also involved in the process in order to enable the network to learn its parameters. First, when the forward pass is performed, the predicted value is compared with the real output value, and the difference between the real and predicted values—also referred to as cost—is calculated through any type of cost functions, such as mean squared error (MSE). After that, the error is back-propagated in order to update the weights and bias in each individual layer via an optimisation method, like gradient descent (GD). This process is repeated multiple times in order to minimise the cost calculated at the output.

The number of HLs, number of neurons in each HL, cost function, and optimisation method are referred to as the hyper-parameters of an ANN, and play a crucial role in the performance of the network. Similar to many ML algorithms, there are two primary concepts hindering the good performance of ANN: under-fitting and over-fitting [51, 112, 114]. The latter means that, during training, the developed model cannot approximate the actual relationship between the input and output data, and thus underperforms when it is given a new data set. The former, on the other hand, occurs when the model almost exactly matches with the data given in the training phase, which subsequently results in remembering the training samples, thereby the algorithm cannot perform well with the new data set. The aforementioned hyper-parameters can help alleviate these issues. For example, on one hand, if the number of HLs is too few, then the ANN algorithm becomes incapable of developing a good model that approximates to the actual relationship hidden in the given data. On the other hand, if the number of HLs is too high, then the algorithm becomes too complex and produces a model accordingly. Nevertheless, since it is trained with only some portion of the whole data (training samples), the model becomes specific to the training data set, or in other words, the developed model is unable to generalise.

In addition to the proper adjustments in hyper-parameters, there are some other ways to avoid the over-fitting problem including early stopping, regularisation, etc. [112, 115]. With cross validation, the training data set is split into some portions of training and validation, and then the model is tested through the validation set during the training. Then, the performance of the algorithm is monitored on both training and validation set, and the iterations are stopped once the error in the validation set starts increasing, even if the performance on the training set is improving. This process is called early stopping [116].

There are various regularisation methods, such as L1, L2, and dropout. In L2 regularisation, for example, an extra regularisation parameter is appended into the cost function in order to prevent the network from learning large weights. This is because a noise in the data could be significant with large weights, whereas small weights could make it less effective [115]. Thus, a regularised ANN algorithm will try to minimise both the overall cost and the network weights.

2.2.2 Unsupervised Learning

Unlike the case in supervised learning, where the data includes both input and output, the data set given to an unsupervised learning algorithm only consists of inputs. Instead, the algorithm is expected to reveal hidden patterns, regularities, or properties in a given input data [112, 117]. A good example for this type of learning is clustering, where the data is clustered into some groups based on their key features, and the unsupervised learning algorithm attempts to distinguish the clusters from each other with certain boundaries. However, unlike supervised learning, the identification of the clusters are not expected[3] after the clustering is performed.

Clustering finds a broad area of applications from customer segmentation [118] to agriculture [119], but it has also an important place in wireless communication networks. For instance, in wireless sensor networks (WSNs), clustering plays a vital role in detecting anomalies in a network [120]. Furthermore, unmanned aerial vehicles (UAVs) have gained significant attention in cellular networks to enhance the network capacity and to help in disaster scenarios [94]. In this regard, clustering can assist in positioning the UAVs, for example, in order to maximise the coverage or the number of users connected [121].

There are great number of unsupervised learning algorithms in the literature, such as self-organising maps and hierarchical clustering, but this Chapter will focus merely on k-means clustering algorithm provided that it is primarily used as an unsupervised learning algorithm throughout the thesis.

k-means Clustering

k-means algorithm is one of the most popular clustering algorithms due to its simplicity and property of guaranteeing the convergence to a point, where all the data samples are associated with a cluster [117]. In other words, it does not need any external stopping criteria, since its inherent working principle already provides it with a criterion to stop the execution [117]; it automatically stops at a point where the centroids keep their same position and no longer move [122].

k-means is an algorithm attempting to discover k different clusters in a data set with various samples iteratively. For each cluster, there is a dedicated centroid [123]. The basic idea behind this algorithm is to place these centroids and

[3] In supervised learning, the classes are labelled; for example, fruit, vegetable, etc. However, in unsupervised learning, there is no label for the clusters.

associate the closest data points to them. In the learning phase, in order to find an optimum clustering, the positions of the centroids are altered by the average value of the associated data points. The process of the k-means algorithm is shown in Algorithm 1 [117, 122, 123], where N_{dp} is the total number of data points; $N_{\text{dp},j}$ is the number of data points in the cluster c_j; z_i, $i \in \{1, 2, ..., N_{\text{dp}}\}$ is the data points in cluster j; and c_j is the centroid of cluster j, $j \in \{1, 2, .., k\}$. Note that the error for k-means clustering is defined as the Euclidean distance

```
Algorithm 1: $k$-means Algorithm
   Data: $k$, data points $(z_1, z_2, ..., z_{N_{\text{dp}}})$
   Result: Clusters
 1 Initialise with a random placements of the centroids $(c_1, c_2, ..., c_k)$;
 2 while cluster assignments change do
 3     for each data point $(z_i)$ do
 4         Determine the closest centroid, $c_j$, through:
               $\arg\min_{j} D(z_i, c_j)$        (2.2)
 5     end
 6     for each cluster $(j \in \{1, 2, ..., k\})$ do
 7         Place centroid $c_j$ to the mean of data points associated with $c_j$
           via:
               $c_j = \frac{1}{N_{\text{dp},j}} \sum_{l} z_l \quad | \{l = i \iff z_i \in c_j\}$        (2.3)
           Go back 3;
 8     end
 9 end
```

between the centroid and its associated data points given by [122, 123]:

$$d_{k\text{-means}}(c_j) = \sum_{z_i \in c_j} ||z_i - \mu_j||^2, \tag{2.4}$$

where μ_j is the mean of cluster j.

Determining the number of clusters is one of the main issues for clustering problems. Intuitively, a small number of clusters gives huge errors, while smaller errors can be obtained with a larger number of clusters [122]. In extreme cases, for example, if the number of clusters equals to 1, then all the samples will belong to the same cluster, making the error enormous. On the other hand, if the number of clusters equals to the number of samples, there will be no error at all as all the samples will be associated to a different cluster [122].

2.2.3 Reinforcement Learning

RL is a distinctive class of ML, since it has inherit properties that distinguish it from supervised and unsupervised learning techniques. First of all, interacting with the environment-of-interest is the way of learning followed in RL, whereas the learning is performed through labelled/unlabelled data in supervised/unsupervised learning [53, 54]. In supervised learning, a labelled data is fed into an algorithm—which is expected to map the given inputs and outputs accordingly—, while unsupervised learning accepts only inputs and tries to explore the hidden patterns and structures. On the other hand, in RL, the agent is expected to take actions by interacting with a given environment and evaluate the corresponding reward, which is aimed to be maximised [54]. As reported in [54], there are four main elements that need to be discussed under RL, namely policy, reward (or penalty/cost), value function, and model. However, before elaborating each of these elements, it is better to define some preliminary concepts:

- **Environment:** this can be anything that produces an output for any taken action. For the case of cellular communication, for example, the environment could be the wireless network that consists of BSs and UEs with various channel conditions.

- **Agent:** this is the entity that takes actions in the given environment. In cellular networks, this could be a BS, which tries to optimise its HO parameters in order to alleviate its traffic burden without affecting its associated users significantly. Alternatively, a UE can also be an agent that aims at reducing its power consumption in order to enhance its battery life.

- **Action:** the agent takes actions in the given environment in order to observe and maximise the reward function. These actions could be a set of possible movements in the selected parameters. Taking the example of cellular networks and assuming a BS is the agent, the action set for the BS could be a range of HO margin values for the objective of load balancing. Thus, the agent takes its actions by changing its HO margin and calculates the resulting reward function.

- **State:** the condition of the agent according to the taken action. Continuing from the previous load balancing example, the agent could be in overloaded, fully-loaded, or under-loaded states based on the action taken.

A general structure of a typical RL model is demonstrated in Fig. 2.2 [54].

The aforesaid elements of RL can be described as follows:

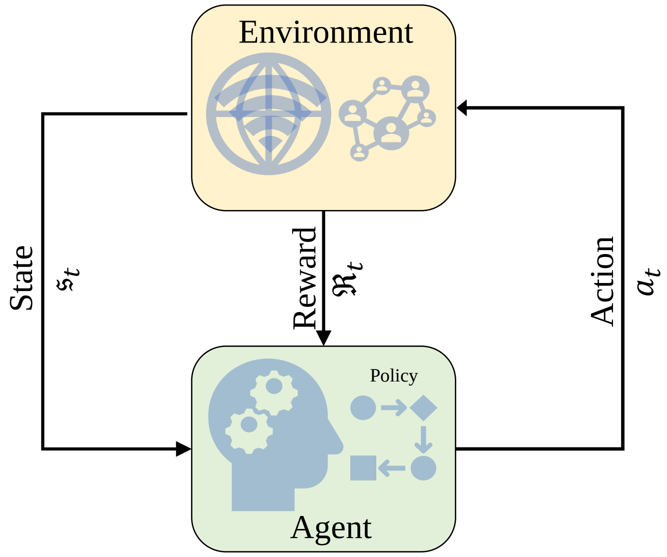


Figure 2.2: Typical RL process including environment and agent.

- **Policy (π):** determines how the actions are taken by the agent, and constitutes the interface between the actions and the states [52, 54].

- **Reward function:** a predefined function that is aimed to be maximised. From the optimisation perspective, this can be interpreted as the objective function of the problem. According to the nature of the problem at hand, the reward function can sometimes be converted to a penalty function, in which the objective becomes minimisation rather than maximisation.

- **Value function:** there are two categories in value function, namely state-value function and action-value function. In the former case, it indicates the expected value of visiting a state, whereas, in the latter case, the expected value of taking an action while being in a particular state.

- **Model:** a representation of the environment of interest, which is utilised to comprehend how the environment reacts to actions taken by the agent [54]. Note that some algorithms do not require a model, as they are executed in a model-free fashion.

Q-learning

Q-learning is a model-free method, meaning that, unlike model-based RL methods, it does not require a model of the environment in advance. Instead, it interacts with the environment through an empirical approach in order to gain experiences [54]. Moreover, Q-learning is also an off-policy method, in which the selection of the next state follows an ε-greedy policy (π_ε), where $\varepsilon > 0$, while updating the action-value function follows a greedy policy (π_g), where $\varepsilon = 0$. This property offers early convergence to Q-learning due to the simplified algorithm analysis [54].

The action-value function update policy in Q-learning for the case of reward-based framework is as follows:

$$Q(\mathfrak{s}_t, a_t) := Q(\mathfrak{s}_t, a_t) + \alpha(\mathfrak{R}_{t+1} + \varphi \max_a(Q(\mathfrak{s}_{t+1}, a)) - Q(\mathfrak{s}_t, a_t)) \tag{2.5}$$

where $\mathfrak{s}_t$ and $\mathfrak{s}_{t+1}$ are the current and next states, respectively. $\mathfrak{R}_{t+1}$ is the expected reward for the next time step, a_t is the taken action, and a is the set of possible actions. α is a learning rate while φ being a discount factor. The max function in (2.5) should be converted to a min function to make the update policy suitable for the penalty-based framework, in which the $\mathfrak{R}$ in (2.5) replaces with a penalty/cost function, $\mathfrak{C}$. The idea behind max and min functions is that the agent tries to maximise the reward for the reward-based approach, while it attempts to minimise the penalty in the penalty-based model [54].

The algorithm of Q-learning is presented in Algorithm 2 [54].

Algorithm 2: Q-learning algorithm [54]

```
Initialise $Q(\mathfrak{s}, a)$;
for each episode do
    Initialise $\mathfrak{s}$,
    for each iteration do
        Select action, $a_t$ with the $\varepsilon$-greedy policy ($\pi_\varepsilon$),
        Take action, $a_t$,
        Determine the expected numeric value of the reward function,
          $\mathfrak{R}_{t+1}$ along with the next state $\mathfrak{s}_{t+1}$,
        Update action-value function using (2.5),
        $\mathfrak{s}_t \leftarrow \mathfrak{s}_{t+1}$
    end
end
```

SARSA

SARSA, which stands for state–action–reward–state–action, is another technique in RL and follows a similar procedure with Q-learning, since they are both model-free algorithms. However, unlike off-policy Q-learning, SARSA is an on-policy method, meaning that it follows an ε-greedy (π_ε) policy for both selecting the next action and updating the action-value function, whose update policy for SARSA is as follows [54]:

$$Q(\mathfrak{s}_t, a_t) := Q(\mathfrak{s}_t, a_t) + \alpha(\mathfrak{R}_{t+1} + \varphi(Q(\mathfrak{s}_{t+1}, a_{t+1})) - Q(\mathfrak{s}_t, a_t)) \tag{2.6}$$

where a_{t+1} is the action for the next time step.

The sequence of "state–action–reward–state–action" that constitutes the name of the method is derived from the order of the implementation, which is also shown in Algorithm 3 [54].

Algorithm 3: SARSA algorithm [54]

```
Initialise $Q(\mathfrak{s}, a)$;
for each episode do
    Initialise $\mathfrak{s}$,
    for each iteration do
        Select action, $a$ with the $\varepsilon$-greedy policy ($\pi_\varepsilon$),
        Take action, $a$,
        Determine the expected numeric value of the reward function,
          $\mathfrak{R}_{t+1}$ along with the next state $\mathfrak{s}_{t+1}$,
        Update action-value function using (2.6),
        $\mathfrak{s}_t \leftarrow \mathfrak{s}_{t+1}$,
        $a_t \leftarrow a_{t+1}$
    end
end
```

Linear Value Function Approximation

Even though conventional RL algorithms, such as Q-learning and SARSA, are strong tools in decision making, they are limited from various perspectives. In both Q-learning and SARSA, there is a need to keep a lookup-table for each state and action pair. Even though this is viable solution for many cases, there are some particular scenarios, where the lookup-table approach becomes inefficient and inapplicable. For example, when the number of possible states to visit is too large, then the lookup-table will enlarge equally, which subsequently not only increases the computational power and memory consumption, but also hinders generalisation of the algorithm [54].

In value function approximation (VFA), on the other hand, instead of keeping a lookup-table for all the states and actions, it is aimed to approximate the value function through a vector of weights, $\vec{w}_Q$. In particular, the value function denoted by $Q_\pi(\mathfrak{s}, a)$, which is obtained by following a policy π, is made the subject for an approximation denoted by $\hat{Q}(\mathfrak{s}, a, \vec{w}_Q)$; such that $Q_\pi(\mathfrak{s}, a) \approx \hat{Q}(\mathfrak{s}, a, \vec{w}_Q)$ [54]. Although various methods can be used for approximation, the main objective is to minimise the error between $Q_\pi(\mathfrak{s}, a)$ and $\hat{Q}(\mathfrak{s}, a, \vec{w}_Q)$, such that

$$\min_{\vec{w}_Q} \; [Q_\pi(\mathfrak{s}, a) - \hat{Q}(\mathfrak{s}, a, \vec{w}_Q)]^2. \tag{2.7}$$

Algorithm 4: SARSA with VFA [54, 124]

```
for Each episode do
    Initialise the current state, $\mathfrak{s}_t$;
    for Each action do
        Get the features, $\vec{\mathfrak{X}}$;
        Estimate the value of $Q$ through (2.9);
    end
    Pick the action, $a_t$, according to a policy;
    for Each iteration do
        Take the action $a_t$;
        Observe the reward, $\mathfrak{R}$, or penalty, $\mathfrak{C}$;
        Go to the next state;
        for Each action do
            Get features, $\vec{\mathfrak{X}}$;
            Estimate value of $Q$ through (2.9);
        end
        Pick the next action, $a_{t+1}$, according to a policy;
        Update the weights, $\vec{w}_{Q,t}$, using (2.10);
        $\mathfrak{s}_t \leftarrow \mathfrak{s}_{t+1}$;
        $a_t \leftarrow a_{t+1}$;
        Check the stopping criteria (if applicable);
    end
end
```

Moreover, $\hat{Q}(\mathfrak{s}, a, \vec{w}_Q)$ can be expressed by

$$\hat{Q}(\mathfrak{s}, a, \vec{w}_Q) = f(\vec{w}_Q), \tag{2.8}$$

where $f(\cdot)$ is a known function, which is also referred to as hypothesis.

Even though any function can be used as the hypothesis, some may render (2.8) either too hard or infeasible. Linear functions, shallow neural networks, and deep neural networks are commonly used as $f(\cdot)$, and the choice depends on the type of problem. With regards to optimisation, a popular strategy is to use GD or stochastic GD (SGD) to find $\vec{w}_Q$ based on the known examples [54].

In a linear hypothesis, $\hat{Q}(\mathfrak{s}, a, \vec{w}_Q)$ is approximated by a linear combination of input features [54], in other words

$$\hat{Q}(\mathfrak{s}, a, \vec{w}_Q) = \vec{\mathfrak{X}} \vec{w}_Q^T, \tag{2.9}$$

where $\vec{\mathfrak{X}}$ is a vector containing input features, and $\vec{w}_Q^T$ indicates $\vec{w}_Q$ transposed. Furthermore, using GD and taking the SARSA case, $\vec{w}_Q$ is updated according to [54, 124]

$$\vec{w}_{Q,t+1} = \vec{w}_{Q,t} + \alpha_{\mathrm{GD}}[\mathfrak{R}_{t+1} + \varphi Q_\pi(\mathfrak{s}_{t+1}, a_{t+1}) - \hat{Q}(\mathfrak{s}_t, a_t, \vec{w}_{Q,t})] \cdot \nabla_{\vec{w}_{Q,t}} \hat{Q}(\mathfrak{s}_t, a_t, \vec{w}_{Q,t}), \tag{2.10}$$

where α_{GD} is the learning rate for GD optimisation, and $\nabla_{\vec{w}_Q}$ represents the gradient with respect to $\vec{w}_Q$. Note that, $\vec{w}_Q$ is initialised with an arbitrary value, such as zero.

After some iterations, if adequate different examples have been observed, $\vec{w}_Q$ will converge and can be used to find the action-value function, which in turn can guide the policy. Algorithm 4 demonstrates the overall SARSA algorithm with VFA implementation [54, 124].

2.2.4 Markov Chains

Markov chain is a stochastic process that is used for time-variant probabilistic modelling. It is basically used for modelling the transition probabilities among states, which can be a condition that an agent is in. For an electronic device, for instance, the switching options can be defined as states, such that the device is in the *on* state when it is switched on and in the *off* state if it is switched off.

A formal definition of Markov chain, as described in [125], would be as follows: Let $(\mathcal{Y}, \mathcal{F}, P)$ be the probability space, where $\mathcal{Y}$, $\mathcal{F}$, and P are sample spaces that include all the possible outcomes; set of events; and measure function for the probabilities of the events, respectively. Moreover, let $(X, \mathcal{X})$ be a measurable space and $\mathcal{F}_i, i \in I$ be a filtration, where I is the index. Then, an adapted stochastic process $\{(X_i, \mathcal{F}_i), i \in I\}$ is said to be a Markov chain if

$$P(X_{i+1} \in \mathcal{G} | \mathcal{F}_i) = P(X_{i+1} \in \mathcal{G} | X_i), \tag{2.11}$$

for all $i \in I$ and $\mathcal{G} \in \mathcal{X}$ [125]. Therefore, the Markov property states that the outcome of any experiment depends only on the information available at the present time, which is i in this case.

Fig. 2.3 demonstrates transitions among three different states, where $p_{i,j}$ is the transition probability from state i to state j, and $\sum_{j=1}^{N_{\mathrm{m,s}}} p_{i,j} = 1$, where $N_{\mathrm{m,s}}$ is the number of available states. Markov chains based predictors define the transition probability from a current state to other states by building a transition matrix, which consists of transition probabilities between the current state and all other possible states. One of the main properties of Markov chains is that its

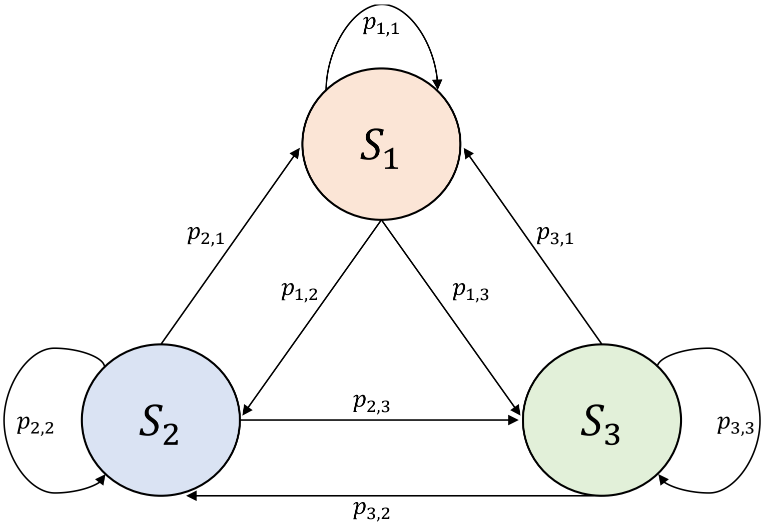


Figure 2.3: A sample Markov chain with three states, $N_{\mathrm{m,s}} = 3$.

predictions only dependent on the current state of a user. The transition matrix can then be written as

$$\vec{\mathcal{T}} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,N_{\mathrm{m,s}}} \\ \vdots & \vdots & \vdots \\ p_{N_{\mathrm{m,s}},1} & \cdots & p_{N_{\mathrm{m,s}},N_{\mathrm{m,s}}} \end{bmatrix}. \tag{2.12}$$

Then, the Markov chains based predictor makes a prediction by following a model: [126]

$$\vec{p}_m = \vec{p}_0 \vec{\mathcal{T}}^m, \tag{2.13}$$

where $\vec{p}_m$ and $\vec{p}_0$ are the probability vector belonging to the mth transition and the initial distribution vector, respectively.

2.2.5 Machine Learning Applications in Cellular Networks

ML has a very broad range of applications in cellular networks from radio resource management to energy efficiency. This is because ML has strong optimisation capabilities in addition to its adaptability in dynamic environments [51–53, 59, 62, 127]. In the system level, in particular, ML application is more common due to the fact that the problems are case-specific and it is very hard to obtain a generic solutions that can easily be adopted. Thus, the adaptability through learning makes ML a good candidate for optimising these types of problems. Moreover, in cases the optimal solutions are too complex to implement and/or not scalable, ML can help in reducing the complexity and making the solution more scalable [62, 128]. However, in order to achieve this, the ML implementation should also be efficient and well-designed, otherwise it can also contribute on the existing complexity. In addition, it is not a rule of thumb that ML implementations are always scalable; this can also be accomplished through proper and careful design. In the case of ANN, for instance, if the number of input or output neurons grows exponentially with the increase of the wireless network size, it cannot be said that the solution is scalable.

On the other hand, well-designed ML solutions have gained a considerable amount of attention [51–53, 59, 62, 127]. Various types of ML implementations in different domains are available in the literature, and some examples will be discussed in the following paragraphs. It has been proven that a significant saving on HO signalling can be achieved though predictive mobility management while reducing HO failures and improving radio resource management [59,129–134]. Even though the corresponding works may include HO predictions, location prediction, route prediction, etc., the generic purpose is to make the mobility management in cellular networks more efficient and proactive. Similarly, ML plays an important role in green cellular networking as well. In particular, cell switching mechanisms [61, 85, 135, 136], energy harvesting applications [137], energy efficient UAV positioning [138] are few examples where ML can be implemented in order to make the cellular networks greener.

Resource management in cellular networks is another vital aspect to reiterate, since the available resources in terms of bandwidth, energy, computation, and memory are limited and should be managed efficiently. In this regard, the applicability and feasibility of ML implementation has already been proven in resource management schemes. Computational power and memory management through caching [139–141], predictive spectrum sensing for better utilisation of radio resources [95, 97, 142] are good examples of how ML can help in this aspect. ML is also a viable and robust tool for anomaly detection in cellular networks. Cell outages, for example, can be challenging issues to combat, since it is hard to identify the problems, and mobile networks operators often rely on the client complaints [3, 111], which delays the repair and are not always reliable. However, through ML implementations, the identification of cell outages can be done faster and more reliable [143, 144], which in turn helps mobile network operators in reducing their OPEX, while keeping the users satisfied.

Backhauling is one of the major problems for future generations of cellular networks due to the increased data rate demand as well as the dramatic inflation in the number of connected devices. Especially for UDNs, where the intensity of the devices and BSs are comparatively much higher, the bottleneck is expected to appear more in the backhaul rather than the access network [145]. To this end, ML offers intelligent backhauling solutions, as in [146], that can eventually help resolve the issues.

In addition to these, ML has many other application domains in cellular networks, such as load balancing, coverage optimisation, neighbouring cell list management, etc. [52]. For more detailed discussions on the application of ML algorithms into wireless communication networking, there are multiple survey papers available in the literature to refer to, such as the ones in [51–53, 59, 62, 127] to mention a few.

Machine Learning Implementation for the Key Design Challenges

As already discussed in Chapter 1, the key design challenges investigated in this thesis are identified by considering the futuristic demands from cellular networks. Moreover, the strong inter-relation between these design challenges, which were already explained in Chapter 1 via Fig. 1.1, appears as another rationale behind choosing them as the primary topics.

Cognitive networking, on the other hand, is considered as the main underlying idea behind this thesis, such that in order to bring intelligence and dynamism to cellular networks, the identified problems are solved using the concept of cognitive networking. Given that cellular networks have been gaining popularity and getting more complex from one generation to the next [4, 14, 17, 26, 37, 38], cognition (intelligence) is required in order to address the ever-growing demands in more challenging scenarios.

Data is the main ingredient for intelligence, and fortunately, the amount of data generated in the next generations of cellular networks will be much higher due to: i) the prevalence of IoT devices; ii) more BS deployments with network densification; iii) more data transmitted by users; etc. [28, 29, 31, 36–38]. Learning, on the other hand, plays a crucial role in exploiting the huge amount of data, which subsequently enables cognitive networking. Therefore, as also demonstrated in Fig. 1.2, ML is employed as the fundamental methodology, thereby—along with the aforementioned inter-relations among the topics—cognitive networking, in general, and ML, in particular, create the underlying idea behind the thesis. In other words, all the problems formulated here are tackled through ML using the data generated by communication networks, and thus:

- learning from data;
- gaining experience;
- taking more informed actions;

are at the origin of this work in order to provide cellular communication networks with cognition (intelligence), tying up the considered design challenges and constituting the story of the thesis.

When to Employ Machine Learning?

Although ML algorithms often provide efficient and effective solutions for the problems within cellular networks, their implementation is not always as feasible as other methods, such as statistical analysis and heuristic algorithms. Since each problem has its own characteristics, the decision should be made according to the conditions and requirements, and there are some preliminary considerations with regards to the implementation of ML:

- properties of the data;
- big data;
- computational complexity;
- performance of ML algorithms.

As ML algorithms predominantly require data to learn from, the following issues regarding the required data are worth considering: i) availability, ii) quality, and iii) sufficiency [147,148]. If, for example, the data is not always (or not at all) available, then the implementation of an ML algorithm might be hard or even impossible. In the case of availability, on the other hand, its quality becomes the next to consider, because there could be redundancy—such as duplications—and/or noise—such as missing entries—in the data, thereby preprocessing is required before using it for a certain objective [147]. On one hand, the sufficiency of the data is a vital element, since training phases of many ML algorithms necessitate a certain amount of data to build an accurate model. When the volume of the data is huge (i.e., big data), on the other hand, other challenges take the stage including storage, security, latency, etc. [149, 150], which could potentially undermine the feasibility and practicability of ML algorithms.

After solving the data-related issues, the challenges are then transferred to the characteristics and performances of ML algorithms, such that the computational requirements and the end performances (e.g., accuracy) of the developed algorithms are the ones being at the front line to determine their applicability. Following comprehensive comparisons and analyses, the authors in [151], for instance, found that the statistical methods outperform the ML counterparts in time-series predictions with less computational requirements. This is an important observation, as it reveals that the use of ML is not a must, instead the methodological decision should be made by considering both the characteristics of a given problem and alternative solutions. Furthermore, it was concluded in [128] that ML (deep learning example is taken therein) is not always a good choice, especially for physical layer applications where optimal solutions without any significant complexity already exist. However, the authors noted that deep learning can be considered in cases, where the optimal solutions are computationally quite demanding.

Based on these, it can be concluded that the followings should be considered before choosing ML as a methodology:

- there is no issue regarding the required data in terms of availability (e.g., data collection), quality, sufficiency, etc.

- in case the data volume is large, the data management plan is supposed to be feasible, such that the side effects, including privacy, storage, latency, etc., do not dominate the gain obtained through the use of ML.

- the implementation of the algorithms is more advantageous than alternative solution: this could be in terms of various metrics including performance, computational complexity, scalability, etc.

In that regard, ML considered throughout this thesis for the following reasons:

- the scenarios are quite dynamic, thereby implementation of a non-ML algorithm would be no more applicable due to the fact that they would require repetitive implementations for changing conditions. ML, on the other hand, can adapt itself to the changes after the initial learning phase.

- optimal solutions (e.g. the one in Section 3.2) could be unscalable, meaning that they become very hard to implement when the network sizes grow.

- the data sets at hand comply with the aforementioned conditions, such that there is an availability along with sufficiency, and a minimal preprocessing is needed.

Machine Learning Tools Used for Implementation

All the three major branches of ML are implemented this thesis, namely: supervised learning, unsupervised learning, and RL. The details about the tools used for each ML implementation are as follows:

- **Supervised learning:** ANN is employed in Chapter 5, and MATLAB *Neural Net Fitting App* is used as a toolbox.

- **Unsupervised Learning:** k-means clustering is employed in both Chapter 4 (Section 4.2) and Chapter 5, and *k-Means* function within MATLAB *Statistics and Machine Learning Toolbox* is used during the implementations.

- **RL:** Two different Q-learning algorithms are implemented in Chapter 3 (Section 3.1) and Chapter 5, and they are programmed in MATLAB without using any off-the-shelf toolbox. Furthermore, the SARSA algorithm with VFA in Chapter 3 (Section 3.2) is also programmed in MATLAB without the inclusion of any existing toolbox.

2.3 Energy Optimisation in Cellular Networks

In this section, first, life-span extension techniques will be presented in two categories as energy conservation and energy harvesting. Then, typical BS components as well as the power consumption profiles of common BS types will be detailed, followed by a discussions on the cell switching concept with the state-of-the-art. The section is concluded by a research gap analysis.

2.3.1 Life-time Extension for Wireless Sensors

Energy is always the main determinant for the life-span of wireless sensors, as they typically use external batteries, which have limited capacities [68]. Moreover, even a single node can determine the life-span of a whole network if its absence affects the working routine of the system; i.e., it can be a sink node aggregating and routing the data from sensor nodes to an end-user, or a relay node, which is located at a very strategic location and relays data coming from multiple neighbouring nodes. Alternatively, it can be a regular sensor node that is acquiring very important information, which the whole system relies on. Therefore, it is crucial to improve the energy balance of each individual node to extend its life-span, which in turn prolongs the life-span of the whole network. Let the energy balance of a single node be E_{b}, such that

$$\mathsf{E}_{\mathrm{b}} = \mathsf{E}_{\mathrm{in}} - \mathsf{E}_{\mathrm{out}}, \tag{2.14}$$

where E_{in} and $\mathsf{E}_{\mathrm{out}}$ are the input and output energies, respectively [68].

It is obvious from (2.14) that the energy balance of networks can be increased by either rising the energy input or decreasing the energy output, or by doing both. Given that wireless sensors work with limited energy sources (batteries), increasing the energy input refers to using alternative energy sources (e.g., solar, wind, etc.) along with the available batteries. Methods, which enable to use such kind of alternative energy resources, are regarded as energy harvesting. On the other hand, decreasing the energy output refers to energy conservation, which implies using the available sources as efficiently as possible by avoiding any kind of energy wasting.

Energy Harvesting

Energy harvesting techniques play a crucial role in extending life-spans of WSNs, as they basically use alternative energy sources to contribute to the energy balance of networks. However, as this process is performed with ambient energy sources, the harvested energy should be converted to a form that is usable in sensor nodes [68].

Energy source, harvesting methods, and load are reported as the key elements of an energy harvesting system in [152], wherein energy sources are categorised into two as controllable and non-controllable. Controllable energy sources are the ones that can always provide energy on-request, whereas non-controllable sources, such as solar and wind, have their on routine and may not be available when needed [152, 153].

Moreover, there are two harvesting architectures available [152]: harvest-use and harvest-store-use. In the former, harvested energy is directly used in sensor nodes, while in the latter it is stored in a storage element, which is then used as an energy supplier by the sensor nodes.

The energy conversion technique to be employed depends on the type of the energy source. In the case of solar energy, for instance, solar panels, which convert the solar energy to electrical energy, should be employed. However, solar panels generate direct current (DC) power, and thus a DC-DC converter might be needed in order to make the output of the panels suitable for a load—a sensor node in this case. Therefore, the DC-DC converter is supposed to be designed in an efficient manner, since it is more likely to result in a loss during conversion. Given that the conversion efficiency of the current solar panels are already not sufficiently high, the loss in DC-DC converters makes the case worse with a negative contribution to the energy balance.

Various harvesting techniques are surveyed in both [152] and [153] in a comprehensive manner. The overall list of the surveyed techniques is as follows [68]:

- solar energy harvesting [72–74];
- wind energy harvesting [75];

- radio frequency (RF) harvesting [76];
- piezoelectric energy harvesting [77];
- thermal energy harvesting [78].

Energy Conservation

In order to decrease E_{out} in (2.14), the energy consumption of wireless sensors should be minimised to extend their life-span. Among the three main tasks (i.e., sensing, processing, and communicating) performed in a typical sensor node, communication is often the most energy consuming one [154]. However, since sensor nodes are energy-constrained devices, it is crucial to reduce the energy consumption in each phase. Energy conservation methods in sensor networks are comprehensively surveyed in [155], of which the high-level taxonomy is adopted as follows [68]:

- duty-cycling [79];
- data-driven approaches [80];
- mobility-based approaches [81].

2.3.2 Base Station Components

A BS power consumption model is created in [156], wherein the main energy consuming components of a typical BS are identified as power supply, cooling system, baseband unit, power amplifier, and antenna interface. They will be discussed individually in the following paragraphs.

- **Power supply:** the mains energy supply is used to power BSs. Since the mains supplies alternating current (AC) to BSs, this mostly needs to be converted to DC due to the fact that the vast majority of BSs use DC power rather than AC [157]. This conversion also incurs some loss, which degrades the overall energy efficiency of a BS.
- **Cooling system:** especially for the case of MC, BSs need air-conditioning systems in order to keep their system operable, since the components that are sensitive to temperature would break down when a certain temperature threshold is exceeded. The cooling system can account for up to 30% of the total energy consumption of a BS [41, 157], thereby energy saving from the cooling system would contribute significantly to the energy efficiency of the BS.

- **Baseband unit:** baseband signal processing, such as modulation, conversion, filtering, signal detection, fast Fourier transform (FFT), etc., is carried out by the baseband unit in a BS, and each of these tasks results in additional energy consumption [156]. Between 5% and 15% of the energy consumption of a BS comes from the signal processing [158], and thus it is important to reduce the energy consumption at the baseband unit.

- **Power amplifier:** this is one of the most power consuming elements in a BS, and it is used to increase the power level of RF signals during transmitting and receiving. However, there are many design challenges including improved back-off efficiency and linearity, putting the semiconductor type used in power amplifiers at a very important position [159].

- **Antenna interface:** the antenna is the main component to enable signal transmission/reception, at which it is responsible of converting the electrical current into EM waves, and vice versa. Since there are multiple losses related to the antenna interface, such as feeder loss and matching loss, the design and type of an antenna play important roles in energy efficiency [156].

2.3.3 Base Station Power Consumption Model

In the Energy Aware Radio and neTwork tecHnologies (EARTH) power consumption model [156], $\mathcal{P}_j$, the instantaneous consumption of a BS, B_j, is given by [160]

$$\mathcal{P}_j = \begin{cases} \mathcal{P}_{\mathrm{o},j} + \eta_j \Lambda_j \mathcal{P}_{\mathrm{T},j}, & \text{if} \quad 0 < \Lambda_j < 1 \\ \mathcal{P}_{\mathrm{s},j}, & \text{if} \quad \Lambda_j = 0, \end{cases} \tag{2.15}$$

where $\mathcal{P}_{\mathrm{o},j}$ and $\mathcal{P}_{\mathrm{s},j}$ are the operational and sleep circuit power consumption, respectively, η_j is the power amplifier efficiency, Λ_j is the load factor, and $\mathcal{P}_{\mathrm{T},j}$ is the transmit power. These values vary according to the type of the BS, such as macro, remote radio head (RRH), micro, pico, and femto, whose power profiles are provided in Table 2.2.

2.3.4 Energy Saving through Cell Switching

BSs have been the major contributor for the power consumption in cellular networks [41, 161], putting the energy saving from BSs at the heart of the green cellular networking concept. In this regard, various techniques for energy saving are shortlisted in [41], including: hardware based energy efficiency enhancement, component switching off, radio transmission based energy efficiency improvement, use of renewable energy resources, and making the network more heterogeneous.

Table 2.2: Power profiles for different types of BSs [156]

BS Type	Efficiency	Power Consumption [W]		
		Transmit	Operational	Sleep
	η_j	$\mathcal{P}_{\mathrm{T},j}$	$\mathcal{P}_{\mathrm{o},j}$	$\mathcal{P}_{\mathrm{s},j}$
Macro	4.7	20	130	75
RRH	2.8	20	84	56
Micro	2.6	6.3	56	39
Pico	4.0	0.13	6.8	4.3
Femto	8.0	0.05	4.8	2.9

Of all these techniques, while the use of renewable energy sources can be classified as an energy harvesting method, the rest would be in the category of energy conservation. Given that the focus of this thesis is energy conservation, the energy harvesting methods are out of the scope, thereby they will not be discussed further.

In the aforementioned energy conservation techniques, component switching off has a special place due to the following reasons:

- Even after implementing all the other techniques, such as advanced hardware and transmission efficiency, and increased network heterogeneity, there is still room for saving energy by switching off the components. Let us use an analogy to make this statement more clear: consider a software company producing intelligent solutions for their clients. There are hundreds of employees working in an office environment with their desktop computers from 9am to 5pm on weekdays. Imagine the company wants to cut its energy bills in order to increase their profit. To this end, the company decides to replace the low energy efficient desktop computers and light bulbs with the new very high energy efficient ones. Although this is a good move in saving energy, it is not adequate provided that the computers and light bulbs are kept switched on for 24 hours everyday. Therefore, more energy could be saved by switching off the computers and light bulbs when they are not used. Considering this analogy, the cell switching introduces another dimension in energy saving: while the other energy conservation techniques focus on reducing the energy consumption when the components are operating, the cell switching brings the idea of putting the components to sleep when they are under or not utilised.

- Enhancing the energy efficiency of the hardware components and making the network more heterogeneous either requires new deployments or introduces new problems [41]. Radio transmission optimisation, for example, would require new wireless communication standards and/or concepts. However, cell switching can be implemented without extensive new replacements or communication standards, thus its plugging into the existing cellular networks is quite straightforward [41].

To this end, the underlying idea of cell switching is to deactivate components of a BS at low traffic, which is usually referred as putting the BS in a sleep mode. Various types of sleep modes have already been identified in [162], in which the sleep mode changes based on the depth of the sleep that is correlated to the deactivation time of the component switched off. More particularly, the components in a BS are categorised according to their deactivation times in [162], and the depth of the sleep determines the components to be slept; the more components are put into the sleep mode with increasing depth.

State-of-the-art in Cell Switching

A broad range of works related to cell switching are available in the literature, as cell switching has been recognised as a valid solution to minimise the energy consumption of cellular networks. A brief state-of-the-art review is provided in the following paragraphs with an objective of highlighting the diverse set of existing works.

A comprehensive energy management scheme is proposed in [82]. The authors consider a smart grid integration with cellular networks, where the smart grid communicates with a cellular network for adjustments in power consumption. In particular, based on the energy demand from the grid, a smart grid operator requests increments or decrements in the power consumption of the cellular network. If the cellular network responds to these requests, it obtains benefits in return in terms of reduction in energy bills. The proposed system use both grid and renewable energy as power supplies along with an energy storage unit. The cell switching concept is also employed in order to decrease the energy consumption of the network in an attempt to increase the number of responses to the requests of the smart grid operator. A threshold and rule based cell switching approach is adopted, where the SCs are switched off if:

- the traffic load of the SCs are below a certain threshold, which is subject to an optimisation; and
- the MC has enough capacity to handle the traffic load of the switched off SC.

The authors in [83] propose a mutual repulsive cell switching strategy, and investigate both the energy and coverage efficiencies in a SC network scenario with MIMO system. The strategy is introduced for light traffic cases, and the process is as follows:

- each SC calculates its proportional traffic load;
- the BS with the largest proportional load in a circle, with a given centre point and radius, is kept on, while the other BSs inside the circle are switched off.

The obtained results reveal that both the energy and coverage efficiencies are enhanced with the BS density and the distance between the two active BSs.

Both site and sector based switch-off regular patterns—which are basically combinations of cellular network settings—are investigated in [84]. In this context, in order to perform the cell switching, the authors utilise cellular layouts rather than building a real-time decision making unit. In particular, the developed method relies on deciding the sectors/sites in advance through an offline process. Thus, the authors identify some regular switching patterns and compare the performance of the proposed solution in terms of energy efficiency, the number of UEs served, and SINR distribution.

A motion sensor based cell switching approach is designed in [65]. The work is performed particularly for indoor environments, where typically a certain number of people stays in a room that is covered by a SC. In an office environment, for instance, when the employees leave, the SCs becomes idle as no traffic is generated. Thus, this work uses a motion sensor at each room/office in order to detect the presence of users, which in turn triggers the cell switching decision. The authors conduct experimental studies with an Open Air Interface based 5G test-bed, and the results show a considerable amount of energy saving.

Another interesting idea is proposed in [85], where the HO traces of users are utilised in order to perform cell switching decisions. As such, rather than relying on the observations on traffic variations, the authors develop a model that predicts the user HOs in order to determine the *on/off* status of a cell. More specifically, in the developed model, future traffic loads of SCs are determined through HO predictions, which are enabled by a semi-Markov process. Based on the estimated future loads, an optimisation problem of minimising the energy consumption is created, with constraints of minimum coverage and bit rate requirements. In this regard, a joint optimisation of the following variables are conducted:

- optimum set of SCs to turn off/on for energy saving purposes;
- cell-specific offset for load balancing purposes, which subsequently helps maintain a good level of QoS.

2.3.5 Research Gap Analysis

Although there are many cell switching algorithms are available in the literature, there is still need for a dynamic, robust, real-time, and practical solution. The heuristic methods, for example, are often unscalable an inadaptive albeit performing well in certain scenarios. For a binary (*on/off*) cell switching problem, for instance, the number of possible switching combinations increases exponentially with the size of the network, and even though this can be handled with heuristics, such as genetic algorithm, to some extent, many practical scenarios would necessitate more scalable and less complex algorithms. This gains more importance considering the expected network densification in 5G networks.

On the other hand, predictive cell switching methods, which is enabled by direct or indirect (through HOs) traffic predictions, requires historic data along with a training phase. This might be impractical for many circumstances, since the historic data would not be immediately available at all the scenarios. Moreover, computational power and storage demands are additional issues to consider for such implementations, and thus this kind of solutions would come with side effects. Furthermore, approaches that require additional hardware deployments are not always feasible and practical. For example, the work in [65] necessitates a motion sensor deployments for indoor environments, and this would be needed in each room where SCs are deployed. However, in addition to being infeasible for many cases, such as indoor scenarios with huge number of independent rooms, the maintenance of this kind of implementation is another issue to address.

IoT networks are expected to generate a huge amount of data, as it is envisioned that an immense number of devices—with their diverse characteristics and application areas—are connected to the Internet through the IoT concept. Therefore, this data can be exploited for the benefit of IoT networks in terms of making them more energy efficient. In this regard, data analytics and ML play vital roles, since it is hard to deal with such a large volume of data via conventional optimisation techniques. Moreover, it is quite likely that IoT devices can be deployed into locations which are inherently dynamic; e.g., ports and urban areas. As such, adaptability gains more importance, thereby the network optimisation needs to be intelligent, which can be enabled through learning. Furthermore, conditions for each IoT application would be different from each other (e.g., the requirements of agricultural applications would be distinct from health-care applications), hence context awareness should also be one of the key ingredients in optimising IoT networks. A more detailed research gap analysis can be found in Sections 3.1 and 3.2 in Chapter 3 for IoT energy optimisation and cell switching, respectively.

2.4 Cellular Network Capacity Enhancement

Cellular network capacity enhancement will be investigated from two different perspectives in this section, namely predictive mobility management and UAV positioning. After discussing the mobility management in cellular networks in general, the concept of predictive mobility management will be introduced along with a state-of-the-art presentation. Then, UAV assistance in cellular networks will be elaborated, followed by presenting the corresponding literature review. Lastly, a research gap analysis concludes the section.

2.4.1 Mobility Management in Cellular Networks

3GPP has defined mobility as *"the ability for a user to communicate whilst moving independent of location"*, while mobility management has a definition of *"a relation between the mobile station and the UTRAN (UMTS Terrestrial Radio Access Network) that is used to set-up, maintain and release the various physical channels."* [163]. Although these definitions are from Release 4, 3GPP sticks with them in Release 15 for 5G [164]. The aforesaid definition of mobility yields that the connections for the users are supposed to be maintained when they are mobile, and mobility management is the concept that ensures this connection. Therefore, mobility management plays a crucial role in cellular networks, since it is a challenging task to provide services to the users with diverse mobility profiles; e.g., stationary, low-mobility, high-mobility, etc.

Moreover, there are two different radio resource control (RRC) connection states in legacy networks, namely idle and connected. However, inactive state has been introduced in 5G NR as a new type of RRC connection state [165]. Before going in detail about it, it is better to give the 3GPP definitions for conventional idle and connected states [163]. If a UE has no RRC connection with any radio access network (RAN) while being turned on, it is referred as in the idle state. This means that the UE is tractable (in connection with the network), but is unable to transfer data. The connected state, on the other hand, refers to the case when the UE is not only turned on and tractable, but also has an active RRC connection established with a RAN, and is able to transfer data. There is a trade-off in switching between the connected and idle states, such that, from the UE's perspective, it switches to idle state in order to save energy, since measurement reporting drains its battery. However, in order to transmit data, the UE then needs to switch back to the RRC connected state, which results in latency and signalling overhead [166].

With the introduction of the novel inactive state, now the UE in the connected state can be classified as either active or inactive, depending on the data transferring activity; i.e., inactive if there is no data transferring session, or active when it has an ongoing data transfer. Then, the UE that is not active in connected state can be switched to the inactive state rather than the idle state by releasing the RRC connection while keeping the core network connectivity [165–167]. As such, by entering the inactive state, the UE can save energy while avoiding the heavy signalling that arises from the switch between the connected and idle states.

From the mobility perspective, the mobility of the UE in the idle or inactive states are controlled by the UE itself, whereas the network becomes responsible for the mobility if the UE is in the connected state [165]. As reported in [168], mobility management in cellular networks can be broadly divided into two categories as idle/inactive state and connected state mobility. A similar fashion was followed in [169] with a slightly different terminology, namely location and HO management. In the following paragraphs, these different mobility concepts are investigated in a more detailed way.

- **Idle/Inactive State Mobility (Location Management):** corresponds to the procedure taken just after the UE is powered on, as defined by 3GPP in [165, 167]. Upon powering on, the UE is first supposed to select a public land mobile network (PLMN), and it performs a cell selection based on current measurements and the "S" criterion, as defined in [167]. When it selects a cell to camp on, it needs to continuously search for a better cell, which is called as cell re-selection. The UE is also required to execute location registration, which helps the network to have an approximate location of the UE in order to perform a paging process. Therefore, in idle/inactive state mobility, location registration, cell selection, and cell re-selection are the primary tasks to be performed, and the procedures for the idle/inactive states are individually and comprehensively elaborated in 3GPP Release 15 [170].

- **Connected State Mobility (Handover Management):** corresponds when the UE is in the connected state and has an ongoing data transmission. As such, the user is regarded as active and its mobility management becomes more challenging when compared to the aforementioned idle/inactive state mobility. Moreover, unlike the idle/inactive state version, the connected state mobility is network controlled, but UE assisted. In particular, the UE keeps performing measurements on the signal quality from both the serving and neighbouring BSs in order to ensure that it is connected to the best cell around it [43, 59, 168, 169]. When the UE is mobile, meaning that it changes its geographic location, it is quite likely that the signal qualities that it receives through measurements would vary continuously. This, by its turn, results in some neighbouring BSs providing better signal qualities than the serving one. Therefore, in these kind of cases, the UE would require its connection to be switched from the current serving cell to a neighbouring cell that is better than the serving cell in terms of signal quality. In this regard, HO is defined as the change of a serving cell of the UE while being active; i.e, having an ongoing data transmission.

Moreover, in terms of measurement reporting, it can be divided into two categories, as periodic and event-triggered [168]. In periodic reporting, the UE performs measurements at regular intervals. On the other hand, in event-triggered reporting, the UE sends its measurement reports to the serving BS only upon the occurrence of certain event conditions [168]. In other words, the UE assists the HO process in the following ways:

- by making measurements on signal strength from serving and neighbouring BSs;
- sending the measurement reports to the serving BS through one of the aforementioned strategies (i.e., periodic or event-triggered).

The network control starts after this point, in which the serving BS communicates with the target BS—selected as the best cell—to check its availability. Once the target cell admits the UE by ensuring that it has enough resources for it, the HO process is executed and completed through multiple signalling exchanges between different entities (e.g., UE, serving BS, target BS, and core network). This means that the UE is handed over to the target cell—which is the new serving cell—and the resources, which were allocated for the user in the previous serving cell, are released.

HOs can also be put in different taxonomies, such as intra-frequency and inter-frequency [168–170]. Intra-frequency HOs occur when the UE switches to a BS that transmits in the same frequency with the previous serving cell, while the HO is referred as an inter-cell HO if the carrier frequencies of the serving and the target cells are different [168–170]. Provided that two different FRs[4] are already introduced in 5G NR, namely FR-1 and FR-2 [15], HOs across different FRs and within the same FR are possible. Even though the HOs between different FRs can only be performed as inter-frequency,

[4] While FR-1 covers sub-6GHz frequencies, FR-2 includes mmWave frequencies above 24 GHz.

the HOs within the same FR can be done as either intra-frequency or inter-frequency [170].

2.4.2 Predictive Mobility Management

It has become a cliché that mobility management in cellular networks can be made more efficient and agile with the help of mobility prediction [43,59,168]. Considering the cognitive cycle [70] and cognitive networking [69] concepts for wireless communication networks, predictive mobility management can be broadly explained in three steps:

- **Obtaining contextual information:** mobility-related behaviour of users are obtained in this phase. Some examples of contextual information to be gathered are [59]:

    1. exact locations of users (e.g., geographic coordinates);
    2. attraction points visited;
    3. trajectories taken;
    4. mode of transportation (e.g., walk, car, bicycle, etc.);
    5. sojourn time;
    6. identification of places visited (e.g., home, work, etc.);
    7. HOs performed.

    While some contextual information (e.g., location, sojourn time, and HOs) can be directly obtained with the help of both the UE and the network, some of them (e.g., identification of places and attraction points visited) require additional processing/analysis. Considering the needs of the application and capabilities of the agents[5], the information to be collected can be decided accordingly.

- **Analysing/Learning:** the collected information can be analysed in order to exploit meaningful information. This phase is needed due to the fact that the raw data could be hard to understand and does not necessarily include useful information [171–173]. Moreover, there could be hidden patterns in the data, which is difficult to figure out without further analysis. A raw data set consisting of coordinates of a user with corresponding time stamps

[5] Agent can refer to the entities that are involved in the predictive mobility management. It can be an individual entity, such as UE, as well as cooperation of different entities, such as UE-BS pair.

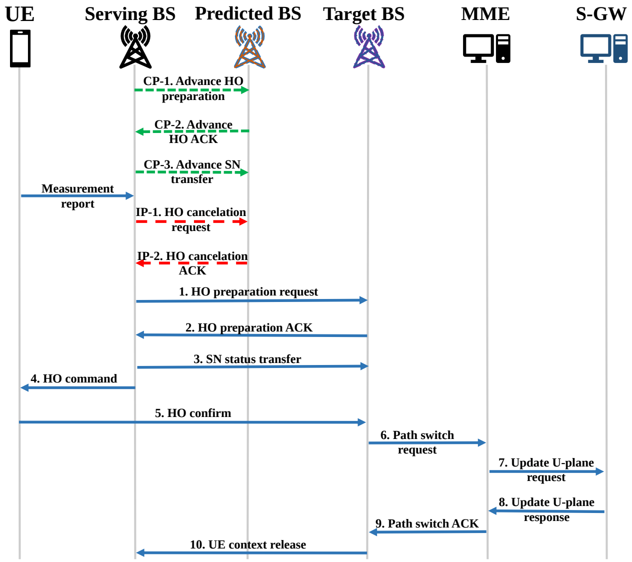


Figure 2.4: LTE X2 interface based HO process adapted from [43, 133]. CP: Correct Prediction, IP: Incorrect Prediction, SN: Sequence Number, ACK: Acknowledgement. Steps CP-1 , CP-2, and CP-3 only apply to the CP case, while steps IP-1 and IP-2 are only valid for IP case.

at certain time intervals would be not meaningful as it is, however, some additional processing, such as statistical analysis or ML algorithms, can reveal some correlations and patterns in the data. For example, a user always leaves their home at around 08.30 in the morning in weekdays, and commutes to work. The user also stops by a certain café to have a cup of coffee before completing its route to work. However, the user does not have any regular pattern at weekends, and takes different trajectories; e.g., sometimes stays at home the whole day, and some other times goes to various events, such as the cinema, a musical concert, etc., at different locations. Therefore, it can be inferred that this specific user has certain patterns with almost strict timings in the weekdays, whereas no obvious patters/regularities are observed for weekends. In summary, this analysing/learning phase converts the obtained information to meaningful knowledge.

- **Future predictions:** after data collection and its corresponding analysis, the developed knowledge/model can now be utilised for various use-cases. By analysing the HO occurrences for a specific user by taking into account the involved BSs and the time stamp, the next or multi-step ahead HOs of the users can be predicted, which in turn help in achieving seamless HOs.

Fig. 2.4 demonstrates the LTE X2-based HO flow diagram for both predictive and non-predictive HO cases [43,133]. Note that predictive HO refers to the case where the target BS for the next HO of the user is predicted, which subsequently helps reducing the number of steps taken during the HO by executing some of the steps in advance. Furthermore, HO predictions are divided into two categories according to the accuracy of the prediction, namely: correct predictions (CP) and incorrect predictions (IP). While the former means that the target BS for the next HO is predicted accurately, the latter happens when the predicted and target BSs are not the same. In particular, steps 1 to 10 are performed during non-predictive HO; 4 to 12 are for predictive HO with CP; and IP-1, IP-2, and 1 to 10 (all inclusive) for predictive HO with IP [43, 133]. It is worth noting that steps CP-1, CP-2, and CP-3 are taken in advance of a HO for the CP case, meaning that steps 1, 2, and 3 of non-predictive case are eliminated with the predictive HO management with CP.

Therefore, as it is obvious from Fig. 2.4, the predictive HO with CP case is more profitable compared to the non-predictive case in terms of signalling during an HO, whereas IP makes the predictive HO case more costly than the non-predictive one. There are two primary takeaways from these phenomena:

- HO processes can be made less costly via predictive HO with CP, resulting in less signalling overhead and HO latency. This, in turn, promotes seamless HOs, which is envisioned to be one of the integral characteristics of 5G networks [174, 175].

- Prediction accuracy plays a crucial role, since IPs can make the whole process less efficient by increasing the HO signalling and latency. Thus, more efficient mobility prediction algorithms should be developed given that the average gain from the predictive mobility management process enhances with increasing prediction accuracy, such that [43, 133]:

$$E[H_{\mathrm{t}}] = A_{\mathrm{c}}H_{\mathrm{c}} + (1 - A_{\mathrm{c}})H_{\mathrm{i}}, \tag{2.16}$$

  where $E[H_{\mathrm{t}}]$ is the expected HO cost, A_{c} is the prediction accuracy, and H_{c} and H_{i} are the HO costs for CP and IP cases, respectively. Note that the HO cost can be any cost related to HO, such as signalling overhead and latency.

There are many studies available in the literature regarding the predictive mobility management for cellular networks, and they are discussed in the following paragraphs.

State-of-the-Art in Predictive Mobility Management

In this section, recent predictive mobility management approaches are presented in order to draw the state-of-the-art. In [86], the authors aim to reduce ping-pong HOs as well as HO failures. SINR is predicted in order to obtain knowledge on future received power levels from the serving BS. To this end, HO margin and TTT are considered as HO parameters, such that TTT is dynamically adjusted based on SINR predictions, while HO margin is kept fixed. First, when any candidate BS starts outperforming the serving BS in terms of received signal power, instead of waiting for a fixed TTT for this condition to be maintained, the authors propose to use predicted SINR values from the serving BS, such that the HO is only triggered if the SINR value of currently serving BS is found to be less than a certain threshold—an SINR value sufficient for healthy connection—for a certain time period.

Another similar work is proposed in [87], in which the authors introduce a comprehensive mobility prediction process. Two different types of memories, namely short-term and long-term, with different objectives are initially introduced. In particular, by utilising stored signal strength measurements performed by the UE, future signal quality and signal strength predictions are carried out for both the serving and neighbouring BSs, followed by checking whether the predefined HO criteria is met. After that, the output of this process is consolidated by a statistical analysis with the help of historical HOs that are stored in the long-term memory.

The authors in [88] developed a predictive mobility prediction model for cellular networks with the objective of making the resource utilisation more efficient via proactive HO management. A semi-Markov renewal process is considered to model the user mobility, and corresponding HO predictions are performed through this model. In order to evaluate their model, the authors carried out data collection, where the HOs of selected users were recorded with corresponding time stamps for a period of one month. The developed semi-Markov model trained with this collected data set.

In [89], on the other hand, a recursive neural networks (RNNs) based mobility prediction approach is presented. In this regard, two different RNNs architectures, namely long-short term memory (LSTM) and echo state network (ESN), are developed and compared to a Kalman filter predictor. The trajectories used in training and testing phases are generated artificially (i.e., synthetic data) in order to evaluate the performance of the considered methods. The work mostly focuses on the prediction accuracy by investigating various parameters affecting it, such as RNN architecture and output structure.

A probability suffix tree (PST) based predictive mobility management is introduced in [90] by considering an ultra-dense cellular network with control-data separation. Moreover, two different PST learning algorithms are developed; i.e., user-specific and global. While the former is utilised when users have frequent visit patterns in the master eNodeB in question, the latter is for those who can be considered as random users due to their insufficient presence in the area of the master eNodeB. Furthermore, during the execution phase of the developed mobility method, two different approaches are adopted: network-controlled and user-autonomous. Similar to the aforementioned works, which proposes a predictive HO management, the network-controlled one helps in shortening the HO process by executing some of the steps prior to the HO event. The user-autonomous one, on the other hand, is proposed to transfer some of the HO responsibilities to the user in order to reduce the network load.

2.4.3 UAV Assistance in Cellular Networks

UAVs have already been identified as a promising solution for wireless networking due to their multiple advantages including flexibility and mobility [176]. They can be used for various purposes, such as capacity enhancement—especially for UDNs—and emergency scenarios; e.g., earthquakes, fires, etc. Typically a SC is mounted on a UAV in order for it to act as a flying BS, which have multiple advantages over conventional ground SC deployments:

- they are deployed in the sky, thereby it is more likely to establish LOS with users [177], which, by its turn, enhances the capacity of the network and provides improved data rates to the users;

- they are mobile, hence they can adjust their positions in terms of not only longitude and latitude but also altitude in order to provide better QoS in changing conditions;

- they can be deployed on demand, thus they are cost effective as they can be utilised multiple times at different locations/scenarios [177, 178]. In other words, rather than deploying fixed ground SCs by considering the peak traffic around the geographic area, UAVs can be deployed when needed. This provides not only reduced deployment costs for mobile network operators but also flexibility to cellular networks.

On the other hand, there is a number of designs challenges that should be addressed properly in order to make the UAV assistance viable and sustainable. Some of the primary challenges identified in [176–179] are as follows:

- **Energy efficiency:** UAVs typically present energy limitations owing to their off-grid characteristics, such that they use on-board power supplies including battery and fuel. In this regard, their energy consumption is of the utmost importance, and should be managed efficiently.

- **Backhaul connectivity:** due to the fact that fibre backhauling—or any other wired option—is no more applicable in the UAV SC scenario, a wireless backhauling is required. Even though there are multiple available options, such as free-space optics, satellite, mmWave, each of these constitutes distinctive challenges. Low coverage capability and LOS requirements are the main problem for mmWave and free-space optics, for example, while latency becomes problematic for the satellite case [176, 178].

- **Placement and path planning:** the placement of UAVs considering all three dimensions (i.e., longitude, latitude, and altitude) is one of the major challenges given that the position of the UAVs can have a significant impact on communication performance from the perspective of wireless link establishment. If, for example, lower altitude is chosen, then the LOS probability will be lower given that it is a function of the height of transmitting antenna [180].

  Path planning, on the other hand, is another design challenge for moving UAVs, as the selected trajectory plays an important role on multiple elements including communication performance and energy efficiency [176]. Determining a longer path for a UAV, for example, would drain its on-board power supply quicker, but it could be a good choice in terms of communication performance considering the channel conditions in order to avoid interference [181,182]. As such, there are multiple trade-offs, and the trajectory should be optimised according to the circumstances and requirements.

State-of-the-art in UAV Assisted Cellular Networks

A mathematical framework is developed in [91], wherein the UAVs are used to assist in backhauling. More specifically, the authors consider mmWave backhauling for mobile ground BSs, where NR-BSs are employed as a main backhaul option. Furthermore, when the backhaul connection with the NR-BS is lost (or becomes poor) due to blockages, the mobile BS's backhaul connectivity is switched to UAV relay nodes. The developed framework is tested through system level simulations, where the effects of various UAV related parameters including UAV density, speed, and capacity are observed via different performance metrics, such as outage probability and spectral efficiency.

Both large and small timescale optimisation problems are formulated and solved in [92] for the UAV-assisted industrial power line inspection scenario. In particular, UAVs are employed in the power line inspection due to multiple reasons, namely safety, accuracy, and cost considerations. As such, the UAVs visit a power line site and collect raw image data to be transferred to a central unit for further analysis. Therefore, the UAVs are expected to carry out signal strength based distance calculations in order to refrain from possible UAV collisions. Moreover, rather than real-time data transmission, they first store the collected data on their on-board storage unit and transfer it to the relay UAV, which is responsible for transferring it to the central unit. To this end, the trajectories, velocities, frequency regulations, relay node selection, and power allocation are optimised for energy reduction purposes using dynamic programming, auction theory, and matching theory.

A channel modelling is investigated in [93] by focusing on path loss exponent and shadowing. The authors perform experimental studies, in which a UAV is used as a UE and equipped with a radio network scanner. In this regard, location-specific RSRP measurements are carried out for different UAV altitudes. Besides, a ground-drive test is also conducted for referencing purposes. From the analysis of the obtained results, the authors propose a new path loss exponent and shadowing model that consider the altitude of the UAV, since the measurements results from the ground driving test and the UAV are quite different.

An RL based UAV positioning algorithm is proposed in [94]. In particular, the authors design a UAV assisted cellular networking in the case of emergency, where the existing cellular infrastructure is out of service due to the experienced damage. A Q-learning based positioning of multiple UAVs, on which SCs are mounted, is aimed with an objective of increasing the number of users served by the UAV SCs. Provided that reaching the maximum amount of people is crucial for emergency scenario in order to maximise the rescued lives, this is a well-found objective.

2.4.4 Research Gap Analysis

Even though ML has been widely applied in predictive mobility management, there are still open points to address. For example, since a predictive mobility management with incorrect predictions are likely to incur more costs than the conventional non-predictive mechanisms, it is crucial to a develop cost models according to the objective of the application, such as signalling cost reduction, latency minimisation, admission control, etc. This cost model should include all the possible cases including predictive process with correct and incorrect predictions, and non-predictive process; then, this model can be used to assess the applicability of the predictive process with given prediction accuracies. Most of the existing predictive HO approaches lack this kind of investigation, thereby no information is made available about when to trigger a predictive process.

Furthermore, Markov chains are one of the most commonly used prediction methods [59] due to their simplistic algorithm. However, this simplicity can also bring additional challenges in more complex scenarios, which the conventional Markov chains would be incapable of dealing with. This is mainly arising from the limited contextual information they process, making them prone to producing errors. A higher-order Markov chains, which is capable of including more contextual informations (e.g., certain number of previous transitions), have already been proposed in the literature, but they are often unscalable, as their transition matrix exponentially grows with the number of states. Therefore, better solutions need to be appended into the algorithms in order to improve their capabilities while benefiting from their simplicity.

On the other hand, a considerable amount of research activities focus on UAV assisted wireless communication networking. Among them, many works have been trying to address the UAV positioning problem with different methodologies. However, due to the inherent real-time characteristics of the problem, the solution to be implemented should incur low computational cost and be adaptable, since the mobile environment continuously change. Therefore, implementations of computationally demanding positioning optimisations are infeasible and impracticable owing to the fact that rapid decisions should be taken. In other words, even when ML is used as a positioning algorithm, it is supposed to be adaptable and computationally low-cost. A more detailed research gap analyses can be found in Sections 4.1 and 4.2 in Chapter 4 for mobility management and UAV assistance, respectively.

2.5 Dynamic Spectrum Access Techniques for Cellular Networks

This section starts with a generic introduction of the CR concept. Then, among all the phases involved, the spectrum sensing is exclusively studied by dividing the existing techniques into two based on the size of the bandwidth to be sensed. After that, the predictive sensing approach is demonstrated, followed by the corresponding state-of-the-art. Lastly, the chapter is concluded with a research gap analysis.

2.5.1 Cognitive Radio Networks

The following points are worth noting from the definition of a CR system (CRS) by ITU [183]:

- obtaining information from the environment;
- adjusting parameters dynamically;
- learning from experience.

These steps remind the cognitive cycle introduced by Mitola III: observe, orient, plan, decide, act, learn [70], which is also in line with the definition of cognitive networking made by Thomas, R. W. et al. in [69].

Two types of users are introduced in CR concept: primary users (PUs), who have a priority to access the spectrum as they are licensed users; and secondary users (SUs), who are unlicensed and expected to use the spectrum opportunistically [47, 49, 60, 184]. As stated in the capabilities of CR in the report published by the Federal Communications Commission (FCC) [185], the term *opportunistic* refers to the fact that SUs are required to sense the EM spectrum and find a frequency hole to utilise with a condition that they should not cause an interference with the PUs.

There are four main CR spectrum management phases, namely spectrum sensing, decision, sharing, and mobility [60, 184]. In the following paragraphs, each of these four steps will be discussed in detail.

Spectrum Sensing

Spectrum sensing refers to the action taken by the SUs, since they are only allowed to use the EM frequency spectrum opportunistically. To this end, the SUs need to detect whether there is a PU in a spectrum portion of interest in order to allocate it. If there is a PU, they cannot use it and are supposed to sense another portion owing to the fact that PUs have priority to use the spectrum and SU can only use it when it is available [47, 49, 60, 184]. Therefore, this is an important phase, where the SUs find a way to access the spectrum that is one of the initial requirements to establish a wireless communication link.

Spectrum Decision

Once the available spectrum holes, which are unused at the time, are detected through one of the spectrum sensing methods, then the SUs are supposed to decide the channel to use by considering their QoS requirements. Spectrum decision is performed via three sub-phases [186]:

- spectrum characterisation;
- spectrum selection;
- CR reconfiguration.

In spectrum characterisation, the SUs need to identify and categorise the available frequency holes based on various parameters including interference, path loss, wireless link errors, link layer delay, and holding time [187]. Then, the most appropriate channel is selected by taking the QoS requirements into account in the spectrum selection phase, followed by reconfiguring the radio-related parameters, such as transmit power and modulation scheme [186, 187].

Spectrum Sharing

CRNs can operate both on licensed and unlicensed spectrum bands [187]. In the case of licensed spectrum operation, the SUs are supposed to avoid an interference with the PUs, which use that spectrum with their license and thus have a priority to access it. In this regard, a medium access control (MAC) layer solution is needed to ensure that the SUs do not cause interference to the PUs [188]. This spectrum sharing approach is beneficial not only with regard to the interference control between SUs and PUs but also for obtaining a proper management among the SUs, which try to access the same spectrum portion [187, 188]. When the SUs attempt to use the unlicensed spectrum, on the other hand, efficient spectrum sharing methodologies are supposed to merely safeguard against the possible collisions between the SUs provided that there is no PU in this type of spectrum bands.

Spectrum Mobility

Spectrum mobility is a process that occurs when a PU wants to use the spectrum band, which is already occupied by an SU. In other words, when the SU is operating on a licensed spectrum band, as explained in Section 2.5.1, it is expected to sense the spectrum in order to find a spectrum hole—an unused part of the spectrum [60, 184]. After performing all the aforementioned steps (i.e., spectrum sensing, decision, and sharing), the SU becomes able to use the licensed spectrum, however, given that the SUs are considered as visitors [187], they need to vacate the allocated spectrum band once the PU presence is detected. This phenomena is called spectrum mobility, since the SU moves from the previously allocated spectrum band to another. The process of switching the SU from one spectrum band to another is referred as spectrum HO, which has a direct impact on the overall communication performance [189].

2.5.2 Traditional Spectrum Sensing Techniques

Narrowband Spectrum Sensing

In NB spectrum sensing, only a small portion of the spectrum is the focus of the sensing process, and the bandwidth involved is often smaller than the coherence bandwidth of the channel [190]. In this case, the SU is basically interested in knowing whether a PU is present or absent in the channel of interest and this can be depicted as a binary decision process with 1 indicating the presence of a PU and 0 indicating the absence of a PU. The presence or absence of a PU is ascertained based on certain detectable characteristics in the channel, such as signal-to-noise ratio (SNR), type of modulation, signal periodicity, eigenvalue, etc. [191]

Methodologies: Various approaches for NB spectrum sensing have been developed in the literature [190, 191].

- **Matched filter technique:** the matched filter technique is applied in a situation whereby the SU has prior information about the PU signal including the type of modulation, bandwidth, frequency, etc. Thus, signal received from the spectrum of interest is then compared with the already known PU signal as well as a predefined threshold in order to detect the presence of a PU and differentiate it from a noise signal [190]. The authors in [102], develop a matched filter based spectrum sensing technique in order to determine the power level and presence of the PU for scenarios where the PU transmits with more than one power level.

- **Energy detector approach:** in the energy detector approach, the SU simply estimates the energy level of the received signal in the channel of interest and compares it to a predefined threshold value. Then, a PU is said to be present if the detected energy level is greater than the threshold, otherwise the PU is absent [192]. This is a simple approach where previous knowledge of the signal characteristics is not necessary, however, the energy threshold level needs to be carefully selected as it significantly influences the accuracy of the PU detection. In [103], the authors propose an energy detection based spectrum sensing technique by measuring the noise power of the received signal. A dynamic threshold selection mechanism is also implemented, since the detection performance depends on it.

- **Cyclostationary detection method:** this method depends on certain characteristics of the transmitted signal in the channel of interest, such as the type of modulation, frequency, etc., which are often periodic and are referred to as cyclostationary features [191]. The cyclostationary detection (CD) approach is able to detect the presence of a PU and differentiate it from noise signal because of the stationarity and lack of correlation of noise signal by determining the spectral correlation of the received signal. Once the mean and autocorrelation of the received signal is found to be periodic, it is said to be cyclostationary, and hence the presence of a PU is confirmed. A CD based spectrum sensing approach is developed in [104] to detect the presence of PUs in a channel.

- **Covariance-based detection technique:** in this approach, the presence or absence of a PU is determined with the help of the covariance matrix of the signal as well as singular value decomposition [193]. First, the covariance matrix of the signal obtained from the channel is analysed. Secondly, the eigenvalues of the received signal are obtained by applying single value decomposition on the covariance matrix of the signal. Lastly, the presence or absence of a PU is obtained by dividing the highest eigenvalue by the lowest eigenvalue, followed by comparing the result with a predefined threshold. A value above the threshold indicates the presence of a PU, otherwise the channel is free. The authors in [105] propose an improved covariance-based spectrum sensing algorithm for the detection of PU in a channel. To overcome the limitations in the performance of conventional covariance based technique due to the difficulty in selecting the threshold value, they introduce a scheme which utilises the statistic of the covariance matrix as well as a goodness of fit test to determine the presence or absence of a PU.

Limitations: NB spectrum sensing techniques are often simpler approaches of detecting the presence of a PU. However, this simplicity comes with some drawbacks [190, 191]. Firstly, the SUs have less opportunity to find a suitable spectrum hole, since only a small portion of the spectrum is selected for sensing per time. Secondly, repeated sensing of the spectrum needs to be carried out in order to find a suitable vacant spectrum for the SU, thereby leading to increased sensing latency and energy consumption. Finally, the performance of all the sensing techniques discussed is dependent on the proper selection of the threshold value, hence choosing a suitable threshold becomes a challenging issue that needs a careful consideration [103].

Wideband Spectrum Sensing

When the SUs conduct spectrum sensing for a bandwidth, which is larger than the coherence bandwidth, it is referred to as WB spectrum sensing, and the SUs are supposed to employ WB sensing methods rather than previously discussed NB ones. The underlying idea of why NB sensing methods are not considered in the WB scenario is that the NB sensing methods have binary decision process, where the SUs try to detect the presence of PUs with a simple binary decision (i.e., 0: PU absent, 1: PU present) [190, 191]. This binary decision phenomenon would make the sensing process inefficient for a WB spectrum owing to the fact that the decision is made for the complete spectrum, resulting in the omission of some available spectrum holes. Therefore, various techniques have been developed to detect the PU presence efficiently in the WB spectrum without loosing opportunities.

Methodologies: Since this section follows the taxonomy provided by [191] and [190], the WB spectrum sensing methods are split into two different categories: Nyquist-based and sub-Nyquist based. Nyquist–Shannon sampling theorem states that the sampling frequency must be at least twice of the signal frequency in order to avoid alising, which is a phenomenon occurring when two consecutive sampling copies overlap in the frequency domain so that they become indistinguishable. To this end, the methods that sample the received WB signal at the Nyquist rate, which is exactly twice of the signal frequency, are referred as Nyquist-based methods, while those perform sampling under the Nyquist rate are called sub-Nyquist-based methods.

Nyquist-based Approaches: Some of the main Nyquist-based spectrum sensing approaches are listed below:

- **Wavelet transform based spectrum sensing:** even though the Fourier transform provides information about the frequency components included in the signal, it does not state the occurrence time of the frequency components. As such, the wavelet transform theorem helps in observing both frequency and time related information simultaneously, making it a convenient solution for WB spectrum sensing [194]. Both continuous and discrete wavelet transform are employed in [106], in which the authors implement continuous and discrete wavelet transform and logarithmic scaling along with a threshold to perform an edge detection. To this end, in order to combat noise amplificiation, the authors develop a discrete wavelet transform method with a moving average filtering.

- **Multiband spectrum sensing:** as performed in [195], the multiband spectrum sensing method is implemented through the following steps [191, 196]:

  1. the received WB signal is sampled by an analog-to-digital converter (ADC) at a high sampling rate;
  2. serial-to-parallel conversion is applied in order to split the data into multiple parallel data streams;
  3. FFT is implemented for time-to-frequency domain conversion;
  4. a series of narrow band signals are obtained by splitting the resulting WB signal;
  5. for each NB signal obtained, one of the NB spectrum sensing techniques is utilised to detect the PU presence.

  There is a more advanced version of this type of multiband spectrum sensing—named swept-multiband spectrum sensing—that benefits from less ADC sampling rate requirements [107, 191, 196], and in [107], the authors investigate its IQ (I: in-phase, Q: quadrature) imbalance and aliasing problems.

- **Filter-bank based multiband spectrum sensing:** filter-bank based spectrum sensing method, as the name suggests, proposes to use a prototype filter bank—where each filter has different central frequency—in order to sense the received WB signal [190, 191, 196, 197]. In particular, a baseband prototype filter, which is used to obtain the baseband signal, is initially designed. Its frequency is then shifted to other frequency bands in order to estimate them via downsampling, which subsequently helps the method to cover all the WB signal. The estimated individual frequency bands, which become NB signals, are then sensed with a NB spectrum sensing algorithm to determine the presence of PUs [190, 191, 196, 197]. In this regard, the authors in [108] design a multi-band spectrum sensing model through the filter-bank approach with PHYDYAS and prolate sequence window as prototype filters.

Sub-Nyquist-based Approaches: Although the aforementioned Nyquist-based WB spectrum sensing methods are successful in detecting spectrum opportunities, they are often costly in terms of sampling rate, hardware, and implementation complexity [196, 198]. As such, sub-Nyquist-based spectrum sensing techniques, where the sampling rate of the receiver ADC is less than the Nyquist rate, come forward due to their more efficient and cost-effective approach. By following the top-level taxonomy provided by [196], the sub-Nyquist-based approaches are elaborated as follows:

- **Compressive sensing based WB spectrum sensing:** after the introduction of compressive sensing [199], which offers the reconstruction of sparse signals through sampling below the Nyquist rate, a broad range of application areas have emerged [200]. Similarly, it has also been recognised as an important spectrum sensing method owing to the sparsity of the WB spectrum [196]. In compressive sensing, first, the sparse representation of the original signal is needed in a basis that is incoherent with the measurement matrix, which is obtained through random sensing [198–200]. After the signal acquisition is executed through sparse representation, sampling, and compression, the signal is supposed to be reconstructed using one of the Bayesian, greedy, thresholding, etc., approaches [198,200]. In [109], a compressive sensing based WB spectrum sensing approach is proposed, where the authors use relative inner product norm value to find the correlation between the measured signal and the sensing matrix, which is then used to detect the spectrum holes.

- **Multichannel sub-Nyquist WB spectrum sensing:** multi-coset sensing is one of the most popular multichannel sub-Nyquist based WB spectrum sensing approach. In this method, the WB signal goes through multiple sampling channels, whose time offsets are different from each other. In particular, each sampling channel (or coset) has a sampling frequency of $f_{\mathrm{s}}/\mathfrak{m}$, where f_{s} is the Nyquist rate and $\mathfrak{m}$ is the decimation factor [201,202]. Then, the average sampling rate becomes $\dfrac{f_{\mathrm{s}}N_{\mathrm{cs}}}{\mathfrak{m}}$, where N_{cs} is the number of sampling channels. Satisfying $N_{\mathrm{cs}} < \mathfrak{m}$, the number of measurements becomes less than that of the Nyquist case, which is beneficial in decreasing the complexity of the system. In short, in multi-coset based spectrum sensing, each sampling channel operates at $\mathfrak{m}$ times lower sampling frequency than the Nyquist rate, and—by providing $N_{\mathrm{cs}} < \mathfrak{m}$—less measurements are taken into account compared to the Nyquist rate sampling. In this regard, a multi-coset oriented blind spectrum sensing method is developed in [110] for WB signals, wherein the authors identify the spectrum holes directly from the sub-Nyquist sampling sequence with the help of two different thresholding concepts for detection and terminating the iterative search process.

Limitations: Compared to the NB case, WB spectrum sensing offers more spectrum opportunities, since it covers a larger spectrum portion. In other words, there becomes more opportunities to find suitable vacant spectrum holes as the bandwidth to be sensed increases. On the other hand, this advantage comes at the expense of some design challenges and inherent limitations. For the Nyquist-based WB spectrum sensing techniques, for example, the major problems are the requirement for sampling at higher rates and complexity of implementation [191, 196]. Moreover, IQ imbalance and aliasing are also issues, as raised by [107], especially for the case of multiband spectrum sensing. In sub-Nyquist based spectrum sensing, on the other hand, even though the higher sampling rate requirements are circumvented, the design of sensing matrix, recovery uncertainty, and hardware implementations are the main issues with compressive sensing [198], while multichannel sub-Nyquist WB spectrum sensing suffers from requiring many cosets [196].

2.5.3 Predictive Spectrum Sensing Approach

Predictive spectrum sensing is the process of forecasting or estimating the availability of spectrum holes, that is, the presence of the PUs in a licensed spectrum, in advance using historical data, so that the SUs can make informed decision on when to utilise the spectrum or vacate it for the PUs [203, 204].

The advantages of predictive spectrum sensing approaches over traditional counter-parts are as follows [60, 205]:

- **Reduced sensing latency:** since the SUs already have prior knowledge of possible vacant channels through spectrum predictions, they spend less time in sensing to detect the available spectrum holes. This is because, with the help of spectrum predictions, the SUs only sense the channels that are predicted to be vacant.

- **Proactive and real time decision making:** in traditional sensing methods, the spectrum sensing is first implemented, followed by the decision of which channel to occupy. However, before this process is completed, the channel condition might have changed as the previously sensed spectrum hole may now be occupied, thereby leading to poor spectrum decision and channel collision. With predictive spectrum sensing, the channel occupancy status is known before hand, thereby reducing the delays associated with spectrum sensing and decision making process [206].

- **Throughput enhancement:** by predicting the channel occupancy status in advance, the SU is able to select the channel with better quality in order to enhance its transmission throughput.

- **Energy efficiency improvement:** the process of spectrum sensing is energy consuming, hence the lesser the spectrum sensing time, the more energy can be conserved. Therefore, through predictive sensing approaches the sensing time can be greatly reduced which would results in energy savings for the CR devices.

State-of-the-Art in Predictive Spectrum Sensing

Predictive spectrum sensing approaches have gained considerable amount of interest owing to the fact that it allows a more informed sensing process, increasing the chances of finding a spectrum hole [60]. This, by its turn, reduces the sensing latency and energy consumption. In this regard, there are numerous related works available in the literature.

An ANN based spectrum prediction approach is proposed in [95]. The developed predictor is consolidated by minimum Bayesian risk concept, which tries to minimise the cost function that can be broadly defined as the difference between a variable and its estimation. In particular, the authors, first, design an ANN based model in order to predict the status of a frequency channel in the next time slot. A sequence of previous status information of the channel of interest—in binary form—feeds the ANN model as input features, where each previous time slot becomes a feature. Then, the channel status for the next time slot is supposed to be predicted; however, the authors argue that a fixed type of decision boundary for detecting the presence of PU results in misdetections in the case of error inclusions in the data. Therefore, instead of using the prediction outputs directly for detection, minimum Bayesian risk phenomena is applied with an assumption that the prediction outputs fit to the normal distribution. In this approach, a dynamic decision boundary, which is updated with new measurements, is proposed, and this model is proven to give better prediction results than that of a conventional ANN-based spectrum prediction.

A hidden Markov model (HMM) based cooperative spectrum sensing technique is proposed in [96], with a focus of energy-limited devices, such as the sensor nodes in WSNs. In that regard, the authors, first, define interference zones, which are distinctive in a way that each interference zone specifies a region where different combination of SUs causes interference with the PU. These zones are then treated as the hidden states for the developed HMM model, and the objective becomes predicting the next state of the PU. More particularly, each SU transfers its binary spectrum sensing result to a fusion centre, which is equipped with an HMM predictor. After that, the fusion centre uses these spectrum sensing results obtained from the SUs as observations in order to build the HMM model. According to the predictions, the SUs that would cause interference with the PU are refrained from sensing activity, however, instead of allowing all the rest of the SUs to sense the spectrum, some of them are selected by considering the battery conditions. The proposed method results in an improved throughput-energy rate, which can be interpreted as achieving the same throughput with less energy consumption.

The authors in [97] use LSTM to predict the occupancy states of frequency channels, where—different from previously mentioned related works—spectrum sensors that are responsible for sensing the EM spectrum are employed. The idea behind having these spectrum sensors is to take the burden of sensing process from the SUs. In particular, the spectrum sensors deployed over a wide area are responsible for sensing, then, multi-step ahead spectrum occupancy predictions are performed with the help of LSTM-based local predictors. For a specific SU, the prediction results are then fused in a way that the prediction of the spectrum sensor with the closest proximity to the SU-of-interest has more impact due to the spatial correlations. By doing so, the proposed model performs better in terms of prediction accuracy and error probability.

A comprehensive spectrum sharing approach, which includes predictive spectrum sensing task, is proposed in [98], with a main objective of maximising the sum rate obtained by all the SUs by taking their minimum rate requirements into consideration. As such, all the SUs and the secondary BS carry out spectrum predictions by employing a multi-layer perceptron (MLP) algorithm, where historic observation records are utilised as inputs. Then, the prediction outputs from all the SUs and the secondary BS are forwarded to a fusion centre, which is responsible for fusing what it receives in order to obtain a global prediction result. Given that the prediction results produced by all the SUs and the secondary BS have the same priority, the majority rule[6] is determined as the rule for fusion. In order to achieve the maximum sum rate objective, two beamforming vectors (one for when PU is present and the other when PU is absent) and sensing time are considered as optimisation variables.

A spectrum access technique is demonstrated in [99], where spectrum predictions are performed using an LSTM method. Instead of predicting the future occupancy status of the channels, in their work, the authors execute multi-step

[6] The prediction output that is returned by the majority of the entities (SUs and the secondary BS) is selected as the global prediction output.

future power level predictions for multiple frequency channels. The predicted power levels are then compared with a power threshold by considering an energy detector based spectrum sensing. Based on the predefined power threshold, the binary status of each channel at each time slot is converted to a binary sequence, followed by categorising the channels based on their future availability. After removing the channels in the *unsuitable for sensing* category, the rest of the channels are sorted according to their throughput, which are then sensed by the energy detector method sequentially. An interesting contribution is that the authors optimise the sensing interval to adjust the output length of designed LSTM.

A genetic algorithm assisted MLP based spectrum prediction method is developed in [100], in which the authors criticise the initial random weight selection process in conventional MLP process, and argue that this results in sub-optimality and trapping in local minimum. Therefore, a genetic algorithm based initial wight selection algorithm, where the fitness function is chosen to be the prediction error, is developed for the MLP spectrum predictor. Provided that the prediction error is a fitness function, the algorithm tends to select the weights with comparatively less errors. The proposed genetic algorithm assisted MLP is compared with a conventional MLP (without genetic algorithm inclusion), and the prediction accuracy is found to be enhanced with the proposed initial weight selection.

In addition to HMM, MLP, and LSTM, convolutional neural networks (CNNs) have also been applied in predictive spectrum sensing [207]. In this regard, a fusion centre based predictive spectrum sensing method, which employs CNNs as a predictor, is designed in [101]. Since the SUs in the considered scenario are not capable of sensing the spectrum, this task is performed by data fusion centres. The focus of the work is to exploit the spectrum opportunities from WB while sensing NB spectrum. In other words, first, the occupancy status of WB is predicted, and then only the frequency channels that are predicted to be vacant become subject to the spectrum sensing process. Moreover, an online learning process—where the training with new data is always ongoing even after the prediction algorithm is plugged into the system—is introduced in their work in order to keep the CNN predictor dynamic to the changes.

2.5.4 Research Gap Analysis

Predictive spectrum sensing techniques have already resolved many problems in CRNs, since they build a bridge between WB and NB sensing methods by taking the advantage of both. While the spectral opportunities from WB is exploited, the simplicity and robustness of NB sensing methods can also be benefited. Provided that, with spectrum predictions, SUs conduct more informed sensing process, the sensing latency and energy consumption can be reduced due to the fact that the required number of sensing attempts decay by narrowing down the spectrum to be sensed.

Nonetheless, a majority of the existing works rely on historic channel state knowledge for each channel of interest, making their applicability questionable. This limits the number of channels to be predicted, and thus the underlying idea behind the predictive sensing methods is partly violated. Moreover, the bandwidth of interest can be further minimised by considering more context-awareness during sensing. In other words, in most of the existing works, the decision of selecting the channels to sense is made only considering the occupancy states, such that the one with the most vacancy probability is first sensed. However, that channel (or band) may not fit the requirements of an SU, and thus it is counted as an unnecessary sensing attempt even if the spectrum hole in found. Therefore, a pre-election mechanism would be needed between the prediction and sensing phases, helping in eliminating the bands that are unsuitable for allocation. A more detailed research gap analysis can be found in Chapter 5.

Chapter 3

Energy Optimisation in Cellular Networks

Both device-side and network-side energy optimisation techniques are provided in this Chapter with separate discussions. More particularly, in the device-side, wireless connectivity and data processing location of IoT nodes are optimised considering the battery levels as well as the requirements of the devices, as in [208, 209]. In the network-side, on the other hand, an intelligent cell switching concept is designed in order to minimise the energy consumption of cellular networks while maintaining service qualities.

3.1 Energy-Aware Smart Connectivity for IoT Networks

IoT is today's buzzword, often coupled with big data and AI [208]. However, there is a lot of ambiguity of what is meant by that and scepticism about the actual value generated by the IoT. Moreover, IoT devices have become pervasive, but cover a broad range of technologies and standards. Wireless technology is key to connect these devices through gateways or aggregation points, however, similarly, a wide range of wireless protocols and standards are available and competing [210]. Once these devices are connected, they start reporting the sensed or measured data to the platform. Again, multiple choices are possible in this aspect with different strengths and weaknesses: reporting raw data to the cloud is very costly as every bit gets charged. On the other hand, running scripts locally in the device and reporting the resulting events to the cloud reduces the cloud service cost, while limiting the visibility to the actual data and exhausting the battery of the device[1]. Furthermore, local scripts result in real-time actions and do not expose the privacy of the data, whereas cloud computing incurs latency due to the transmission network and requires stringent security measures to protect the data [208].

Since there are many use-cases of the IoT paradigm [212], it should be approached from a given vertical perspective; e.g., smart health, smart cities, smart manufacturing (Industry 4.0), smart transport, etc. Each of these verticals comprises multiple IoT-based applications with various requirements. In [213], for example, signalling measurements and modelling are performed for both static and vehicular machine-to-machine (M2M) applications, as both have different signalling overhead characteristics. As another example, remote monitoring in smart cities requires full compliance with privacy regulations, whereas security-related applications rank response-time highest among all KPIs.

The smart port use-case is adapted in this work to demonstrate the context-aware smart connectivity, since it includes various types of applications and has a determined need for monetisation—as opposed to smart cities that are primarily developed for the well-being and productivity of the society. According to figures from the World Trade Organization, 80% of worldwide freight is transported through ports[2]. The smart port concept entails the use of technologies to transform the different public services at ports into interactive systems with the purpose of meeting the needs of port users with a greater level of efficiency, transparency, and value. European smart port initiatives include the followings among many others [208]:

- the port of Rotterdam, where IoT-sensors are employed to generate a digital twin and enable augmented intelligence[3];
- the port of Hamburg, which exploits 5G networks to enable virtual reality for vital infrastructure monitoring[4];
- the port of Antwerp employs blockchain technology to enable a secure transfer of rights to be exchanged between often competing parties[5];

[1] The energy consumption of an IoT device has two main components: data processing and transmission. Based on the channel conditions and the distance between the transmitter and receiver, one of these components can dominate the other. Therefore, processing the data locally could be less energy consuming in some conditions, while it results in more energy consumption in other cases. Interested readers are referred to [211] and the references therein.

[2] www.wto.org/.

[3] www.portofrotterdam.com/en/doing-business/port-of-the-future/digitisation/control-management, accessed on 03/05/2020.

[4] www.hamburg-port-authority.de/en/themenseiten/monarch-5g/, accessed on 03/05/2020.

[5] www.portofantwerp.com/en/news/smart-port-blockchain, accessed on 03/05/2020.

- the port of Seville through the Tecnoport 2025 project uses mobile network technology for traffic and goods tracking on port and their logistical transfer on land[6].

Smart ports present a particular challenge due to the necessity of information exchange among competing stakeholders including port authorities and operators; terminal operators; logistics and shipping companies; etc. It is then likely that multiple IoT networks would co-exist and would consist of partly private and partly public or shared infrastructure.

As described in [214], there are various communication standards, with different strengths and weaknesses, that may be used for connecting IoT networks in the context of smart ports. Mobile IoT—connectivity over licensed mobile wireless networks—is often the preferred solution for handling private data, since it is reliable, end-to-end secure (owing to the electronic subscriber identity module (eSIM) card), scalable, ubiquitous, and mature. Two main technologies have been introduced by mobile networks to connect IoT devices: eMTC and NB-IoT [27]. Both of these technologies are compatible with LTE, which means that a software update suffices to deploy the IoT options. The former is geared towards higher rates (>1 Mbps), and supports VoIP (Voice over IP based on ITU H.323 protocol[7]) and flexible mobility. The latter is designed for low data rates (20 kbps) and long range (100 km) but with limited mobility. The NB-IoT technology consists of restricting the energy of an LTE normal carrier in a narrow band, hence allowing a maximum coupling loss that is 20 dB higher (164 dB) than LTE [215]. Mobile IoT is a public service enabled by telecom carriers and may be used by any party who subscribes to it. Other long-range and low-power solutions, such as LoRa and Sigfox, are unlicensed and can reach similar coverage and data rates as NB-IoT and eMTC. These may be privately owned but require the usage of a gateway to connect to the Internet, and are often considered less secure. Many short range unlicensed wireless connectivity solutions are available, such as Wi-Fi (IEEE 802.11g), Bluetooth, ZigBee, etc., as described in [216], and may be shared, public, or private.

In the presence of multiple wireless technologies, disparate IoT applications, competing parties, and a broad range of static and moving IoT devices with multiple connectivity options, it is of key importance to identify the best way to collect, store, cache, and process the IoT data. What qualifies as the best way depends on the device capabilities (e.g., connectivity options, available battery);

[6] www.gsma.com/iot/news/sevilles-tecnoport-2025-project-adopts-iot-security-guidelines/, accessed on 03/05/2020.

[7] https://www.itu.int/rec/T-REC-H.323/e.

the wireless conditions; the security requirements; the processing complexity and availability; and the cost of storage/caching/uploading; etc.

3.1.1 Related Work

As the energy consumption is one of the challenges for IoT networks [217], recent works, such as [211] and [218], study the trade-off between local and cloud computing in terms of device energy consumption. The former proposes an analytical framework that minimises the energy consumption by optimising the offloading decision of multiple user devices. The latter elaborates a theoretical framework for establishing trade-offs in the energy consumption and IoT infrastructure billing comprising cloud computing. Mobile wireless networks are a prime contender in the race to connect IoT networks owing to their well-established and ubiquitous coverage and secure communication based on the eSIM card. In [219], the authors investigate the connectivity of NB-IoT and LoRa in terms of both area and population coverages in order to highlight the importance of the network deployments. In [220], big data analytics based user-centric smart connectivity is argued by providing corresponding research challenges.

Although data aggregation seems a promising solution to ease the signalling overhead, it is one of the causes of the transmission delay. In [221], the authors discuss the trade-off between delay and signalling overhead in order to demonstrate the impacts of data aggregation. The authors in [222] analyse the joint optimisation of caching and task offloading in such networks with mobile edge computing. They present an efficient online algorithm based on Lyapunov optimisation and Gibbs sampling that succeeds in reducing computation latency while keeping the energy consumption low. A recommendation system is proposed in [223] to address the challenge of link selection in a cloud RAN. A data-driven scheme is introduced that results in optimised classification of link strengths between RRHs and IoT devices.

A deep learning algorithm for edge computing is introduced in [224] to boost the learning performance in IoT networks. They also attempt to increase the amount of edge tasks by considering the edge capacity constraints. An open-source database is designed in [225] for the edge computation of industrial IoT (IIoT) networks. The authors use a time-series analysis for predicting conditions of IIoT machines in order to decrease the amount of condition reports to be sent to the cloud. A holistic view of communication, computation, and caching is presented in [226] using graph-based representations as learning methods for innovative resource allocation techniques. The performance of the edge-caching as well as the energy efficiency and delivery time is investigated in [227] with QoS constraints.

3.1.2 Objectives and Contributions

An environment, which is rich in IoT devices that are connected to a platform, qualifies as digitised, and often as intelligent. Analytics, which use AI, is the added layer that transforms such an environment into a smart one. The default application of AI is to draw actionable insights from data in order to generate value to a given vertical. This work argues that IoT solutions should not be addressed through a layered perspective, but, instead, a holistic optimisation approach is needed to generate the desired added value efficiently. In such a holistic approach, ML, among other AI tools, is employed in every stage of the solution including connectivity, storage, computing, and analytics.

In this regard, RL is employed in this work in order to manage multiple optimisation objectives jointly as follows:

- **energy consumption and monetary cost:** based on the battery conditions of IoT devices, the objective is to minimise the total monetary cost (incurred by fog and cloud processing) along with the total energy consumption of a device (incurred by data processing and transmission).
- **requirements of IoT devices:** the requirements of devices, such as latency, security, etc., are also included in the optimisation problem, of which meeting the given requirements is made one of the objectives.

Hence, the followings are performed in order to deal with the aforesaid objectives:

- **quality features:** four key quality features, dominating IoT applications in general and smart ports in particular, are identified as security, energy consumption, latency, and monetary cost.
- **problem formulation:** to enhance the aforementioned quality features, dynamically selecting the best connection and data processing unit for each device is identified as a problem to combat. Moreover, the amount of data to be offloaded is also subject to an optimisation, since it plays an important role in selecting the connectivity and data processing unit.

This work is one of the few attempts to address these multiple IoT optimisation objectives jointly using RL. Different from the state-of-the-art research, a joint optimisation of the wireless connectivity, processing unit, and the percentage of data to be offloaded is proposed by considering the energy consumption,

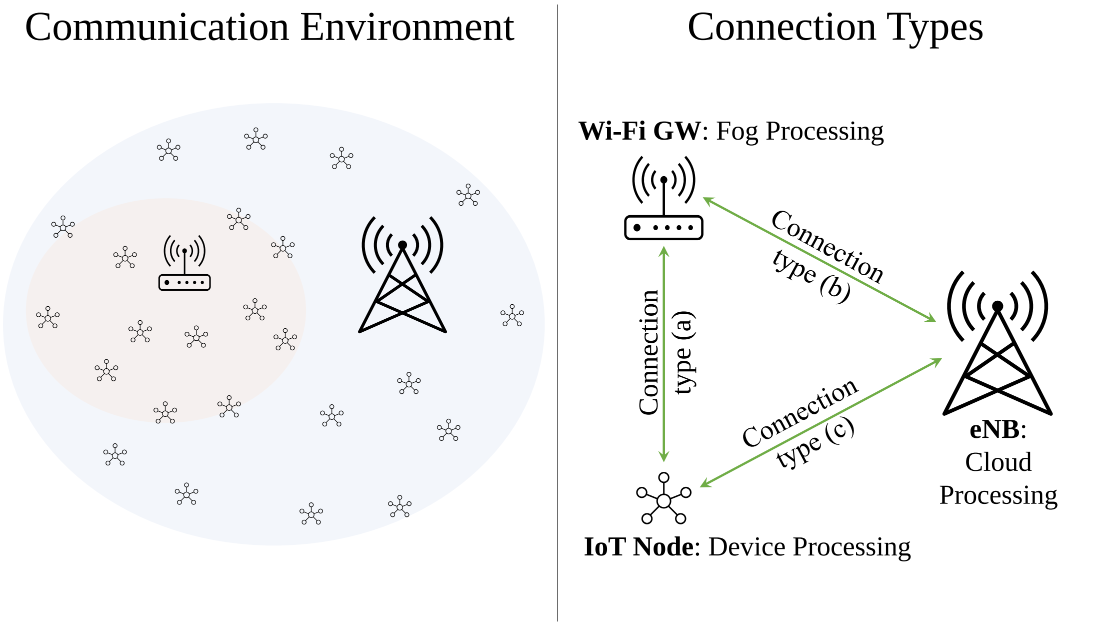


Figure 3.1: System modelling. The IoT network scenario is depicted on the left hand side, while the descriptions of the considered wireless connection and processing unit types are demonstrated on the right hand side. GW: gateway, eNB: evolved node-B, IoT: Internet of things.

response-time, security, and monetary cost. The proposed novel approach is compared to the state-of-the-art connectivity solutions, and demonstrate significant gains in all aspects. Moreover, the proposed approach is a concrete solution that is able to meet the context-aware requirements fully, while minimising the monetary cost and the energy consumption.

3.1.3 System Model

The novel energy-aware smart connectivity approach proposed in this work applies to any IoT network with diverse options of connectivity and processing. For the sake of clarity in the presentation, the system model is built around a smart port scenario. All IoT devices are battery operated and have different battery lives. They all have some processing power to perform basic tasks, and can either offload the task to the gateway (or fog); i.e., the Wi-Fi access point, or to the evolved node B (eNB or cloud). The considered system model is illustrated in Fig. 3.1, where a sample of simulated IoT network environment and a visual description of the considered wireless connection types along with processing units are presented.

It is assumed that every IoT device is controlled by a given application and they jointly determine the context-aware constraints. Each combination of connectivity option and processing location offers specific characteristics and limitations. In the following paragraphs, the models adopted to capture the propagation loss, energy consumption, and response time for the proposed system are described.

Propagation model

As shown in Fig. 3.1, there are three wireless connections that require modelling: (a) Device-to-Gateway (Wi-Fi), (b) Device-to-eNB (NB-IoT), and (c) Gateway-to-eNB (LTE). Connection (a) is often interference limited, as the employed spectrum is likely to be shared by other neighbouring connections. Connections of type (b) and (c) are, however, considered to be noise limited, as it is assumed that there are no other eNB in the surrounding employing NB-IoT technology and a scheduler employed for LTE connections. The objective of the propagation modelling is to determine the transmission power required to cater for each of the wireless connection types. Accordingly, the energy consumption will be calculated. It is started with the propagation loss, PL, which follows log-distance path-loss model as follows:

$$PL = PL_0 + 10\varkappa \log_{10}\frac{d}{d_0} + X_{\mathrm{g}}, \tag{3.1a}$$

$$PL_0 = 20\log_{10}\left(\frac{4\pi d_0}{\lambda}\right), \tag{3.1b}$$

$$\lambda = \frac{c}{f_{\mathrm{c}}}, \tag{3.1c}$$

where $\varkappa$ is the path-loss exponent, d is the distance between the transmitter and receiver with d_0 being the reference distance. X_{g} is the shadowing component with a standard deviation, σ, and zero mean. PL_0 is the path-loss at the reference distance d_0, λ is the wavelength, c is the speed of the light, and f_{c} is the transmission frequency.

Moreover, although the connection types suffer from the same propagation loss per decade, their receiver sensitivities are different from each other[8]. For all types of links, the received power at a distance d_x from the transmitting device can be expressed as $\mathcal{P}_{\mathrm{r}} = \mathcal{P}_{\mathrm{t}}/PL$. Next, the required received power, $\mathcal{P}_{\mathrm{r}}$, is calculated in order to achieve the target data transmission $\mathcal{D}$ in bits:

$$\mathcal{D} = TW\log_2\left(1 + \frac{\mathcal{P}_{\mathrm{r}}}{\mathcal{P}_{\mathrm{I}} + \mathcal{N}_0 W}\right), \tag{3.2}$$

[8] Threshold receiver sensitivity is taken as −141 dBm, −121 dBm, and −82 dBm for NB-IoT, LTE, and Wi-Fi connections, respectively. In this work, these differences in the receiver sensitivities are captured as link margins, such that the more sensitive the receiver gets the more link margin is incurred.

where T is the time period, W is the channel bandwidth, and $\mathcal{P}_\mathrm{I}$ is the cumulative interference power on the given channel during time period T. Please note that $\mathcal{P}_\mathrm{I}$ is null for wireless connections of type (b) and (c). Using (3.2) and solving for $\mathcal{P}_\mathrm{r}$:

$$\mathcal{P}_\mathrm{r} = \left(2^{\mathcal{D}/(TW)} - 1\right)\left(\mathcal{P}_\mathrm{I} + \mathcal{N}_0 W\right). \tag{3.3}$$

The energy consumption is optimised in this work along with monetary cost and requirements of IoT devices, and data transmission is one of the main components of the total energy consumption of an IoT device. Therefore, these calculations for the transmission power consumption are performed in order to obtain the energy consumption, which is then optimised accordingly. The summary of this process is given in the following paragraphs.

Energy consumption model

There are two major processes that consume energy in an IoT network: wireless transmission and task computation. The energy consumption of the former is E_t and the latter is E_p, thus the total energy consumption is the sum of both. Depending on the route of communication taken by the device, the energy consumed due to transmission power can be a result of either one hop using NB-IoT ($\mathsf{E}_\mathrm{t,b}$) or two hops using Wi-Fi for the first link and LTE for the second ($\mathsf{E}_\mathrm{t,a} + \mathsf{E}_\mathrm{t,c}$).

The process of calculating E_t followed during this work can be compiled as follows: first, $\mathcal{D}$—a finite-length data packet to be transmitted during the time period T—is obtained for each IoT device. Second, in order to transmit $\mathcal{D}$ successfully, the required received power ($\mathcal{P}_\mathrm{r}$) is computed via (3.3), which is derived from (3.2). Third, based on the obtained $\mathcal{P}_\mathrm{r}$ value—for each IoT device—the required transmit power ($\mathcal{P}_\mathrm{t}$) is calculated using (3.1). Lastly, $\mathcal{P}_\mathrm{t}$ is observed for a period of time T in order to arrive at the energy consumption caused by data transmission. The energy consumed for processing the task, on the other hand, is a function of the volume of data ($\mathcal{D}$), computational power of the processing unit[9] ($\Gamma \in \{\Gamma_\mathrm{d}, \Gamma_\mathrm{f}, \Gamma_\mathrm{c}\}$)[10], and the energy consumption per computational cycle (ϵ), such that $\mathsf{E}_\mathrm{p} = f(\mathcal{D}, \Gamma, \epsilon) = \dfrac{\mathcal{D}}{N_\mathrm{bit}}\Gamma\epsilon$, where N_bit is the number of bits per data element [211].

[9] It is measured in the number of computational cycle per data element; i.e., higher Γ yields less computational power.

[10] Γ_d, Γ_f, and Γ_c are the computational powers of device, fog, and cloud, respectively.

Response time model

The response time perceived by the IoT device is the combination of the uplink and downlink delays between the IoT device and the server. In this work, the uplink delay is modelled, while the downlink delay is assumed the same for all devices.

The uplink delay is caused by two phenomena: task processing (processing delay, t_{p}) and data transmission (transmission delay, t_{t}). The processing delay depends on the processor's computational power (Γ). Naturally, a server has higher computational power than a small gateway, and much higher than a simple IoT device ($\Gamma_{\mathrm{c}} < \Gamma_{\mathrm{f}} < \Gamma_{\mathrm{d}}$). Thus, in this work, t_p is modelled based on the computational powers of the processing locations as follows: $\frac{t_{\mathrm{p,d}}}{\Gamma_{\mathrm{d}}} = \frac{t_{\mathrm{p,f}}}{\Gamma_{\mathrm{f}}} = \frac{t_{\mathrm{p,c}}}{\Gamma_{\mathrm{c}}}$, where $t_{\mathrm{p,d}}$, $t_{\mathrm{p,f}}$, and $t_{\mathrm{p,c}}$ are the task processing delays for device, fog, and cloud, respectively. In addition, while the input to the task processing stage is large raw data, the output is compressed data with comparably less volume. To that end, the compression rate between the input and output data volumes is given as $\mho$; $\mathcal{D}_{\mathrm{r}} = \mho\mathcal{D}_{\mathrm{p}}$, where $\mathcal{D}_{\mathrm{r}}$ and $\mathcal{D}_{\mathrm{p}}$ are the volumes of raw and processed (compressed) data, respectively.

The transmission delay is affected by the type of RAT and the volume of data to be transmitted. Since Wi-Fi access employs the unlicensed frequency bands, it often suffers from higher retransmission rates, which results in increased transmission delays, due to frequent collisions. Therefore, in this work, this effect is captured by the factor $F > 1$, whereby the delay incurred for transmitting the same volume of data over Wi-Fi is F times higher than that over LTE or NB-IoT; $t_{\mathrm{t,a}} = t_{\mathrm{t,b}}F = t_{\mathrm{t,c}}F$, where $t_{\mathrm{t,a}}$, $t_{\mathrm{t,b}}$, and $t_{\mathrm{t,c}}$ are the transmission delays for connection types (a), (b), and (c), respectively. This model is represented in Fig. 3.2, in which the source could be either the IoT device or the gateway, and the recipient could be either the gateway or the cloud.

Consequently, the overall response time for each action is calculated as follows:

$$R = t_{\mathrm{p}}\mathcal{D} + \sum_{i=1}^{N_{\mathrm{h}}} t_{\mathrm{t},i}\mathcal{D}_i, \tag{3.4}$$

where $N_{\mathrm{h}} = \{1, 2\}$ is the number of hops, and $\mathcal{D} \in \{\mathcal{D}_{\mathrm{r}}, \mathcal{D}_{\mathrm{p}}\}$. Besides, $t_{\mathrm{t},i}$ and $\mathcal{D}_i$ represent the values of t_{t} and $\mathcal{D}$ for the i^{th} hop, respectively. Then, a feature scaling is applied to the values calculated through (3.4)[11] as:

$$f(x) = \frac{x - \min(X)}{\max(X) - \min(X)}, \tag{3.5}$$

where X is the set of values of x. Note that both (a) and (b) type connections constitute the first hop, while the connection type (c) is the second hop.

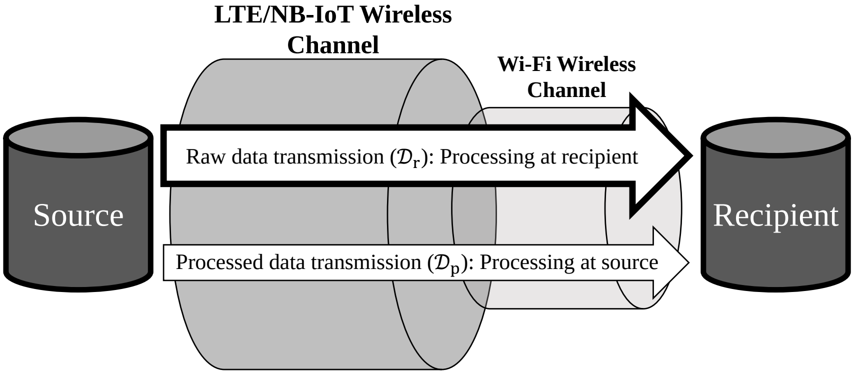

LTE/NB-IoT Wireless Channel
Wi-Fi Wireless Channel
Source
Raw data transmission ($\mathcal{D}_\mathrm{r}$): Processing at recipient
Processed data transmission ($\mathcal{D}_\mathrm{p}$): Processing at source
Recipient

Figure 3.2: Uplink delay model capturing the factors affecting both processing and transmission delays over any hop in our system.

3.1.4 Problem Formulation

All the possible connection type and processing unit pairs are given in Table 3.1. Note that although device, fog, and cloud processing are all available for Wi-Fi case, NB-IoT includes only device and cloud processing owing to the fact that it does not have/require a Wi-Fi gateway to connect to the Internet.

Table 3.1: Possible connection type and processing unit options

Option	**Connection Type**	**Processing Unit**
A	Wi-Fi	Device
B	Wi-Fi	Fog
C	Wi-Fi	Cloud
D	NB-IoT	Device
E	NB-IoT	Cloud

Considering the diversity in the available options provided in Table 3.1, there are multiple components and objectives of the developed optimisation problem:

[11] x is used here as an arbitrary variable to indicate that the function given in (3.4) can be used for any parameter.

- **requirements of IoT devices:** there could be a broad range of requirements based on the use-case, scenario, and conditions, such as data rate, security, latency, etc. to name a few. In this work, response time—as another interpretation of latency—and security are considered as possible requirements of IoT devices. As such, let Σ_{r} be the response time requirement of an IoT device, and $\mathcal{O}_{\mathrm{r}}$ be the response time offered by the selected option. Therefore, in order to satisfy the response time requirement of the IoT device, the following criterion must be met:

  $$\mathcal{O}_{\mathrm{r}} \leq \Sigma_{\mathrm{r}}. \tag{3.6}$$

  The security requirement, on the other hand, is captured by eSIM protection, such that IoT devices opt for eSIM protection if data security is of importance to them. In a more formal way, let $\Sigma_{\mathrm{s}} \in \{0, 1\}$ be the security requirement of an IoT device, and $\mathcal{O}_{\mathrm{s}} \in \{0, 1\}$ be the level of data security offered by the selected wireless technology, where 1 indicates the need for eSIM protection and 0 means eSIM protection is unnecessary. In this regard, the following condition is needed in order to meet the security requirement of the IoT device:

  $$\mathcal{O}_{\mathrm{s}} \geq \Sigma_{\mathrm{s}}. \tag{3.7}$$

- **energy consumption:** the total energy consumption of an IoT device ($\mathsf{E}_{\mathbb{T}}$), which consists of E_{t} and E_{p}, should be minimised in order to keep the device alive for longer. However, the requirements of IoT devices may undermine this objective, since the number of available options may reduce in an attempt to satisfy the requirements. Furthermore, the battery level of an IoT device, represented by β, is also considered in this work provided that it could be another variable in the optimisation problem, such that the energy consumption of the device can be prioritised if β goes low.

- **monetary cost:** it is also important to reduce the monetary costs due to the fact that a smart port scenario, which is more related to trading and business, is targeted in this work. Since lower costs is vital to keep businesses sustainable and profitable, the total data processing cost, represented by $M_{\mathbb{T}}$, is supposed to be minimised as well, constituting another objective for the optimisation problem. To this end, each processing unit incurs different costs: M_{d}, M_{f}, and M_{c} are the processing cost per bit for device, fog, and cloud processing, respectively.

In addition to the requirements and objectives, the optimisation problem has strict constraints as well. For example, the available computational capacity of the processing unit is the major constraint, since it can render the selection infeasible. An IoT device, for instance, selects Option C from Table 3.1, which requires offloading the collected raw data to the cloud. However, if the cloud does not have sufficient computational capacity to handle the data processing task, the selection becomes invalid. In this regard, let $\mathcal{O}_{\chi} \in \{\chi_{\mathrm{d}}, \chi_{\mathrm{f}}, \chi_{\mathrm{c}}\}$ be the computational capacity of the selected processing unit, where χ_{d}, χ_{f}, and χ_{c} are the available capacities of device, fog, and cloud, respectively. Then, the following condition must be obeyed to ensure that the selected processing unit has enough capacity for the required amount of data:

$$\mathcal{D}_{\mathrm{r}} \leq \mathcal{O}_{\chi}. \tag{3.8}$$

It is worth nothing that network capacity is not considered as a constraint in this work, and it is assumed that the IoT devices can obtain their required radio resources from the network, regardless of the connection type. In other words, the network has sufficient radio resources to connect the IoT devices.

In addition to the computational capacity, the aforementioned requirements of IoT devices create additional constraints due to the fact that poorly addressed response time and security requirements can make the whole process impractical. For example, meeting the security requirement can be a must for some use-cases, which value data privacy and cannot tolerate data breaches. In this regard, for such use-cases, Options A, B, and C in Table 3.1 are eliminated from the possibilities, since they are with Wi-Fi connectivity that does not offer an eSIM protection.

Moreover, partial offloading—where IoT devices are allowed to offload portions of their raw data to the fog or cloud—is also considered in this work. As such, the amount of data to be offloaded should also be optimised, thereby a joint optimisation of connection-processor pair and amount of data to be offloaded is proposed. The reasoning behind the partial offloading concept is that it is not sufficient to optimise the best connection-processor pair alone, since this kind of optimisation cannot be done properly without considering and optimising the amount of data to be offloaded given the aforementioned constraints. For example, imagine an IoT device selects Option C in order to reduce its energy consumption. However, if the amount of data to be offloaded had not been a part of the decision, it means that the decision was taken blindly. Then, imagine the cloud is capable of processing only 10% of the data due to its capacity limitations, while the fog could process 100% of the data, which happens to result in less energy consumption than processing 10% at the cloud. It is quite clear from this example that the amount of data to be offloaded should also be a part of the optimisation.

In this regard, the overall optimisation problem can be written formally as follows:

$$\begin{aligned} \min_{\mathcal{O},\Psi_{\text{sel}}} \quad & \mathsf{E}_{\mathbb{T}}(\mathcal{O},\Psi_{\text{sel}}), M_{\mathbb{T}}(\mathcal{O},\Psi_{\text{sel}}) \\ \text{s.t.} \quad & \mathcal{O}_{\text{r}} \leq \Sigma_{\text{r}}, \\ & \mathcal{O}_{\text{s}} \geq \Sigma_{\text{s}}, \\ & \mathcal{D}_{\text{r}} \leq \mathcal{O}_{\chi}, \end{aligned} \tag{3.9}$$

where Ψ_{sel} is the selected percentage volume of data to be offloaded. $\mathcal{O} \in \{\text{A},\text{B},\text{C},\text{D},\text{E}\}$ is the selected option from Table 3.1, and is a 5-tuple as follows:

$$\mathcal{O} = [\mathcal{O}_{\text{r}}, \mathcal{O}_{\text{s}}, \mathcal{O}_{\chi}, \mathsf{E}_{\mathbb{T},\mathcal{O}}, M_{\mathbb{T},\mathcal{O}}], \tag{3.10}$$

where $\mathsf{E}_{\mathbb{T},\mathcal{O}}$ and $M_{\mathbb{T},\mathbb{O}}$ are the resulting total energy consumption and monetary cost offered by the selected option, $\mathcal{O}$, respectively.

3.1.5 Proposed Scheme

RL, which is a goal-seeking technique, is employed in this work. It is a trial and error approach, in which the agent (or learning device) learns to take the correct action by interacting with its surroundings and being rewarded or penalised for each action. RL is selected in this work due to its great applicability to the presented problem. For example, IoT devices need to interact with its environment in order to assess the circumstances and to take subsequent actions in terms of selecting the connection type and processing unit, which makes RL a solid candidate of solution given that it performs the optimisation via environmental interactions.

Being one of the most prominent RL algorithms, Q-learning aims to find the optimum policy for a given problem, that is, the best action to take at any given state. Thus, Q-learning is considered in this work as a methodology, since it offers two key features that enable an efficient solution to the problem in question. First, as it is a model-free learning approach [55, 228], it is 1) capable of operating in dynamically changing environments; 2) a low-complexity algorithm which does not require a lot of power, thus reducing the energy consumption of IoT networks. Second, Q-learning is known to converge in most cases [229], such as multi-agent non-cooperative environments [146] as in IoT networks.

It has already been proven in [54,230,231] that Q-learning certainly converges to the optimal policy under the following certain conditions:

1. visiting all the pairs of state and action infinitely often;
2. $\sum_{t}^{\infty} \alpha_t = \infty$, where $0 \leq \alpha < 1$, yielding that the learning rate should be sufficiently large;
3. $\sum_{t}^{\infty} \alpha_t^2 < \infty$, where $0 \leq \alpha < 1$, meaning that the learning rate should eventually be sufficiently small.

In this regard, an ε-greedy policy, where $\varepsilon > 0$, is employed in the developed Q-learning algorithm in order to satisfy the condition (a) of convergence, since it allows continuous exploring with a probability of ε. Moreover, the learning rate is chosen to be $\alpha = 0.5$ in this work, which could help in satisfying the condition (b) of the convergence. The condition (c), on the other hand, would require to decay the value of α over time. However, instead of α decaying, the aforementioned ε-greedy policy is benefited to consolidate the probability of convergence [54]. Furthermore, ε decaying—where the value of ε is slightly reduced over time—is employed during the implementations, since it can yield asymptotic convergence to the optimal policy [54]. Lastly, the convergence of the developed Q-learning algorithm is empirically observed and verified during the simulations.

In the following paragraphs, the proposed model for the employed Q-learning algorithm will be elaborated. Note that Algorithm 2 is used as a generic algorithm of Q-learning, while (2.5) is employed as the action-value function update policy.

Actions and States

In the considered scenario, the IoT devices are supposed to choose one of these options in Table 3.1 to conduct their connection and data processing tasks. In this regard, these options could also be treated as an action set for the developed Q-learning algorithm. However, Options B, C, and E include either fog or cloud processing, meaning that the IoT devices are supposed to offload their collected data to the fog or cloud for processing if they choose one of these options. Provided that partial offloading is also captured in this work, the amount of data to be offloaded should also be optimised, and thus considering the options in Table 3.1 as the action set would not be adequate for this objective. In other words, the action set should include not only the options for the connection-processing pairs, but also the amount of data to be offloaded.

Therefore, the action set is determined as follows:

$$\mathcal{A} = \mathbb{O} \times \Psi, \tag{3.11}$$

where × represents a Cartesian product, $\mathbb{O}$ is the set of all the possible options included in Table 3.1 (e.g., A, B, C, D, E), and Ψ is the set of all the possible offloading percentage options, such that

$$\Psi = \{\psi m \mid m \in \{0, 1, 2, ..., 20\}\}, \quad \psi \in \mathbb{R}^+, \tag{3.12}$$

where m is discretisation factor that is used to discretise the continuous values from 0% to 100%, and ψ is the resolution of the discretisation process. Note that Options A and D do not have Ψ parameter, since they do not perform any offloading at all. Using this phenomena, the actions set in (3.11) can be rewritten as

$$\mathcal{A} = \begin{cases} \mathbb{O} \times \Psi, & \text{if } \mathcal{O} \in \{\mathrm{B}, \mathrm{C}, \mathrm{E}\} \\ \mathbb{O}, & \mathcal{O} \in \{\mathrm{A}, \mathrm{D}\}. \end{cases} \tag{3.13}$$

Since each of these actions also defines the state of the agent, the state space—denoted by $\mathbb{S}$—is designed to be the same with the action space, such that $\mathbb{S} = \mathcal{A}$. After having both (3.11) and the fact that $\mathbb{S} = \mathcal{A}$, it is now worth discussing the selection of ψ value: i) considering (3.11), the size of $\mathcal{A}$ grows linearly with the size of Ψ; and ii) from $\mathbb{S} = \mathcal{A}$, the size of $\mathbb{S}$ grows equally with the size of $\mathcal{A}$. Thus, it can be deduced that the size of the action-value table grows exponentially with ψ, which would increase the requirements of memory, time, and computational resources [54], thereby the selection of ψ plays an important role. Hence, ψ can take any value in $\mathbb{R}^+$, but with a trade-off: the smaller it gets, the higher the resolution is, resulting in a more precise decision. However, smaller ψ creates an additional computational burden, since it increases $\dot{\mathcal{A}}$, where $\dot{\mathcal{A}} = |\mathcal{A}|$. Without loss of generality, $\psi = 5$ is taken in this work, as it provides sufficient resolution without significantly increasing the consumed computational resources.

Penalty function

In the proposed Q-learning algorithm, two novel prioritisation concepts are adopted:

- **prioritisation of requirements:** IoT devices are allowed to prioritise their requirements using a weighting mechanism, such that $w = \{w_{\mathrm{r}}, w_{\mathrm{s}}\}$, where $w_{\mathrm{r}} \in \mathbb{R}$ and $w_{\mathrm{s}} \in \mathbb{R}$ are the weight parameters for response time and security requirements, respectively. IoT devices are asked to rate the strictness of their requirements, such that lower values indicate that the requirement is loose, while higher values yield a stricter requirement.

- **prioritisation of energy consumption and monetary cost:** the total energy consumption and monetary cost are subject to prioritisation as well. However, unlike the requirement prioritisation case, where IoT devices control their weights (w_{r} and w_{s}), the energy consumption and cost prioritisations are triggered by the network. Moreover, this mechanism is linked to the battery level of an IoT device, β. More specifically, a certain threshold, denoted by $\beta_T \in \mathbb{R}$, is determined for the battery level, and

  – if the battery level of the IoT device i is above or equal to the threshold, such that
  $$\beta_i \geq \beta_T, \tag{3.14}$$
  the monetary cost is prioritised.

  – if, on the other hand, the battery level of the IoT device i is less than the threshold, such that
  $$\beta_i < \beta_T, \tag{3.15}$$
  then the energy consumption is prioritised.

Based on that, the overall penalty function for the developed Q-learning algorithm is formulated as follows:

$$\mathfrak{C}_Q = \Theta_{\mathrm{r}} + \Theta_{\mathrm{s}} + \Theta_{\mathrm{c}} + \Theta_{\mathrm{m}} + w_{\mathrm{e}} \mathsf{E}_{\mathbb{T}}, \tag{3.16}$$

where Θ_{r}, Θ_{s}, Θ_{c}, and Θ_{m} are the penalty elements for response time, security, capacity, and monetary cost, respectively, such that

$$\begin{aligned}
\Theta_{\mathrm{r}} &= \begin{cases} \Omega^{w_{\mathrm{r}}} + \mathcal{O}_{\mathrm{r}}, & \text{if } \mathcal{O}_{\mathrm{r}} > R_{\mathrm{r}} \\ 0, & \text{otherwise}, \end{cases} && \text{(3.17a)} \\
\Theta_{\mathrm{s}} &= \begin{cases} \Omega^{w_{\mathrm{s}}}, & \text{if } \mathcal{O}_{\mathrm{s}} < \Sigma_{\mathrm{s}} \\ 0, & \text{otherwise}, \end{cases} && \text{(3.17b)} \\
\Theta_{\mathrm{c}} &= \begin{cases} \Omega^{w_{\mathrm{c}}}, & \text{if } \mathcal{D}_{\mathrm{r}} > \mathcal{O}_{\chi} \\ 0, & \text{otherwise}, \end{cases} && \text{(3.17c)} \\
\Theta_{\mathrm{m}} &= w_{\mathrm{m}} M_{\mathbb{T},\mathcal{O}} \Psi \mathcal{D}_{\mathrm{r}}, && \text{(3.17d)}
\end{aligned}$$

where $\Omega \in \mathbb{R}$ is the global penalty factor, and $w_{\mathrm{c}} \in \mathbb{R}$ is the penalty factor incurred when the computational capacity is exceeded. Note that $w_{\mathrm{c}} > \max\{w_{\mathrm{r}}, w_{\mathrm{s}}\}$, since the computational capacity is a physical constraint that cannot be breached.

w_{e} and w_{m} are the weights for energy consumption and monetary cost set by the network, such that

$$w_{\mathrm{e}} = \begin{cases} w_{\mathrm{e}}^{+}, & \text{if } \beta_i < \beta_T \\ w_{\mathrm{e}}^{-}, & \text{otherwise,} \end{cases} \tag{3.18a}$$

$$w_{\mathrm{m}} = \begin{cases} w_{\mathrm{m}}^{+}, & \text{if } \beta_i \geq \beta_T \\ w_{\mathrm{m}}^{-}, & \text{otherwise,} \end{cases} \tag{3.18b}$$

where x^{+} and x^{-} represents the high and low values of x, respectively.

3.1.6 Performance Evaluation

In this section, the proposed RL approach is implemented in a simulation environment, as illustrated in Fig. 3.3.

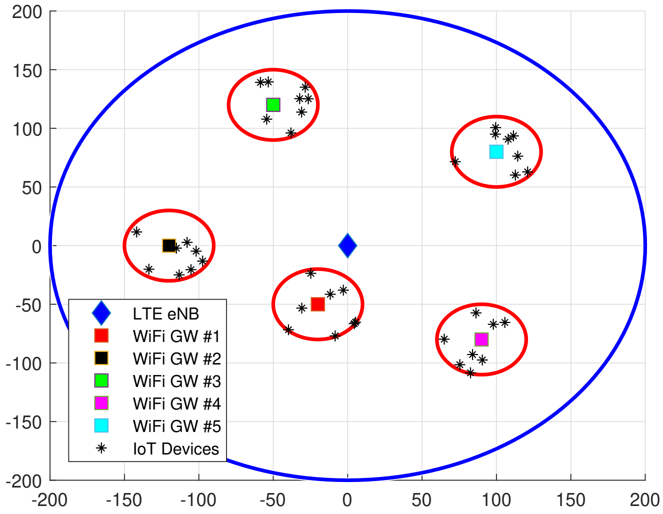


Figure 3.3: A sample snapshot of the simulation environment. While the eNB has a fixed location throughout all the repeating simulations, the gateways and IoT devices are located randomly, and thus their locations may vary from this sample snapshot. Note that only one of the five gateways is considered as the main gateway, and the simulations are performed for the IoT devices within the coverage area of this main gateway; the others are created as interference sources.

Benchmarking

It is considered that half of the IoT devices connect with NB-IoT in view of the data privacy and related security requirements; these represent Group-X. The remaining devices connect to the eNB through the Wi-Fi gateway, hence over two wireless hops, and represent Group-Y. Consequently, there are six possible fixed benchmark scenarios that may be formed by selecting the processing location of each group of devices; these are listed in Table 3.2.

Table 3.2: List of fixed benchmark scenarios with connection types and data processing unit

Scenario	**Group-X**	**Group-Y**
Sc_{A}	Device	Device
Sc_{B}	Cloud	Device
Sc_{C}	Device	Fog
Sc_{D}	Cloud	Fog
Sc_{E}	Device	Cloud
Sc_{F}	Cloud	Cloud

Performance Metrics

The obtained results are evaluated in five different metrics, namely: energy consumption, monetary cost, response time, security dissatisfaction, and a novel joint metric that is developed for this work. Moreover, these metrics are presented in a comparative fashion, where the performances of the benchmark methods—provided in Table 3.2—are compared to the proposed RL based method.

The performance metrics can be elaborated as follows:

- **energy consumption:** the accumulated energy consumption of all the IoT devices, which is caused by data processing and transmission, is calculated by

$$\ddot{\mathsf{E}}_{\mathbb{T}} = \sum_{i=1}^{N_{\mathrm{iot}}} \mathsf{E}_{\mathbb{T},i}, \tag{3.19}$$

  where N_{iot} is the number of IoT devices, and $\mathsf{E}_{\mathbb{T},i}$ is the total energy consumption of i^{th} IoT device.

- **monetary cost:** the aggregated monetary cost that the IoT devices are charged for data processing:

$$\ddot{M}_{\mathbb{T}} = \sum_{i=1}^{N_{\mathrm{iot}}} M_{\mathbb{T},i}, \tag{3.20}$$

  where $M_{\mathbb{T},i}$ is the total monetary cost for i^{th} IoT device.

- **response time:** the accumulated response time that occurs for all the IoT devices:

$$\ddot{R}_{\mathbb{T}} = \sum_{i=1}^{N_{\mathrm{iot}}} R_{\mathbb{T},i}, \tag{3.21}$$

where $R_{\mathbb{T},i}$ is the total response time occurred with i^{th} IoT device, and given by

$$R_{\mathbb{T},i} = \begin{cases} \mathcal{O}_{\mathrm{r}} - \Sigma_{\mathrm{r}}, & \text{if } \mathcal{O}_{\mathrm{r}} > \Sigma_{\mathrm{r}} \\ 0, & \text{otherwise.} \end{cases} \tag{3.22}$$

- **security dissatisfaction:** the number of IoT devices, whose security requirements are not satisfied:

$$\ddot{N}_{\mathrm{dis},\mathbb{T}} = \sum_{i=1}^{N_{\mathrm{iot}}} v_{\mathrm{dis},i}, \tag{3.23}$$

where $v_{\mathrm{dis},i}$ security dissatisfaction variable for the IoT device i, such that

$$v_{\mathrm{dis}} = \begin{cases} 0, & \text{if } \mathcal{O}_{\mathrm{s}} \geq \Sigma_{\mathrm{s}} \\ 1, & \text{otherwise.} \end{cases} \tag{3.24}$$

- **joint metric:** the combination of all the aforementioned metrics, such that

$$J = \hat{\ddot{\mathsf{E}}}_{\mathbb{T}}\xi_{\mathsf{E}} + \hat{\ddot{M}}_{\mathbb{T}}\xi_M + \hat{\ddot{R}}_{\mathbb{T}}\xi_R + \hat{\ddot{N}}_{\mathrm{dis},\mathbb{T}}\xi_N, \tag{3.25}$$

where $\hat{\ddot{\mathsf{E}}}_{\mathbb{T}}$, $\hat{\ddot{M}}_{\mathbb{T}}$, $\hat{\ddot{R}}_{\mathbb{T}}$, and $\hat{\ddot{N}}_{\mathrm{dis},\mathbb{T}}$ are the normalised versions of $\ddot{\mathsf{E}}_{\mathbb{T}}$, $\ddot{M}_{\mathbb{T}}$, $\ddot{R}_{\mathbb{T}}$, and $\ddot{N}_{\mathrm{dis},\mathbb{T}}$, respectively. The normalisation operation is performed here in order to keep the scale of each metric in the same range, thus preventing one from dominating another. Equation (3.5) is utilised for the normalisation (feature scaling) operation; i.e., obtaining $\hat{\ddot{\mathsf{E}}}_{\mathbb{T}}$, $\hat{\ddot{M}}_{\mathbb{T}}$, $\hat{\ddot{R}}_{\mathbb{T}}$, and $\hat{\ddot{N}}_{\mathrm{dis},\mathbb{T}}$ from $\ddot{\mathsf{E}}_{\mathbb{T}}$, $\ddot{M}_{\mathbb{T}}$, $\ddot{R}_{\mathbb{T}}$, and $\ddot{N}_{\mathrm{dis},\mathbb{T}}$, respectively. ξ_{E}, ξ_M, ξ_R, and ξ_N, where $\xi_{\mathsf{E}} = 1$, $\xi_M = 1$ Joules/AC, $\xi_R = 1$ Joules/s, $\xi_N = 1$ Joules, are coefficients used to make the units of the elements of J in (3.25) the same. Note that AC in the unit of ξ_M stands for arbitrary currency.

Battery Regimes

Two different battery regimes, namely low and high, are considered in this work, and results are produced separately in order to observe the behaviours of the proposed method[12].

[12] It is worth noting here that the benchmark methods do not consider the remaining battery level of IoT devices, and thus their behaviours are not expected to change with the battery level.

- **low-battery regime:** when the energy level of the battery of a particular IoT device is under a certain threshold ($\beta_i < \beta_T$), the device is treated as in a low-battery regime, and the proposed Q-learning algorithm starts to prioritise the total energy consumption of the device along with meeting the response time and security requirements. It is worth noting that, in this regime, the priority between the requirements and the energy consumption is determined by the weights of the requirements, w; such that if the requirements are strict (i.e., with high weights), they become more important than the energy consumption, and vice versa. The underlying idea here is that if the device is strict in any requirement, then it means that it is unwilling to compromise on that. For instance, a use-case might value the security very much, thereby replacing the battery of the correspondent IoT device would be more preferable than compromising on security to last the battery for longer. Moreover, the monetary cost is completely discarded, therefore, IoT devices are expected to be charged more when they are in this regime.

- **high-battery regime:** to be in the high-battery regime, the remaining energy in the battery of an IoT device should be above the aforementioned threshold for the battery level ($\beta_i \geq \beta_T$). In this high-battery regime, the monetary cost is valued significantly due to the absence of the battery level pressure[13]. Furthermore, similar to the low-battery regime, the importance of the requirements are determined by their correspondent weights, and the energy consumption is loosely prioritised[14].

Results and Discussions

This section begins with the obtained results when the IoT devices are in the low-battery regime. Table 3.3 shows the values of the parameters used in all the simulation campaigns in this work.

Fig. 3.4 demonstrates the performances of all the methods including the proposed one and the benchmarks when the entire set of IoT devices are in the low-battery regime. Moreover, both response time and security requirements of the IoT devices are prioritised in a rigid way, where both w_{r} and w_{s} are ranked as 3. As seen from the results, the proposed method performed better than all

[13] The pressure of keeping the device alive for longer by limiting its energy consumption.

[14] It is important to mention that unlike the low-energy regime, where the monetary cost is completely discarded, the energy consumption is still considered in the high-battery regime albeit with much less importance, since energy consumption of an IoT device should always be in the equation.

Table 3.3: Simulation parameters

Parameter	**Value**	**Description**
		Communication
$\varkappa$	3	Path loss exponent
d_0	10 m	Reference distance
σ	8 dB	Standard deviation for shadowing component
c	3×10^8 m/s	Speed of light
$f_{\mathrm{c,a}}$	2.4 GHz	Carrier frequency for IEEE 802.11g
$f_{\mathrm{c,b}}$	1700 MHz	Carrier frequency for NB-IoT
$f_{\mathrm{c,c}}$	1800 MHz	Carrier frequency for LTE
$\mathcal{N}_0$	-204 dbW/Hz	Noise density
T	$1/N_{\mathrm{iot}}$ s	Time period
W	180 kHz	Bandwidth
F	2	IEEE 802.11g retransmission rate
$r_{\odot}^{\mathrm{eNB}}$	1 km	Coverage radius of eNB
$r_{\odot}^{\mathrm{Wi\text{-}Fi}}$	30 m	Coverage radius of eNB
		General
N_{iot}	10	Number of IoT devices
ϵ	5×10^{-6} J	Energy consumption per computation cycle
S	8	Number of bits per data element
χ_{d}	30 kbps	Computational capacity of device
χ_{f}	100 kbps	Computational capacity of fog
χ_{c}	10 Mbps	Computational capacity of cloud
Γ_{d}	100	(Device) Computation cycles per data element
Γ_{f}	10	(Fog) Computation cycles per data element
Γ_{c}	1	(Cloud) Computation cycles per data element
M_{d}	AC 10^{-4}	(Device) Cost of processing per bps
M_{f}	AC10^{-1}	(Fog) Cost of processing per bps
M_{c}	AC1	(Cloud) Cost of processing per bps
$\mho$	200	Data compression rate
β_T	30%	Threshold for battery level
		Q-learning
α	0.5	Learning rate
φ	0.9	Discount factor
ε	0.8	Chance of choosing random action
N_{ep}	10^3	Number of episodes
N_{it}	10^3	Number of iterations per episode
Ω	10	Global penalty factor
w_{c}	5	Penalty of exceeding computation capacity
w_{e}^{+}, w_{e}^{-}	10, 1	High and low values of energy weight
w_{m}^{+}, w_{m}^{-}	10, 0	High and low values of monetary cost weight

the benchmark methods in the majority of the metrics[15]:

- it outperformed the benchmark methods in energy consumption, security dissatisfaction, and joint metric with varying scales, such that:
  - in energy consumption, the proposed method reduced the total energy consumption to 0 from 0.0081, 0.0069, 0.4547, 0.4535, 1.00, and 0.9988 for Sc_{A}, Sc_{B}, Sc_{C}, Sc_{D}, Sc_{E}, and Sc_{F}, respectively.
  - in security dissatisfaction, all the benchmark methods performed the same, and the proposed method managed to drop the dissatisfaction index from 1.00 to 0.
  - in the joint metric, the prosed method produced 0 cost, while Sc_{A}, Sc_{B}, Sc_{C}, Sc_{D}, Sc_{E}, and Sc_{F} produced 0.1561, 0.3934, 0.3706, 0.6077, 0.7627, and 1.00, respectively.
- it performed almost equally well with Sc_{A} in response time, while outperforming other benchmark scenarios: both the proposed method and Sc_{A} resulted in 0 response time while Sc_{B}, Sc_{C}, Sc_{D}, Sc_{E}, and Sc_{F} resulted in 0.0770, 0.4566, 0.5335, 0.9232, and 1.00, respectively. Note that since $(\mathcal{O}_{\mathrm{r}} - \Sigma_{\mathrm{r}})$ is the total excess amount of response time (please refer to (3.22)), the methods are entitled to result in no response time when they satisfy the response time requirements.
- in monetary cost, the proposed method managed to outperform only Sc_{F}, while being outperformed by the rest of the benchmark methods. In particular, the proposed method performed 0.6302, while Sc_{A}, Sc_{B}, Sc_{C}, Sc_{D}, Sc_{E}, and Sc_{F} resulted in 0, 0.4855, 0.0518, 0.5369, 0.5145 and 1.00, respectively.

In the following paragraphs, there will be individual discussions on the results for each performance metric:

- **energy consumption:** owing to the low-battery regime, it was expected for the proposed method to perform well in minimising the energy consumption. This is due to the fact that, when the battery level is under the threshold, the energy consumption component in the penalty function in (3.16) is prioritised through its weight (w_{e}) by setting it to its high value (w_{e}^{+}), as seen in (3.18a). In other words, given that the total energy consumption is strictly prioritised along with the the security and latency requirements,—as mentioned earlier while discussing the battery regimes—the algorithm

[15] The result values that will be mentioned in this section are normalised into the range of $[0, 1]$, and therefore the methods can output zero results.

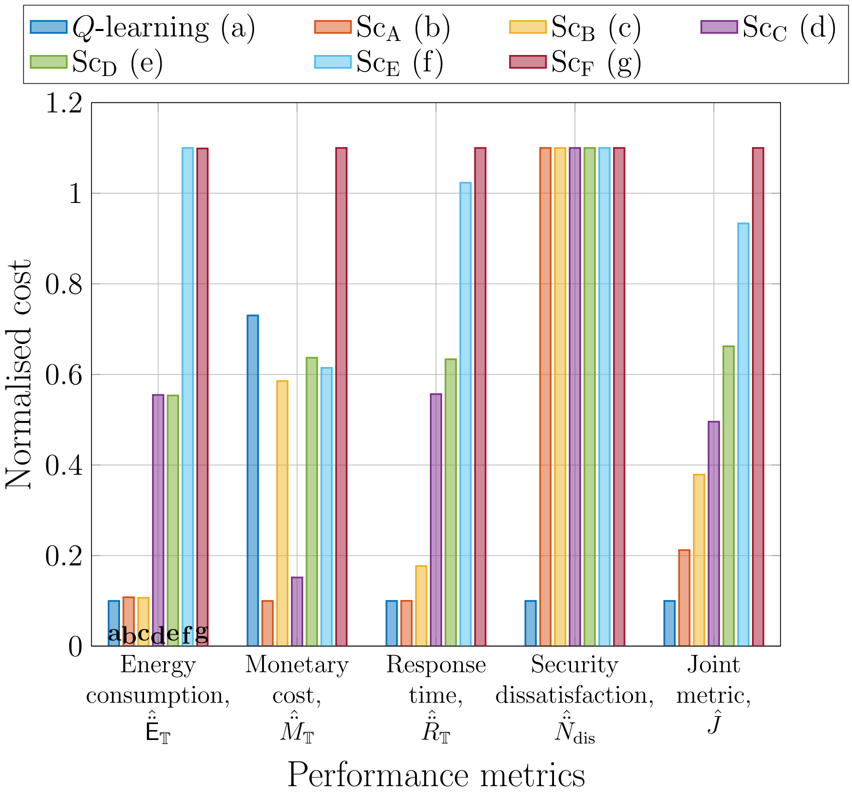


Figure 3.4: Performances of the proposed method (named as Q-learning in the legend) and the benchmark scenarios in terms of considered metrics when all the IoT devices are in the low-battery regime. The response time and security requirements are strictly prioritised, such that $w_\mathrm{r} = w_\mathrm{s} = 3$. Note that the results shown are normalised values in the range of $[0, 1]$ with an offset of 0.1, which is used only due to visualisation purposes. However, the results are discussed in the text without considering the offset value. The indexing from (a) to (g) in the legend and the first set of bars are done for identification, and the same order follows for all sets of the bars.

values the total energy consumption after ensuring the satisfaction of the requirements. This is a reasonable behaviour, since the energy consumption becomes more crucial when the battery is about to be depleted, which in turn interrupts the communication until the battery is recharged or replaced with a new one. Therefore, this results confirm the proper design of the penalty function of the developed Q-learning algorithm.

The energy consumption results for the benchmark scenarios are also worth discussing. From Section 3.1.3, it is already known that there are two components of the overall energy consumption, namely data processing and transmission. Based on the channel conditions and the distance between a transmitter and a receiver, the transmission energy consumption can prevail the processing energy consumption, or vice versa [211]. However, receiver sensitivity, which is captured by the link margin in this work, also plays an important role provided that it is directly correlated to the required received power, which in turn affects the required transmit power. Therefore, the connection type (i.e., NB-IoT, Wi-Fi, and LTE) is also involved in the breakdown of the total energy consumption. Due to the random distribution of the IoT devices and the Wi-Fi gateways at each repeat[16], it is avoided the two components of energy consumption dominating each other[17]. Nonetheless, owing to the link margin assumptions, in this work, the NB-IoT connection happened to be the least energy consuming connection type in most of the cases. Besides, in terms of the energy consumption, the descending order of the tasks is as follows: Wi-Fi connection, data processing, NB-IoT connection. Note that the energy consumption difference between the Wi-Fi connection and data processing happened to be much more than the difference between the data processing and NB-IoT connection. This may seem counter-intuitive given that the Wi-Fi gateway is much closer to the IoT devices than the eNB. Nonetheless, since the receiver sensitivity of NB-IoT is less than Wi-Fi and LTE, it happens to result in less energy consumption owing to the less required transmit power, which is caused by the less path-loss.

In this regard, since all the options include the same number of IoT devices with NB-IoT (Group-X) and Wi-Fi (Group-Y), there is no difference in terms of the number of connection types. However, the point that matters here is the processing unit. On one hand, when an IoT devices is connected through Wi-Fi, device processing is expected to consume less energy than cloud processing, with fog processing in between. On the other hand, when the device is connected through NB-IoT, device processing consumes more energy than, cloud processing, with fog processing in between. In addition, the maximum energy consumption with Wi-Fi connection is expected to be more than the maximum energy consumption with NB-IoT connection. Based on that, Sc_{B} resulted in the least energy consumption due to the fact that the devices with Wi-Fi connection process the data locally[18], while the devices using NB-IoT connection performs cloud processing[19]. Sc_{B} is followed by Sc_{A}, since it also processes the data locally for Wi-Fi connec-

[16] The simulations are repeated for 25 times to avoid the effect of randomness.

[17] This is ensured via random process; the path-loss can sometimes be huge, which results in the transmission energy consumption surpassing the data processing energy consumption, or vice versa. However, this effect is minimised by averaging out the simulation repeats.

[18] On average, the device processing is the minimum energy consuming option with Wi-Fi connection

[19] On average, cloud processing is the least energy consuming option for NB-IoT connection.

tions, and he small difference between Sc_{A} and Sc_{B} comes from the cloud and device processing for the NB-IoT connection. In a similar fashion, Sc_{E} consumed the maximum energy among the benchmark methods, since it employs cloud processing for Wi-Fi connected devices and device processing for NB-IoT connected devices. In summary, cloud processing is less energy consuming for NB-IoT, whereas it is more energy consuming for Wi-Fi. Moreover, as mentioned before, the maximum energy consumption with Wi-Fi is much more than that of NB-IoT.

- **monetary cost:** as a result of the low-battery regime, the proposed algorithm inclines towards being much looser in terms of the monetary cost, and thus it is not expected to be competitive in this metric. Considering (3.16), (3.17d), and (3.18b) together, the monetary cost behaviour of the algorithm is not surprising due to the following reasons:

  - from (3.18b), w_{m} is equalised to w_{m}^{-}, which is set to 0, as seen in Table 3.3;
  - this makes $\Theta_{\mathrm{m}} = 0$ in (3.17d); and
  - the monetary cost element in (3.16) is subsequently eliminated; thereby

  the algorithm does not have any monetary cost consideration at all. Based on that, as anticipated, the proposed method performed worse than all the benchmark methods other than Sc_{F}, which purely includes cloud processing.

  Similarly, the results of the benchmark methods are obtained as expected; the cloud is the most expensive means of data processing, followed by the fog, and the device (local), respectively, such that $M_{\mathrm{d}} < M_{\mathrm{f}} < M_{\mathrm{c}}$. Thus, the scenarios with device processing (e.g., Sc_{A}) resulted in less amount of monetary cost, whereas the scenarios with cloud and/or fog processing (e.g., Sc_{D} and Sc_{F}) become the most expensive ones. The results obtained in Fig. 3.4 confirms this statement.

- **response time:** the proposed method performed quite well in response time by outperforming all the benchmark methods. Provided that $w_{\mathrm{r}} = 3$, which yields a strict prioritisation of response time, it was expected for the proposed method to reduce the response time. Considering (3.16) and (3.17a) together, the effect of response time in the penalty function in (3.16) increases with growing w_{r}. As such, the primary objective of the developed Q-learning algorithm is to minimise the overall penalty, thereby the response time satisfaction becomes a key for this objective.

Similar to the previous metrics, the benchmark methods[20] also performed as anticipated. As discussed in Section 3.1.3, response time is a function of the number of hops (N_h), the computational power (Γ), retransmission rate (F), the data volume ($\mathcal{D} \in \{\mathcal{D}_\mathrm{r}, \mathcal{D}_\mathrm{p}\}$), and compression rate ($\mho$). Given that NB-IoT connection has only one hop and lower retransmission rate than Wi-Fi connection, the scenarios with NB-IoT resulted in a comparatively less response time. Similarly, from (3.4), albeit suffering from a higher computational time, device processing is also preferable due to the processed data transmission, which entails $\mho$ times less data volume[21]. Thus, for example, a cloud processing with a Wi-Fi connection would result in the highest response time owing to: 1) Wi-Fi connection, which has higher retransmission rate; 2) two hops taken; and 3) raw data transmission. In this regard, Sc_A resulted in the least response time, whereas Sc_F caused the highest response time among all the methods.

- **security dissatisfaction:** similar to the response time case, security requirement is also strictly prioritised in these simulation campaigns by setting w_s to 3. Considering (3.16) with (3.17b), higher values of w_s incurs more cost by increasing Θ_s, which in turn inflates the penalty function, $\mathfrak{C}_Q$ in (3.16). Given that the objective of the Q-learning algorithm is to minimise $\mathfrak{C}_Q$, satisfying the security requirement of IoT devices becomes crucial for the proposed algorithm, as Θ_s returns 0 when the requirement is met. To this end, the developed Q-learning algorithm achieved a significant reduction in terms of the security dissatisfaction when compared to the benchmark methods.

  One can question the equal results of the benchmark methods, but there is a rationale behind it: the half of the IoT devices are connected with NB-IoT (Group-X in Table 3.2), while the other half communicates through Wi-Fi (Group-Y in Table 3.2), and, as discussed in Section 3.1.4, the security requirement is captured by the need for an eSIM card, which is only available for NB-IoT connections. Thus, those connected with NB-IoT do not have any issue with the security dissatisfaction, since they always meet the requirements due to their eSIM card availability. Those connected through Wi-Fi, on the other hand, cannot respond to the eSIM card requirement. Based on that, the number of IoT devices with security dissatisfaction al-

[20] It could be worth noting here again that no partial offloading is allowed for the benchmark scenarios, such that they operate in an *on/off* fashion.

[21] The response time of the options (connection and processing) would alter for different values of $\mho$, F, and Γ. Therefore, the response time of the benchmark methods would change accordingly, but the discussions here are based on the current assumptions for $\mho$, F, and Γ.

ways equals to the number of IoT devices that: 1) is connected through Wi-Fi and 2) requires eSIM protection. Thus, the number of dissatisfied devices is the same for all the benchmark methods.

- **joint metric:** while each individual previous metric reflects the behaviours of the methods in a specialised manner, this joint metric summarises the overall performances. Therefore, this metric can be seen as a holistic cost of each method, which demonstrates how they performed when all the previous metrics are combined. The proposed method performed quite well and outperformed all the benchmark methods in different scales. The reasoning behind this is that there are four metrics in total other than joint metric, and the proposed method outperformed all the benchmark methods in three of them. Therefore, it is quite reasonable that the proposed method performed the best in terms of the joint metric. Similarly, the benchmark methods responded to the joint metric according to their results in each individual metric, namely energy consumption, monetary cost, response time, and security dissatisfaction.

The comparison between the cases of requirement de-prioritisation ($w_\mathrm{r} = w_\mathrm{s} = 0$), which will be referred to as low-battery requirement-aware (LBRA) hereafter, and prioritisation ($w_\mathrm{r} = w_\mathrm{s} = 3$), which will be referred to as low-battery requirement-unaware (LBRU) hereafter, under the low-battery regime is demonstrated in Fig. 3.5. The loss calculations for each metric is calculated as follows:

$$L = \frac{\Delta_{Q,\mathrm{LBRU}} - \Delta_{Q,\mathrm{LBRA}}}{\Delta_{Q,\mathrm{LBRU}}}, \tag{3.26}$$

where $\Delta_{Q,\mathrm{LBRA}}$ and $\Delta_{Q,\mathrm{LBRU}}$ are the differences between the values obtained via the proposed method and the minimum value obtained with the benchmark methods for LBRA and LBRU, respectively, such that

$$\Delta_{Q,\mathrm{LBRA}} = V_{Q,\mathrm{LBRA}} - \min(V_\mathrm{b}), \tag{3.27a}$$

$$\Delta_{Q,\mathrm{LBRU}} = V_{Q,\mathrm{LBRU}} - \min(V_\mathrm{b}), \tag{3.27b}$$

where $V_\mathrm{b} \in \{V_{Sc_\mathrm{A}}, V_{Sc_\mathrm{B}}, V_{Sc_\mathrm{C}}, V_{Sc_\mathrm{D}}, V_{Sc_\mathrm{E}}, V_{Sc_\mathrm{F}}\}$ is the obtained values in the aforementioned metrics with the benchmark methods. $V_{Q,\mathrm{LBRA}}$ and $V_{Q,\mathrm{LBRU}}$ are the obtained values in the aforementioned metrics with the proposed method for LBRA and LBRU, respectively.

There is an important caveat to be noted here: positive values of L calculated through (3.26) indicate loss, where the LBRU performed worse than LBRA, and the negative values of L indicate gain, where LBRU performed better than LBRA.

From the findings in Fig. 3.5, it is obvious that LBRA outperformed LBRU in all

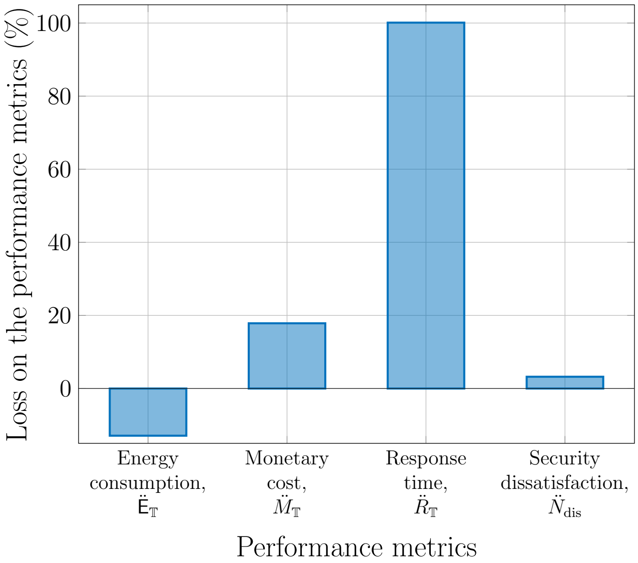


Figure 3.5: Performances of the proposed method in terms of considered metrics when all the IoT devices are in the low-battery regime. The results show the percentage loss when LBRU ($w_{\mathrm{r}} = w_{\mathrm{s}} = 0$) is compared to LBRA ($w_{\mathrm{r}} = w_{\mathrm{s}} = 3$). Positive values yield loss; i.e., the superiority of LBRA, while negative values yield gain; i.e., the superiority of LBRU.

the metrics[22] other than the energy consumption. Starting from response time and security dissatisfaction, the results are quite expected provided that both of the requirements are prioritised in LBRA with weight values of $w_{\mathrm{r}} = w_{\mathrm{s}} = 3$. Hence they are given a special care in LBRA when (3.16) is considered together with (3.17a) and (3.17b). Although the scale difference between the response time and security dissatisfaction is worth discussing, however it is better to first analyse the energy consumption results, which will be then more beneficial to explain such difference.

As seen from Fig. 3.5, the energy consumption is the only metric that LBRU performed better than LBRA. The rationale behind is that energy consumption is the only focus of LBRU given that: 1) the IoT devices are in the low-battery regime, and 2) both w_{r} and w_{s} are set to 0. In this regard, LBRU does not consider any other metric other than energy consumption, which is basically the reason why it managed to outperform LBRA.

Considering this rationale as a base, it would be more straightforward to explain the obtained results from the other metrics, as the reasoning for all the results are linked to each other and they all arise from this base. There are

[22] The joint metric was not considered in this set of results due to the fact that the objective of this comparison is to reflect the behavioural differences of LBRA and LBRU. Thus, the joint metric is out of the scope for this comparison, since it is a combination of the other metrics.

additional supporting facts as follows:

- LBRA aims at minimising the energy consumption, but with the response time and security constraints. In other words, the energy consumption is minimised after the response time and security requirements of IoT devices are satisfied. Provided that Options A, D, and E happen to result in minimal response time[23], meaning that they could be the options for the devices with low response time requirements.

- similarly, Options D and E are the ones that provide the eSIM protection, which means an IoT device should select one of these if it has security concerns.

Based on that, Options D and E followed by A are the intersection ones that are most likely to be selected when both response time and security are prioritised. For example, the response time requirement of an IoT device can only be satisfied with Option A and D, but Option E can result in less energy consumption. In such cases, LBRA would select Option A or D that has the least monetary cost due to device processing, whereas LBRU goes for Option E that result in the highest monetary cost due to cloud processing. As such, LBRA is more likely to perform better in terms of monetary cost, while LBRU is better in energy consumption. These explain the performance differences between LBRA and LBRU in terms of energy consumption and monetary cost.

It is now better to turn back to the discussion on the scale difference between the response time and security dissatisfaction. There are two points to consider:

- switching among Wi-Fi options (i.e., A, B, and C) and among NB-IoT options (i.e., D and E) do not change the security dissatisfaction results, but it changes the response time. In other words, selecting a different option from Table 3.1 definitely changes the response time behaviour, whereas the security behaviour might remain the same. In this regard, there is more room for response time to alter than that of the security behaviour.

- Options D and E have NB-IoT connection, which was already discussed in this section as being the least energy consuming one in majority of the cases, and thus the agent would be more prone to stick with them. Owing to the fact that Options D and E are with NB-IoT connection, both LBRA and LBRU would more possibly select one of these options, which would have an impact on the response time but the security behaviour remains unaffected.

[23] Obviously, these arguments would be subject to a change for the case of partial offloading, but they hold in general.

Due to these reasons, it is quite reasonable that LBRA outperforms LBRU more significantly in response time than that of security dissatisfaction.

Fig. 3.6 reveals the performance comparison between LBRA and the case when the IoT devices are in the high-battery regime and their requirements are fully prioritised ($w_\mathrm{r} = w_\mathrm{s} = 3$), which will be referred to as high-battery requirement-aware (HBRA) hereafter. Note that only energy consumption and monetary cost results are presented in Fig. 3.6, since LBRA and HBRA performed equally well in response time and security dissatisfaction given that they both fully prioritise the device requirements.

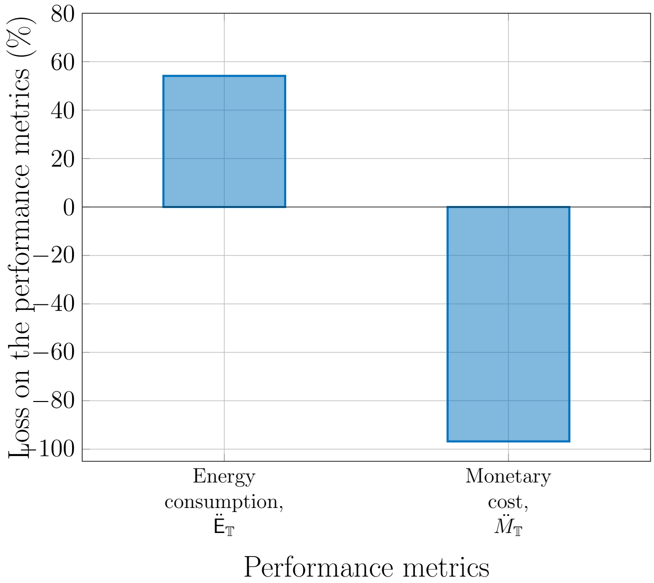


Figure 3.6: Performances of the proposed method in terms of energy consumption and monetary cost. The results show the percentage loss when HBRA is compared to LBRA. Positive values yield loss; i.e., the superiority of LBRA, while negative values yield gain; i.e., the superiority of HBRA.

The findings in Fig. 3.6 show that LBRA reduced the energy consumption of HBRA by around 53%, while HBRA managed to decrease the monetary cost of LBRA by around 97%. These results are quite expected because LBRA focuses only on minimisation of the energy consumption, while completely ignoring the monetary cost reduction[24]. HBRA, on the other hand, aims at minimising the monetary cost rather than the energy consumption, since the battery levels of IoT devices are high.

Although these explanations are adequate to understand why they surpass each other in the two considered metrics, there is still room for clarification for

[24] Energy consumption minimisation and/or monetary cost reduction are the secondary objectives for both LBRA and HBRA, since they fully prioritise the device requirements. In other words, they only start considering the energy consumption and monetary cost after ensuring the satisfaction of device requirements in terms of response time and security.

the question of why the scales of the outperformance are quite different from each other. Considering (3.16) together with (3.17d), (3.18b), and (3.18a), it is obvious that both energy consumption and monetary cost have their own impacts in $\mathfrak{C}_Q$. On one hand, when the IoT devices are in the low-battery regime, the weight for energy consumption (w_{e}) takes its higher value (w_{e}^{+}), which is set to 10, while the weight for monetary cost (w_{m}) takes its lower value (w_{m}^{-}), which is set to 0. On the other hand, when the IoT devices are in the high-battery regime, w_{e} is set to its lower value as $w_{\mathrm{e}}^{-} = 1$, while w_{m} is set to its higher value as $w_{\mathrm{e}}^{-} = 10$. This means that

- when the IoT devices are in the low-battery regime, the algorithm fully focuses on the energy consumption minimisation while completely discarding the monetary cost reduction; but

- when the IoT devices are in the high-battery regime, the algorithm mainly takes care of the monetary cost reduction, but without completely ignoring the energy consumption minimisation.

This is done because energy consumption is always important for an IoT device regardless of the battery level, which would only change the degree of importance. As such, HBRA still tries to conserve some energy while focusing primarily on the monetary cost reduction, and thus this puts a barrier for HBRA's loss in energy consumption. Nonetheless, LBRA does not take the monetary cost reduction into account at all, thereby the scale of HBRA's gain in monetary cost is more than HBRA's loss in energy consumption.

As a result of these observations, it is concluded that the penalty function of the proposed Q-learning algorithm, given in (3.16), is well designed, since the behaviours of the algorithm in each single metric for each scenario came as expected. In particular, the algorithm penalises the agent: i) exponentially for the requirements, and ii) linearly for the energy consumption and monetary cost. Moreover, the remaining level of the battery is considered as a determinant for the policy to be followed, such that energy consumption is valued in low-battery regime, while high-battery regime imposes the prioritisation of the monetary cost. The obtained results reveal that the algorithm follows these rules and works in line with the design objectives owing to the careful construction of the penalty function. Towards that end, the findings from the obtained results confirm that the design of the reward/penalty function for Q-learning is at the heart of the algorithm, and it is capable of determining the overall performance.

3.1.7 Summary

A novel approach is presented in this work for energy and context-aware IoT connectivity that jointly optimises the energy consumption, monetary cost, security, and response time. More specifically, the objective of this work is to determine the wireless connection type and data processing unit along with the amount of data to be offloaded, which is the case when data to be processed at a unit other than the device. In that regard, IoT devices come with diverse requirements in terms of response time and security, and they are allowed to prioritise their requirements. For example, an IoT device would declare that the security is strictly important for it, while response time has a loose priority. In addition to this prioritisation, the proposed scheme also takes the battery level of a device into account, such that the minimisation of energy consumption is focused if the battery level is under a certain threshold, while the reduction of monetary cost is mainly targeted in case the battery level is above that threshold. In particular, the proposed approach first observes the prioritisation weights along with the remaining battery level of the device. Then, based on these inputs, it determines its policy, such that: if the requirements are strictly prioritised, the first priority becomes satisfying them. After that, according the remaining battery level, the energy consumption or monetary cost is minimised. The proposed scheme employs Q-learning algorithm, and manages to achieve significant gains compared to deterministic routes. Although some benchmark scenarios may result in lower costs in some of the metrics, none was able to meet the holistic context-aware performance target.

3.2 Energy Optimisation through Traffic-Aware Cell Switching

BSs are the major energy consumers in cellular networks, accounting for about 60%-80% of the total energy consumption [161]. As such, reducing the energy consumption of BSs will amount to significant reduction in the total energy consumption of cellular networks. From an energy efficiency perspective, the ideal operation of a BS is to be in such a way that the energy consumption scales with traffic load, meaning that the power consumption should be negligible at no load and gradually increase with the increasing load. However, in practice that is not the case, as BSs still consume a significant amount of energy (about 50%-60% of its maximum energy consumption) when not serving any user. Since the traffic load of cellular networks exhibit temporal and spatial variations, the traditional technique of keeping the BS always *on*, even when it is not serving any user, results in energy wastage. Therefore, load adaptive network operation, where BSs are turned off or operated in low power modes during periods of low or no traffic in order to save energy, has been the focus of many studies [41, 232–235].

Nevertheless, it is not always feasible to completely switch off SCs in the conventional HetNet architecture because it often creates coverage holes, which in turn degrade the QoS of users initially covered by the inactive SCs. In addition, sleeping BSs do not transmit pilot signals needed by the UE for cell discovery, channel estimation, and subsequent connection, hence in the conventional HetNet architecture, certain components of the SCs need to be left always *on* even in a sleep mode, resulting in sub-optimal energy savings [236–238].

Handling these challenges of BS switching in conventional HetNets requires a paradigm shift towards CDSA [239], where the MCs—also known as control BS (CBS)—maintain constant coverage and provide signalling functionalities and low data rate services, while the SCs—also known as data BS (DBS)—provide high data rate services and are connected to the MCs through the backhaul. This architecture provides support and flexibility for dynamic cell switching operations as the MC always ensures constant coverage for both idle and active users. The MC is also responsible for switching BSs off/on as well as associating users to the SCs, thereby making the complete switch off of DBSs possible [236, 237, 240].

In this work, an RL algorithm with linear function approximation—known as SARSA with VFA—is proposed for cell switching and traffic offloading in ultra-dense RAN. The main advantage of the proposed framework is that all the states need not be visited as in [61] before the algorithm learns the optimal policy; instead, the developed algorithm exhibits quick convergence and is simpler to implement compared to deep RL approaches. The learning algorithm is implemented at each MC and it has the ability to learn the optimal switching pattern even when a large number of SCs are deployed under the coverage area of the MCs.

3.2.1 Related work

Cell switching with traffic offloading has been identified as one of the techniques for reducing the energy consumption of cellular networks. Several methods have already been proposed for scheduling cell switching in the literature using various methods from analytical modelling to heuristic algorithms [41,232–235]. However, it is very difficult to develop accurate analytical models for network optimisation when network dimensions become very large due to network complexity and high computational overhead [241].

The authors in [232] develop a load based dynamic SC switching scheme for UDNs in order to minimise the signalling overhead resulted from user traffic offloading during the cell switching process as well as to optimise the energy savings of the network using two heuristic algorithms. In [233], the problem of user association and cell sleeping in multi-tier ultra-dense SC networks is formulated as a complex integer programming and two low-complexity heuristic algorithms are employed to determine the optimal user association and the cell switching pattern. A greedy heuristic algorithm is proposed in [234] to determine the switching off/on pattern of SCs in a green UDN in order to optimise the network energy efficiency by considering traffic load of the SCs and service requirements of users. Heuristic algorithms often employ exhaustive search, which is often slow and computationally demanding, to find the optimal solution, hence they are only suitable for small network deployment. On the other hand, the ultra-dense 5G network scenario will involve massive deployment of SCs [242], which would make it practically impossible to adopt such heuristic algorithms, as it would result in huge computational overhead and degradation in QoS.

An alternative solution for finding optimal switching pattern for ultra-dense deployment scenarios is to employ RL techniques. The authors in [61,63,243–246] propose Q-learning based cell switching techniques for energy efficiency optimisation. In [243], the BS is assumed to comprise of modular resources, and a Q-learning algorithm is developed to dynamically activate and deactivate certain number of modules in the BS based on the traffic demand at each time instance. A distributed Q-learning algorithm is proposed in [244] to control the sleep depth of the BS in order to minimise energy consumption and network latency. The authors in [245] design a Q-learning algorithm to determine the duration of time that the BS can spend at a particular sleep level in order to optimise energy consumption of the network while considering BS activation latency and the service requirements of users as constraints. Another Q-learning framework is developed in [63] in order to determine the optimal switching and traffic offloading strategy in a two-tier HetNet with separation architecture. The authors in [246] propose a location-aware multi-level sleep mode strategy using Q-learning to determine the sleep mode level of the BS based on user locations in the network and their velocity towards the neighbouring BS. Nonetheless, the works in [63, 243–246] only consider small to medium network deployment scenarios, where the state-action space is suitable for the implementation of conventional RL algorithms. As such, conventional RL algorithms are very challenging to implement when the network dimensions become huge, because it often results in very large state-action space, which is computationally demanding to learn and requires considerable amount of memory to store the action-value table (Q-table). This makes such algorithms impracticable to implement in real networks.

In an attempt to solve the curse of dimensionality problem facing conventional RL algorithms, they are combined with VFA in order to estimate the optimal policy. In this regard, RL with linear function approximators and deep RL approaches are proposed in [61, 135, 136]. The authors in [61] develop a centralised and decentralised Q-learning algorithm with compact state representation (QC-learning) for traffic offloading and cell switching for HetNets to minimise energy consumption. The centralised QC-learning is implemented at the BS controller. The QC-learning is a compact representation of the state-action pair using linear VFA when it becomes practically impossible to explicitly store each state-action pair in a look-up table owing to the very large number of SCs. Moreover, as the number of SCs in the network becomes very large, the action set also grows dramatically even with the compact state representation of the Q-table, thereby making it difficult to implement a centralised cell switching and traffic offloading scheme. Therefore, a decentralised multi-agent QC-learning is presented, where the MCs learn in a cooperative manner and take joint traffic offloading and cell switching actions by exploiting the previous cell switching strategies used by other MCs. Furthermore, in developing decentralised multi-agent QC-algorithm, the authors [61] assume that all the MCs under the controller have similar or stationary network states, which is a requirement for implementing joint cell switching and traffic offloading strategy. This might not be the case in real networks, as networks' states may vary from one MC to another due to temporal and spatial variations in user traffic demands [247]. As such, it might not always be feasible to perform joint cell switching and traffic offloading actions. In addition, the problem of increased state-action pair also arises in multi-agent Q-learning, since each agent also includes its own state-action pair to the joint state action space [247]. This increases the computational complexity as well as the memory required for storing the joint state-action space at the controller.

Deep RL algorithms have the ability to accommodate large state-action space resulting from large-scale network deployment scenarios, however, training such deep neural network models can be computationally demanding, energy consuming, and should only be considered when there are no simpler solution approaches available or when the complexity of the network requires the application of a nonlinear function approximator to estimate the optimal policy.

3.2.2 Objectives and Contributions

In this work, an intelligent cell switching and traffic offloading framework is proposed using an RL technique known as SARSA with VFA [54] in order to reduce the energy consumption of ultra-dense RAN. In the CDSA RAN, a cell switching and traffic offloading mechanism is implemented in a locally centralised manner at each MC, which is responsible for scheduling the switching off/on pattern of all the SCs deployed under its coverage. The proposed algorithm provides a compact form of representing the action-value function like the QC-learning algorithm in [61], but all the states need not be visited as in [61] before the optimal strategy is obtained. The contributions of this work are as follows:

- a scalable traffic-aware RL algorithm is proposed based on VFA to find the optimal policy in terms of energy efficiency for controlling SCs' *on/off* status, which delivers a good level of QoS;
- it is mathematically proven that turning a SC off is not always profitable in terms of energy minimisation, instead there are only some situations, at which turning off action results in the optimal solution. A relationship that evidences those situations is also provided;
- the proposed algorithm is evaluated using a traffic model based on real world data, making the solution more reliable and realistic.

RL is employed due to its fast convergence and learning ability, assisting to make real-time, accurate, and efficient switch off/on decision at each time slot. Moreover, VFA is utilised in the developed RL algorithm, since the state space expands exponentially with increasing number of SCs in the network. Even though this could be handled by conventional RL algorithms to some extent, it becomes infeasible to manage once the network size becomes very large. In addition, due to the careful and proper design of the action set in the proposed algorithm, there is no need to include all the possible switching combinations, thus annihilating the need for cooperative learning. Lastly, the proposed algorithm is tested in a realistic scenario, where all types of SCs given in [156] with their diverse characteristics are included.

3.2.3 System Model

Network Model

As mentioned before, in this work an ultra-dense RAN with a CDSA architecture is considered [239]. The network model, as illustrated in Fig. 3.7, consists

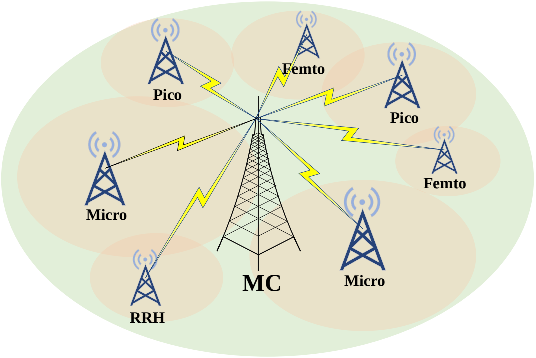


Figure 3.7: A HetNet with CDSA comprising of a MC that works as the CBS and a dense deployment of various types of SCs (micro, RRH, pico, femto) to provide DBS services.

of a dense network, where SCs—acting as DBSs—are deployed under the coverage area of a MC, which acts as the CBS. Moreover, SCs and MC operate on dedicated frequency channels and SCs are connected to the MC via optical fibre links.

The MC is responsible for constant coverage, control signalling, and data services, while SCs handle only data services and user specific requests. Furthermore, the MC coordinates the traffic offloading and switching off/on of all SCs under its coverage by observing their traffic loads and deciding which set of SCs should be turned off during periods of low traffic intensity, taking the available capacity of the MC into account.

Network Power Consumption

EARTH power consumption model [156] is followed in this work, as it is given in (2.15). Moreover, all SCs of the same type are considered to have identical hardware, such that their power amplifier efficiencies, and circuit power consumptions are the same. Also, power allocation is not considered, hence, each type of BS has a fixed transmit power that is constant among BSs of that type.

As such, $\mathcal{P}$, the instantaneous power consumption for the considered CDSA network, is expressed as

$$\mathcal{P} = \sum_{j=1}^{N_{\mathrm{sc}}+1} \mathcal{P}_j, \tag{3.28}$$

where N_{sc} is the number of SCs in the deployment.

3.2.4 Problem Formulation

Considering the architecture described above, the aim of this work is to find the best policy—in terms of energy optimisation—that offers a required QoS to the users. A policy π_{sc} is defined by which SCs should be *on* at a given time t. In other words

$$\pi_{\text{sc}} = \{\delta_1, \delta_2, \ldots, \delta_{N_{\text{sc}}+1}\}, \tag{3.29}$$

where $\delta_j \in \{0, 1\}$ indicates the state of $\mathcal{B}_j$; 1 for *on* and 0 for *off*. $\mathcal{B}_1$ represents the MC, and thus δ_1 is always 1, as it is always *on*.

Considering $j > 1$, when δ_j changes from 1 to 0 at time t, the MC allocates its users, such that

$$\Lambda_{1,t} = \Lambda_{1,t-1} + \phi_j \Lambda_{j,t-1}, \tag{3.30a}$$

$$\Lambda_{j,t} = 0, \tag{3.30b}$$

where $\Lambda_{j,t}$ corresponds to the load factor of $\mathcal{B}_j$ at time t, and ϕ_j is the relative capacity of $\mathcal{B}_j$ with respect to $\mathcal{B}_1$:

$$\phi_j = \frac{C_j}{C_1}, \quad j > 1, \tag{3.31}$$

where C_j indicates the maximum capacity of $\mathcal{B}_j$. Conversely, when δ_j switches from 0 to 1 at time t, the MC offloads some of its traffic to $\mathcal{B}_j$, such that

$$\Lambda_{j,t} = \frac{\tau_j}{C_j}, \tag{3.32a}$$

$$\Lambda_{1,t} = \Lambda_{1,t-1} - \phi_j \Lambda_{j,t}, \tag{3.32b}$$

where τ_j corresponds to the resources used by users served by $\mathcal{B}_j$. Note that $C_j \geq \tau_j$.

Therefore, the problem can be formally written as

$$\begin{aligned} \min_{\pi} \quad & \mathcal{P}(\pi_{\text{sc}}) \\ \text{s.t.} \quad & \Lambda_{1,t} \leq 1. \end{aligned} \tag{3.33}$$

Note that the only constraint in the problem is to ensure that the capacity of the MC is not exceeded, this takes care of the QoS requirement[25]. In other words,

[25] If the capacity of the MC is not exceeded, the QoS is ensured given that all the users are provided with their required bandwidth. It is also assumed in this work that, after the offloading, the users receive sufficient SINR that is needed to achieve the same level of throughput they had before offloading. This is a reasonable assumption, since the output power of the MC increases with its load (refer to (2.15)) that would subsequently increase the received power.

the MC only offloads users from a SC if it can maintain the QoS of any user associated to it. Moreover, this ensures that once a solution is implemented,

$$\mathfrak{T}_{\mathrm{r},1} \leq \mathfrak{T}_{\mathrm{m},1}, \tag{3.34}$$

where $\mathfrak{T}_{\mathrm{r},1}$ and $\mathfrak{T}_{\mathrm{m},1}$ are the required and maximum provided throughputs by the MC, respectively.

Theorem 1. *If draw a random SC $\mathcal{B}_j$ from $\mathbb{B}$, the set of all possible BSs, the probability of $\delta_j = 1$ integrating the optimal policy is 1 if $j > 1$ and*

$$\frac{\mathcal{P}_{o,j} - \mathcal{P}_{s,j}}{\phi_j \eta_1 \mathcal{P}_{T,1} - \eta_j \mathcal{P}_{T,j}} < \Lambda_j, \tag{3.35}$$

when $\phi_j \eta_1 \mathcal{P}_{T,1} - \eta_j \mathcal{P}_{T,j} > 0$.

Proof. Using (3.28), the difference in power consumption $\Delta\mathcal{P}$ when considering changing δ_j to 1 can be expressed as

$$\Delta\mathcal{P} = \mathcal{P}_t - \mathcal{P}_{t-1} = \sum_{i=1}^{N_{\mathrm{sc}}+1} \mathcal{P}_{i,t} - \sum_{i=1}^{N_{\mathrm{sc}}+1} \mathcal{P}_{i,t-1}, \tag{3.36}$$

where $\mathcal{P}_t$ is the total power consumption of the network at time t and $\mathcal{P}_{i,t}$ is the power consumption of $\mathcal{B}_i$ at time t.

Next, (3.36) can be expanded as

$$\Delta\mathcal{P} = \mathcal{P}_{1,t} + \mathcal{P}_{j,t} + \sum_{i=2, i\neq j}^{N_{\mathrm{sc}}+1} \mathcal{P}_{i,t} - \left(\mathcal{P}_{1,t-1} + \mathcal{P}_{j,t-1} + \sum_{i=2, i\neq j}^{N_{\mathrm{sc}}+1} \mathcal{P}_{i,t-1} \right). \tag{3.37}$$

Assuming $\mathcal{P}_{i\notin\{1,j\},t} = \mathcal{P}_{i\notin\{1,j\},t-1}$, meaning all other SCs are kept at their states, (3.37) then becomes

$$\Delta\mathcal{P} = \mathcal{P}_{1,t} + \mathcal{P}_{j,t} - \mathcal{P}_{1,t-1} - \mathcal{P}_{j,t-1}. \tag{3.38}$$

Next, using (2.15) and replacing (3.32a) and (3.32b) into (3.38) yields

$$\begin{aligned} \Delta\mathcal{P} = & \mathcal{P}_{\mathrm{o},1} + \eta_1(\Lambda_{1,t-1} - \phi_j \Lambda_{j,t})\mathcal{P}_{\mathrm{T},1} + \mathcal{P}_{\mathrm{o},j} + \eta_j \Lambda_{j,t} \mathcal{P}_{\mathrm{T},j} \\ & -\mathcal{P}_{\mathrm{o},1} - \eta_1 \Lambda_{1,t-1} \mathcal{P}_{\mathrm{T},1} - \mathcal{P}_{\mathrm{s},j}, \end{aligned} \tag{3.39}$$

which can be further simplified to

$$\Delta\mathcal{P} = \mathcal{P}_{\mathrm{o},j} + \eta_j \Lambda_{j,t} \mathcal{P}_{\mathrm{T},j} - \eta_1 \phi_j \Lambda_{j,t} \mathcal{P}_{\mathrm{T},1} - \mathcal{P}_{\mathrm{s},j}. \tag{3.40}$$

From (3.40), it is easy to see that $\Delta P < 0$ when

$$\mathcal{P}_{\mathrm{o},j} + \eta_j \Lambda_{j,t} \mathcal{P}_{\mathrm{T},j} < \phi_j \eta_1 \Lambda_{j,t} \mathcal{P}_{\mathrm{T},1} + \mathcal{P}_{\mathrm{s},j}. \tag{3.41}$$

Note that, in order to isolate $\Lambda_{j,t}$ in (3.41), both sides must be divided by $\phi_j \eta_1 \mathcal{P}_{\mathrm{T},1} - \eta_j \mathcal{P}_{\mathrm{T},j}$, and thus (3.35) is only valid for $\phi_j \eta_1 \mathcal{P}_{\mathrm{T},1} - \eta_j \mathcal{P}_{\mathrm{T},j} > 0$. Lastly, the index t is removed from (3.41) to make it general and solve for Λ_j, yielding (3.35). □

Based on Theorem 1, it can be seen that the optimal policy will tend to have more SCs turned on when they are more loaded, as the inequality in (3.35) will be more easily met. Similarly, it can also be seen that when the transmit power of SCs is smaller, it becomes more advantageous to use them, and the same can be said if the MC is not very efficient (η_1 is small). Moreover, as one would expect, when the power consumption of a sleeping BS is higher, the less likely it will be that turning it off would be energy efficient. Lastly, it can clearly be observed that when (3.35) occurs for any BS in the network, there will be a situation, where the most optimal policy has SCs turned on and consumes less energy than keeping only the MC operational.

3.2.5 Proposed Solution

SARSA with VFA

As Chapter 2 already discusses SARSA algorithm in detail, its only a brief summary is included in this section. The algorithm works by observing the penalty (cost) of taking actions and updating its estimate of $\vec{w}_Q$ at every iteration and then choosing the best action according to its estimate of the action-value matrix. Algorithm 5 [54] contains a pseudo code implementation of SARSA with VFA, where $\mathfrak{s}_t$ and $\mathfrak{s}_{t+1}$ are the current and next states, respectively; while a_t and a_{t+1} are the current and next actions, respectively. Note that lines 22 to 29 in Algorithm 5 are correspondent to the stopping criteria, where $N_{\mathrm{it,min}}$ is the minimum number of iterations to take before the stopping criteria come into the effect. $\mathfrak{C}_{Q,\mathrm{min}}$ and $\mathfrak{C}_{Q,\mathrm{max}}$ are the minimum and maximum penalties observed up to that iteration, respectively. Lastly, Ω is the threshold for the feature scaled cost, while $N_{\mathrm{it,rep}}$ defines the number of iterations to be repeated (the conditions on lines 23 and 24 are kept satisfied) before stopping.

Algorithm 5: Proposed SARSA with VFA

```
for *Every episode* do
    Initialise the current state, $\mathfrak{s}_t$;
    for *All actions in $\mathcal{A}$* do
        Get features, $\vec{\mathfrak{X}}$;
        Estimate value of $Q$ through (2.9);
    end
    Choose action, $a_t$, according to policy;
    Set $N_{\text{it}} = 0$;
    for *Each iteration* do
        Update $N_{\text{it}} \leftarrow N_{\text{it}} + 1$;
        Take the action $a_t$;
        Observe cost, $\mathfrak{C}_Q$, via (3.47);
        Move to next state;
        for *All actions in $\mathcal{A}$* do
            Get features, $\vec{\mathfrak{X}}$;
            Estimate value of $Q$ through (2.9);
        end
        Choose next action, $a_{t+1}$, according to policy;
        Update the weights, $\vec{w}_{Q,t}$, using (2.10);
        $S_t \leftarrow \mathfrak{s}_{t+1}$;
        $a_t \leftarrow a_{t+1}$;
        if $N_{it} > N_{it,min}$ then
            if $\frac{\mathfrak{C}_Q - \mathfrak{C}_{Q,min}}{\mathfrak{C}_{Q,max} - \mathfrak{C}_{Q,min}} \leq \Omega$ then
                if *$\mathfrak{C}_Q$ is the same for $N_{it,rep}$ iterations* then
                    Stop executing;
                    Jump to the next episode;
                end
            end
        end
    end
end
```

Traffic Aware Energy Efficient Cell Switching

Leveraging the framework described above, a linear VFA solution is proposed to solve (3.33). Since the approach is looking to accomplish a globally optimal solution, the proposed framework is centralised and computed at the MC. As the total number of policies increases exponentially with N_{sc}, it would not be scalable to consider any policy as an action. Therefore, a reduced action space is proposed as follows:

Actions: The actions for the proposed VFA consist of switching off/on different SCs in the network. However, because there are so many possibilities, an alternative representation is proposed, which allows the algorithm to sample several different possibilities by taking different actions.

This representation is done as follows. First, the status of the SCs in the network are converted to a binary number, such that the SCs that are *on* are treated as binary 1, while the SCs that are *off* are considered binary 0. In this regard, the status of the network at time t is

$$h(t) = \{\delta_i(t) | i \in \{1, 2, ..., N_{\text{sc}}\}\}, \tag{3.42}$$

where $\delta_i \in \{0, 1\}$ is the state of the i^{th} SC in the network.

Next, h is represented by a binary number h_{b} with N_{sc} digits, such that the status of each SC represents one of its digits, and thus

$$h_{\text{b}} = h_1 h_2 ... h_s. \tag{3.43}$$

Within the proposed representation, the set of possible actions is defined as

$$\mathcal{A} = \{0, \pm A_{\text{is}}^0, \pm A_{\text{is}}^1, ..., \pm A_{\text{is}}^s\}, \tag{3.44}$$

where A_{is} is a scalar, defining the inter-space between two consecutive actions.

In this case, taking an action at time t, means to perform

$$h_{\text{d}}(t+1) = h_{\text{d}}(t) + A_{\text{z}}, \tag{3.45}$$

where A_{z} is an entry[26] from $\mathcal{A}$ and h_{d} is the decimal representation of h_{b}. In other words, an action consists of turning off/on a number of BSs, depending on the current status and on A_{z}. This ensures that, instead of checking all possible

[26] Note that not all entries of $\mathcal{A}$ can be selected at every time instant. Instead, the only valid actions are those which render $0 \leq h_{\text{d}} \leq 2^{N_{\text{sc}}} - 1$.

statuses of SCs, only some samples of the entire set are taken.

To illustrate the proposed idea, consider the following example. In a network with 4 SCs at time t, let $h(t) = \{1, 1, 0, 1\}$. In this case, $h_{\mathrm{d}}(t) = 13$. Now, let us assume that $A_{\mathrm{is}} = 2$ and that the action selected is $+A^1_{\mathrm{is}}$. In this case, the next status will be $h_{\mathrm{d}}(t+1) = h_{\mathrm{d}}(t) + A^1_{\mathrm{is}} = 15$, or in other words $h(t+1) = \{1, 1, 1, 1\}$, which implies that all BSs are turned on at time $t + 1$.

Features: The features used by the MC in order to find $\hat{Q}$ are the total power consumption of the network and the total load factor of each BS, such that

$$\mathfrak{X} = [\mathcal{P}, \Lambda_1, \Lambda_2, \ldots, \Lambda_{N_{\mathrm{sc}}+1}]. \tag{3.46}$$

Penalty Function: The proposed penalty function is described as the total power consumption plus a penalty component—which is proportional to the number of SCs and the MC load—if the load of the MC is exceeded. It can be formally defined as

$$\mathfrak{C}_Q = \mathcal{P} + N_{\mathrm{sc}} \varsigma \kappa \Lambda_1, \tag{3.47}$$

where $\mathfrak{C}_Q$ is the penalty, κ is a penalty factor, and $\varsigma \in \{0, 1\}$ indicates whether or not the MC is overloaded.

Complexity

The most important benefit of the proposed solution is to greatly reduce the complexity of finding a good operating point. Since there are N_{sc} SCs which could be switched off/on in any combination, an exhaustive search approach would have complexity of $O(2^{N_{\mathrm{sc}}})$, while the proposed approach only keeps track of $2N_{\mathrm{sc}}$ actions, and therefore has a complexity of $O(N_{\mathrm{sc}})$.

However, note that the proposed solution does not strictly guarantee the constraint in (3.33). This is important to give the RL algorithm the chance to explore different actions and learn what is not good. By incorporating $\varsigma\kappa$ into the penalty function, the RL algorithm can be influenced not to violate the constraint, and thus provide a good QoS to the users whilst seeking the best policy regarding energy consumption.

3.2.6 Performance Evaluation

In this section, the employed data set, the benchmark methods developed, the performance evaluation metrics, and the obtained results will be discussed comprehensively. The simulation parameters are provided in Table 3.4, while Table 2.2 presents the power consumption characteristics of the BSs used in the simulations.

Table 3.4: Simulation parameters

Parameter	**Value**
SARSA with VFA	
Chance of choosing random action, ε	0.8
Learning rate, α	10^{-7}
Discount factor, φ	0.9
Inter-space between two consecutive actions, A_{is}	2
Minimum number of iterations for stopping, j_{min}	10
Threshold for the feature scaled cost, Ω	5×10^{-2}
Consecutive iterations ensuring stopping criteria, j_{rep}	10
Maximum number of iterations	100
General	
Number of time slots	144
Number of days	1
Number of grids considered for the MC	2
Number of grids considered per SC	1
Bandwidth for the MC	20 MHz
Bandwidth per SC	20 MHz

Data Set

In order to calculate the power consumption through (2.15), Λ_i is required for BS $\mathcal{B}_i \in \{1, 2, ..., N_{\text{sc}} + 1\}$. In this regard, to obtain Λ_i values, a real CDR data set provided by Telecom Italia[27] is used in this work, in which the city of Milan is divided into 10,000 square-shaped grids with a dimension of 235 × 235 meters. Then, within each grid, user call, text message, and Internet activity levels were recorded with 10-minute resolution for a 2 months period (November and December 2013). Even though the data set consists only of unitless activity level values and there is no information provided regarding the data processing phase, the activity levels can be interpreted as grid-wise relative traffic loads, since they represent the volume of user-mobile network operator interaction at each time slot. In the data processing phase of this work, first, the aforementioned individual activity levels (i.e., call, text message, and Internet) are combined. Then, two random[28] grids for the MC and one random grid for each SC are picked. Note that the activity levels at the two grids selected for the MC are

[27] Data is available online at https://dandelion.eu/datamine/open-big-data/.

[28] The reason for choosing the grids in a random manner is to avoid similar activity level characteristics of the grids that have close proximity.

further combined to create a traffic load data. After that, all the activity levels are normalised together between 0 and 100, and the obtained values are treated as traffic loads for each cell.

Benchmarking

Four different benchmark methods are used for comparison purposes, and they will be elaborated individually in the following paragraphs.

Sorting: Inspired by [248, 249], the sorting algorithm is developed to compare the results of the proposed algorithm. In this method, the SCs are sorted in ascending order based on their load factors, Λ. Then, they are switched off sequentially until there is no available capacity left at the MC, and the rest of the SCs are kept *on*. Given the power consumption profile in (2.15) and the characteristics of different types of BSs in Table 2.2, the MC consumes more power than the SCs for the same value of Λ. As such, it is wiser to switch off a SC with smaller traffic load in order to save more energy. This concept lies at the heart of the sorting algorithm, since it aims at minimising the energy consumption of the network. On the other hand, as the SC switching off is performed only when there is enough capacity at the MC, this method also guarantees the service of the users after the offloading process.

All-*On* method: There is no switching implemented in this method, meaning that all the SCs are always kept *on*. Accordingly, no offloading is needed as well in this case provided that none of the SCs are switched off at any time. Therefore, it can be inferred that there is no concern of QoS in this method, since all the users are served by the BSs (either MC or SC) that they were associated in the first place.

All-*Off* method: In this method, the SCs are always kept switched off and their data traffics are offloaded to the MC. However, this method is performed blindly, meaning that the data traffic of the SCs are offloaded to the MC regardless of its available capacity. This means that the users, which are normally served by SCs, are vulnerable to service disruptions, since there is no guarantee that they will be served by the MC. Even if the service is provided by the MC, the QoS would be reduced in case there are more users than the available capacity, and in that case the MC reduces the available resources for each user by certain amount in order to keep all the users served.

Exhaustive search: This is a method that tries to find the best policy among the set of all possible switching options consisting of the *off*/*on* states of the SCs. In particular, given the available capacity of the MC as a constraint, this method searches for the option with the least energy consumption. Hence, this method guarantees the service for each user in the case of offloading, which in turn prevents the QoS of the users from being violated. Note that exhaustive search returns the optimum policy, and thus the objective of any algorithm should be to mimic it as much as possible.

Performance Metrics

In this section, the metrics, which are used to evaluate the performance of the proposed algorithm and the benchmark methods, are presented.

Gain: In this work, the percentage gain on the total energy consumption compared to All-*On* method is considered. It is calculated as

$$G = \frac{\mathsf{E}_{\text{on}} - \mathsf{E}_{x}}{\mathsf{E}_{\text{on}}}, \tag{3.48}$$

where E_{on} and E_{x} are the total energy consumption in joules with All-*On* method and with one of the other methods, such that $\mathsf{E}_{x} \in \{\mathsf{E}_{\text{on}}, \mathsf{E}_{\text{es}}, \mathsf{E}_{\text{sort}}, \mathsf{E}_{\text{vfa}}\}$, where E_{off}, E_{es}, E_{sort}, and E_{vfa} are the total energy consumption in joules with All-*Off*, exhaustive search, sorting, and the proposed VFA-based methods, respectively.

Power consumption: Power consumption in watts during the simulation time is obtained for each method. This is a beneficial metric to evaluate the performance of the methods, since it reflects the variations in power consumption for different times of a day. Moreover, given that the gain is calculated on the energy consumption by accumulating the power consumption during the simulations, which can also be interpreted as upsampling, the detailed behaviours of the developed methods are kind of lost. Thus, power consumption is also a utilitarian metric that paves the way for detailed behavioural observations.

Average RAN throughput: The total required RAN throughput, $\mathfrak{T}_{\text{T,r}}$, is calculated by combining the throughput required from each cell as follows:

$$\mathfrak{T}_{\text{T,r}}(t) = \sum_{i=1}^{N_{\text{sc}}+1} \mathfrak{T}_{\text{r},i}, \tag{3.49}$$

where $\mathfrak{T}_{\mathrm{r},i}$ is the required throughput from $\mathcal{B}_i$, and is calculated as follows:

$$\mathfrak{T}_{\mathrm{r},i}(t) = \mathfrak{t}_{\mathrm{u},i}(t) N_{\mathrm{u},i}(t), \tag{3.50}$$

where $\mathfrak{t}_{\mathrm{u},i}(t)$ is the average throughputs for users allocated by $\mathcal{B}_i$ and where $N_{\mathrm{u},i}(t)$ is the number of users served by $\mathcal{B}_i$ at time t.

However, there is one caveat that since the backhaul capacity of the cells is limited by the installed backbone, and thus cannot be exceeded, $\mathcal{B}_i$ penalises the throughput for each user by Υ_i when the combined demand of the users exceeds $\mathfrak{T}_{\mathrm{p},i}$, the maximum installed capacity of $\mathcal{B}_i$, such that

$$\hat{\mathfrak{t}}_{\mathrm{u},i}(t) = \mathfrak{t}_{\mathrm{u},i}(t) - \Upsilon_i(t), \tag{3.51}$$

where $\hat{\mathfrak{t}}_{\mathrm{u},i}(t)$ is the average throughput for users allocated by $\mathcal{B}_i$ at time t after penalisation. This also ensures the condition in (3.34).

The throughput penalty, Υ_i, is calculated as

$$\Upsilon_i(t) = \begin{cases} \dfrac{\mathfrak{T}_{\mathrm{r},i}(t) - \mathfrak{T}_{\mathrm{m},i}}{N_{\mathrm{u},i}(t)}, & \text{if } \mathfrak{T}_{\mathrm{r},i}(t) > \mathfrak{T}_{\mathrm{m},i} \\ 0, & \text{otherwise,} \end{cases} \tag{3.52}$$

Next, as explained in Appendix A, normalised throughput is represented by the load factor in this work, thereby the provided normalised network throughput is given as follows[29]:

$$\tilde{\mathfrak{T}}_{\mathrm{T,p}}(t) = \sum_{i=1}^{N_{\mathrm{sc}}+1} u(-\Lambda_i(t) + 1)\Lambda_i(t) + u(\Lambda_i(t) - 1), \tag{3.53}$$

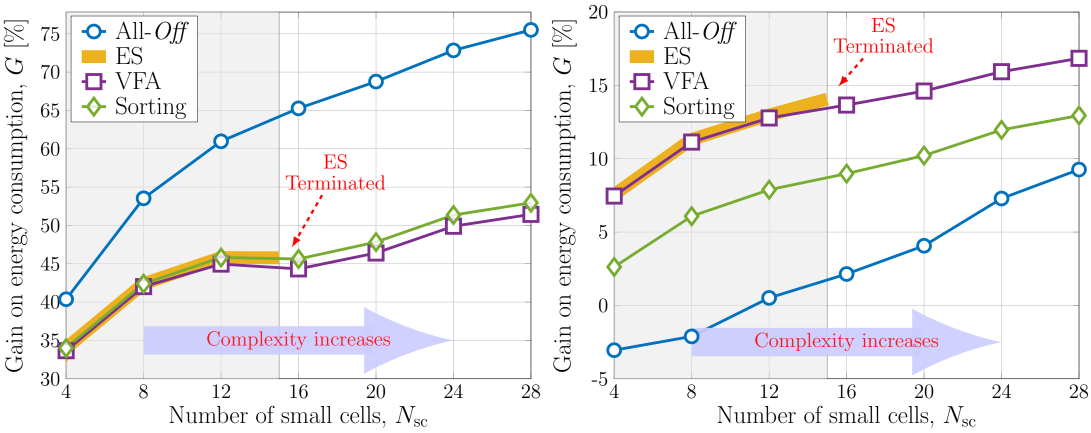
where $\tilde{\mathfrak{T}}_{\mathrm{T,p}}(t)$ is the normalised throughput of the network and $u(\cdot)$ is the unit step function, such that

$$u(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0. \end{cases} \tag{3.54}$$

Scenarios

The developed benchmark methods and the proposed VFA-based switching algorithm are tested in two different scenarios, namely Scenario A and Scenario B.

[29] Refer Appendix A for the details of the derivation.



(a) Results for Scenario A when $\kappa = 20$.

(b) Results for Scenario B when $\kappa = 10$.

Figure 3.8: Percentage gain performances compared to the All-*On* method for Scenario A and Scenario B.

Scenario A: This is a simplistic scenario, where there is only one type of SC (micro) in the network. Moreover, the sleep mode power consumption for the SCs are assumed to be zero, such that the SCs are not contributing to the total power consumption of the network at all when they are switched off.

Scenario B: In this scenario, four different types of SCs, e.g. micro, RRH, pico, and femto, are deployed in the network, and the number of SCs are distributed in these four types almost[30] equally. Moreover, the sleep mode power consumption is not assumed to be zero in this scenario, instead the values in Table 2.2 are used. Therefore, this scenario is more realistic than Scenario A, as there are heterogeneous combinations of SCs in real networks and the sleep mode power consumption is not zero.

Results

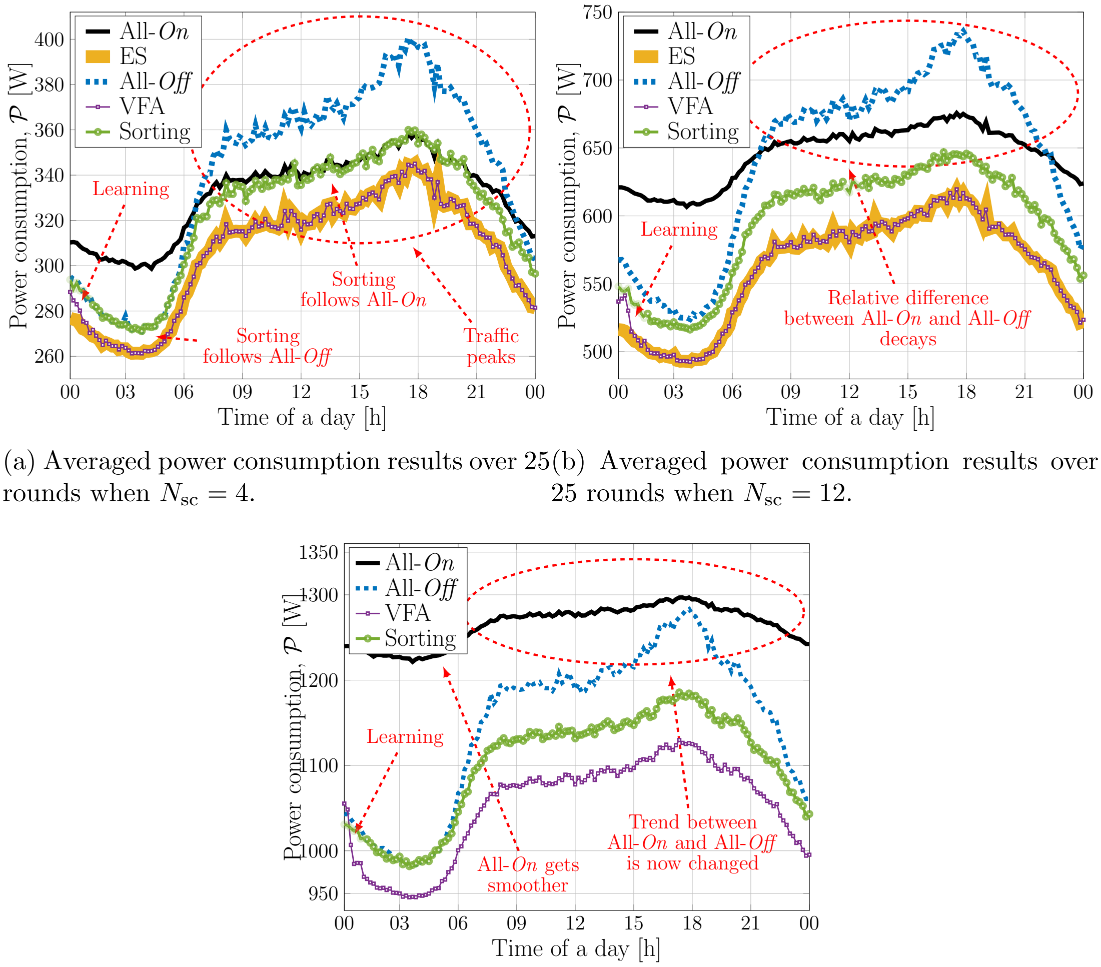
As explained in Section 3.2.6, Fig. 3.8 demonstrates how much gain in energy consumption is obtained when All-*Off*, exhaustive search, sorting, and proposed VFA-based methods are compared to the All-*On* method. Fig. 3.8a shows the gain results for Scenario A, while the results for Scenario B are presented in Fig. 3.8b. Note that since the exhaustive search method is computationally demanding with $O(2^{N_{\mathrm{sc}}})$, where it doubles the elapsed time when N_{sc} is incremented by 1, it is allowed to run only until $N_{\mathrm{sc}} = 15$ for both Fig. 3.8a and Fig. 3.8b. The idea in

[30] The number of SCs in the network are changed to observe the impacts of SC volume on the performance, and it is not always possible to distribute them equally, since the amount of SCs are sometimes not divisible by four, which is the number of SC types in the network. For example, when the number of SCs are 13, then the distribution becomes 3,3,3,4.

Fig. 3.8 is that the performances of All-*Off*, sorting, and the proposed method are compared to the exhaustive search, which is optimum, in a smaller network due to the time requirement for exhaustive search when N_{sc} increases. Then, once an idea is obtained about the performances of the methods compared to the optimum, the exhaustive search is terminated, and the other methods are kept running in order to observe further behaviours in larger scale networks.

It can be seen in Fig. 3.8a that the sorting method is working exactly same with the exhaustive search, while the proposed method follows them quite closely. These outcomes can lead to a conclusion that the proposed method is outperformed by a more simpler and straightforward algorithm, and one can question the validity of the proposed method. However, when Fig. 3.8a is reconsidered together with Fig. 3.8b, it can be inferred that the better performance of the sorting algorithm is not generalised: while it gives promising results in simplistic Scenario A, it starts under-performing in the complex and realistic Scenario B. On the other hand, given that the proposed VFA-based method gives very close results to the exhaustive search in both scenarios, it seems quite immune to the changes in the scenario with a generalised good performance. In other words, although there might be simpler alternatives to the proposed method in basic scenarios, which are mostly unrealistic,—owing to the nature of the developed VFA-based RL algorithm—the proposed method takes the advantage of being capable of generalisation and works properly even when the scenario becomes more complicated and realistic. This, in turn, makes the proposed method work properly regardless of the conditions, while the sorting method, for example, relies on the simplicity of the scenario, making it impractical for realistic scenarios, where the assumptions in Scenario A are no more valid.

Another point about Fig. 3.8 is that the gain decreases significantly when the scenario is switched from A to B. For the proposed method, for instance, the gain drops from around 52% to 17% when $N_{\mathrm{sc}} = 30$, which yields around 67% reduction. This is mainly due to more heterogeneity of Scenario B, where there are four types of SCs. Scenario A includes only micro cell, which is the second most energy consuming SC after RRH according to Table 2.2, making the SCs in Scenario A consume a considerable amount of energy. For Scenario B, on the other hand, there are four types of SCs with distinctive power profiles, and thus total power consumption decreases because of the inclusion of pico and femto cells, which consume small amount of energy, in the network. Besides, while SCs still consume energy when they are switched off in Scenario B, the sleep mode power consumption is assumed to be zero for Scenario A, making it consume overall less energy when switching is performed.



(a) Averaged power consumption results over 25 rounds when $N_{\mathrm{sc}} = 4$.

(b) Averaged power consumption results over 25 rounds when $N_{\mathrm{sc}} = 12$.

(c) Averaged power consumption results over 25 rounds when $N_{\mathrm{sc}} = 28$.

Figure 3.9: Power consumption performances of the developed methods for various number of SCs.

The last point that is worthy to discuss about the findings in Fig. 3.8 is related to the All-*Off* method. While it outperforms all the other method in terms of gain in Fig. 3.8a, it becomes the worst-performing method in Fig. 3.8b. The reason behind this phenomenon is again the characteristic diversity between Scenarios A and B. There is only micro cell, which is demanding in energy, in Scenario A, and thus switching off SCs almost always result in less energy consumption, whereas, due to the heterogeneity of Scenario B, the optimal policies for switching off are different from each other for each type of SC. In particular, (3.35) holds for larger $\Lambda_j,\ j > 1$ values when the type of SC goes from femto cell to RRH. This means that the number of cases, where switching off is profitable, is larger for micro cell than that of femto and/or pico cells. Therefore, since the All-*Off* method switches off all the available SCs regardless of their types, the overall process becomes less profitable in Scenario B when compared to Scenario A.

Fig. 3.9 reveals the power consumption behaviours of the developed methods for various numbers of SCs included in the network. Similar to the results in Fig. 3.8, for consistency, the power consumption results for integer multiples of four is presented, where the types of SCs are distributed equally. Hence, N_{sc} value is altered as $N_{\text{sc}} = 4$, $N_{\text{sc}} = 12$, and $N_{\text{sc}} = 28$, respectively. Note that Fig. 3.9c does not include the exhaustive search method, since it is only allowed to run until $N_{\text{sc}} = 15$ for computational complexity reasons.

Fig. 3.9a demonstrates the power consumption performance of the developed methods when $N_{\text{sc}} = 4$. The findings suggest that the proposed VFA based algorithm manages to mimic the exhaustive search almost perfectly apart from the initial learning phase. Given that the computational complexities for the exhaustive search and the proposed method are $O(2^{N_{\text{sc}}})$ and $O(N_{\text{sc}})$, respectively, these results confirm that the proposed method performs quite well (i.e., producing near-optimal results) with drastic decrease in computational complexity.

It is also worth discussing the learning phase of VFA algorithm, which is common in all the three cases of N_{sc}. This behaviour is expected as an online learning framework is proposed, where VFA-based cell switching method is deployed without any prior knowledge, and thus it learns by interacting with the real environment. In other words, the promising performance after the initial learning phase is due to the experience obtained during the training. The slightly worse initial performance is due the fact that VFA takes more random actions in the beginning in order to increase the knowledge about the environment–as commonly referred to as *exploration*. Then, after the exploration, the randomness in the actions taken decreases with the number of episodes in order to let the VFA start using the information it received, which is known as *exploitation*. One important point here is that the exploration process in the developed model takes a short amount of time, making the online implementation feasible, since longer learning phases would undermine the advantage of the VFA based solutions. The reason behind preferring the online implementation over the offline one is that the former case is model-free, where it does not require any prior knowledge, while a full environmental knowledge is needed in the latter. This, in turn, renders the online implementation more practical. Therefore, even though the offline implementation is free from the possible negative impacts of the training process, it is comparatively less functional in real scenario, where full prior knowledge is often inaccessible.

Another interesting aspect that can be deduced from Fig. 3.9a is the behaviour of the sorting method. It mirrors the All-*Off* method in the beginning, where the data traffic is relatively low, whereas it starts following the All-*On* method when the traffic load increases. This is because the sorting method manages to switch off most (even all) SCs—which is similar (same) behaviour with the All-*Off* method—when the traffic load is lighter, since there is more offloading opportunities in the MC owing to the low traffic patterns. Moreover, the amount of offloaded data is also comparatively less at these times. When the traffic volume becomes higher, on the other hand, since the amount of data to be offloaded and the occupancy of the MC increase simultaneously, the switching off becomes much harder (even impossible), which is quite similar to the behaviour of the All-*On* method.

Fig. 3.9b presents the power consumption results when $N_{\text{sc}} = 12$. Similar to Fig. 3.9a, the sub-optimal results of VFA is observed during the training phase. However, other than being short in time, the appreciable thing about this training phase is that the results that VFA produces are still reasonable even though it is not optimal. Another interesting observation is the relative difference between the All-*On* and All-*Off* methods shrinks compared to the one in Fig. 3.9a. Taking into account the peak points, the relative difference between the All-*Off* and All-*On* methods decreases by around 25% when N_{sc} increases from 4 to 12.

Two important questions arise from these findings. First, why does the All-*Off* method result in more power consumption than the All-*On* method? It is counter-intuitive to observe such results where switching all the SCs off causes more power consumption than always keeping all the SCs *on*. Moreover, it is observed that All-*On* outperforms All-*Off* especially when the traffic loads are higher. The rationale behind this is that, as repeated previously, considering (3.35) together with Table 2.2, it is usually non-profitable to switch off SCs when the traffic load is above some certain threshold, which is different for each type of SC. Therefore, it is not a rule of thumb that the switching off is always resulting in less power consumption. The outcomes in Fig. 3.8b, where the All-*Off* method gives negative gains, also confirm this conclusion. Nonetheless, this is not the only condition that makes All-*On* more favourable than the All-*Off* method in terms of power consumption. It is also the intensity of the SCs in the network. Since N_{sc} is not large enough in both Figs. 3.9a and 3.9b, the contribution of the SCs to the power consumption is comparatively less than that of the MC, therefore the overall energy saving resulting from switching off SCs cannot prevail against the loss caused by offloading traffic to the MC.

Second, why does the relative difference between the All-*Off* and All-*On* methods decay when N_{sc} rises up from 4 to 12? The answer for this question is related to the last discussion for the previous question; since the intensity of the SCs increases in the network with increasing N_{sc}, the dominance of the MC in the total power consumption scales down. This subsequently renders the gain resulting from switching off more significant, and thus the All-*Off* method starts being more reasonable. Hence, the number of instances that All-*On* outperforms the All-*Off* method also decreases when N_{sc} is increased from 4 to 12. The results in Fig. 3.8b again supports this conclusion, as the percentage gain enhances with increasing N_{sc}.

Fig. 3.9c showcases the power consumption results when $N_{\text{sc}} = 28$. It is again worth noting that exhaustive search is not included this time, since it is run until $N_{\text{sc}} = 15$, owing to the computational complexity concerns. Unlike the results in Figs. 3.9a and Fig. 3.9b, there is no point in Fig. 3.9c, where All-*Off* outperforms All-*On*. Similar to the previous discussions on this topic, the distribution of the total power consumption among the SCs and the MC is an integral aspect of the performances of the All-*On* and All-*Off* methods. Given that the SCs now consume considerable amount of energy due to their increased number, the gain obtained from switching off the SCs—which are profitable to switch off according to (3.35)—prevails over the loss incurred by switching off the SCs—which are non-profitable to switch off.

Another interesting point about the findings in Fig. 3.9c is that the power consumption for the All-*On* method becomes smoother compared to the results in Figs. 3.9a and 3.9b. The relative peak-to-peak difference, for example, was around 20% when $N_{\text{sc}} = 4$, whereas it drops to 6.2% when N_{sc} increased to 28. This is again because of the MC loosing its dominance in the total power consumption. While, when fully-loaded, 72.3% of the power consumption comes from the load dependent part for the MC, this rate is around 28% on average (minimum: femto cell with 8.3%, maximum: RRH with 66.7%) for the SCs. In other words, the MC consumes more on the load dependent part, whereas SCs consume more on the static power. Therefore, for the smaller N_{sc} values, the load dependent power consumption is higher as the MC is the main contributor to the total power consumption, while the load dependent power consumption becomes relatively less and the static power gets more significant for higher values of N_{sc}.

Fig. 3.10 shows normalised average RAN throughput, which is calculated through (3.53), for various κ values of VFA, the All-*On* and All-*Off* methods. Furthermore, the activity levels in the data set are assumed to be in Mbps after the pre-processing detailed in Section 5.5. The objective of demonstrating these results is to highlight the impact of κ value on the performance of the proposed VFA based switching algorithm. In addition, the findings also display the cost of switching off all the SCs without taking into account the available capacity at the MC. As such, the results suggest that there is an upper bound for the All-*Off*

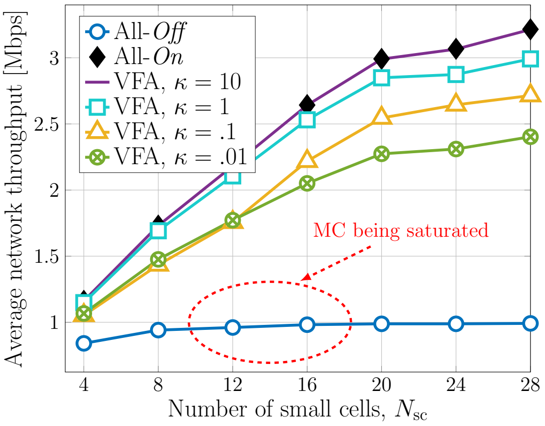


Figure 3.10: Results for normalised network throughput against the number of SCs. The network throughput is calculated by averaging out the obtained throughput values at each time slot during the simulation period.

method, since it only relies on the capacity of the MC, and after normalisation, each BS (MC or SC) has a capacity of unity at maximum. Given that the MC is the only BS that is kept *on*, All-*Off* switches off all the SCs and offloads their traffic to the MC, meaning that it has only one unit of capacity available in the network. Therefore, these results also confirm that having a blind policy, that is, acting without considering the environmental conditions and/or constraints, is not a wise idea, since it results in the degradation of the QoS of users as well as being more costly in power consumption on some occasions, as already proven in Figs. 3.8b, 3.9a, and 3.9b. The purpose of presenting the results for the All-*On* method is to demonstrate the best case (upper bound) that can be achieved in terms of throughput.

As seen in Fig. 3.10, promising results are obtained with the proposed algorithm. It gives quite close results to the All-*On* method, proving a good performance as it is producing similar results to the best case. By considering the findings in Figs. 3.8, 3.9, and 3.10 together, it is easy to deduce that the proposed VFA based switching algorithm performs outstandingly good in terms of both power consumption and throughput, since it reduces the power consumption (similar to exhaustive search) without compromising on the QoS of the users (similar to the All-*On* method). Provided that exhaustive search and All-*On* are the best methods in terms of power consumption and QoS, respectively, the proposed method combines the advantages of both.

Fig. 3.10 also showcases the impact of κ in (3.47) on the performance of the developed VFA model. The results suggest that the throughput performance of the proposed VFA decreases with decreasing values of κ. This is because higher values of κ will result in more penalty being incurred for the case when the demanded capacity exceeds the available capacity at the MC. As explained in Section 3.2.6, when the demanded capacity is higher than the available one, the network reduces the allocated bandwidth for each user in order to accommodate the demands of all users. In this regard, the agent refrains from switching off a SC—whose demanded capacity is larger than the available one at the MC—which in turn helps to keep the QoS above the required level. However, for the smaller values of κ, the agent starts following more relaxed policies on the given constraint, where it takes more actions that are against the above-mentioned demanded/available capacity criterion. Hence, the obtained throughput starts decreasing for lower values of κ. For the extreme scenario, where $\kappa = 0$, then (3.47) becomes $\mathfrak{C}_Q = \mathcal{P}$, meaning that the agent only focuses on the total power consumption and does not care about the constraint of available capacity at the MC. Intuitively, the agent would be reducing the total power consumption as much as possible at the expense of QoS degradation.

3.2.7 Summary

In this work, an RL-based solution is presented for cell switching, which is capable of learning the best policy in a dense HetNet environment, in order to save energy and satisfy the QoS at the same time. The proposed solution is evaluated using real data from Milan, Italy and compared with various benchmark methods. The results in terms of power and energy consumption show that the proposed method can perform just as well as the exhaustive search method, which produces the optimum solutions, regardless of the complexity and size of the given scenario/environment. Moreover, the proposed method resulted in much fewer computations than that of exhaustive search, meaning that it is a scalable method. Furthermore, the RAN throughput was also measured, and it was observed that the proposed method gave similar results with the All-*On* method, which is the best in terms of the QoS due to the fact that it does not include any offloading. Therefore, the proposed VFA based cell switching method resulted in a significant reduction in the network's energy consumption without much compromise on the QoS, thus making it suitable for practical application.

Chapter 4

Capacity Enhancement for Cellular Networks

This Chapter discusses the capacity enhancement concept from two different perspective, namely: predictive HO management and intelligent UAV positioning. First, in the predictive HO management part, two potential issues regarding Markov chains based HO predictions are identified and exclusive solutions are proposed for each, as in [250–252]. Second, in the intelligent UAV positioning part, a k-means clustering based UAV positioning algorithm is developed and obtained results are analysed.

4.1 Improved Markov Chains based Predictive Mobility Management

4.1.1 Introduction

In cellular communication networks, UEs perform measurements on some signal quality indicators; e.g., SINR, received signal strength indicator (RSSI), and RSRP, from both serving and neighbouring BSs in an effort to keep their signal qualities above a certain level by deciding whether an HO is required. Upon the decision of an HO after meeting multiple conditions, including hysteresis and TTT, some certain steps are required to be taken under three main phases, namely: preparation, execution, and completion [168]. Nonetheless, all these phases and steps take some time and causes HO latency as well as signalling overhead [133, 251], thereby predictive HO schemes have been proposed in the literature in order to overcome these problems [59, 253–257].

These schemes mainly try to predict future HOs of users in order to perform some of the aforementioned HO steps prior to the occurrence of the HO so that the HO signalling cost can be reduced. Even though these predictions can be provided by various methods, including statistical analysis, ML, and data mining algorithms, but ML is the one that is commonly employed [59]. Of diverse set of available ML algorithms, one of the most popular algorithms in the literature is found to be Markov chains—a class of stochastic process, characterised by a state space and a transition matrix, given an initial distribution [133]. In the context of mobility management, the states of a Markov chain can be seen as a user-cell association, in which the BS identification corresponds to the state. Moreover, the transition matrix would then represent the probability of a user transitioning from the current BS to one of its neighbours, and the initial distribution can be given according to used mobility data, for example.

Despite its popularity, Markov chains still have problems that need to be addressed in order to provide a reliable performance under different conditions. One of the main problems is their path dependency; i.e. when the trajectory of a user includes revisits[1] to pre-visited locations, the prediction accuracy becomes very prone to drop dramatically. This problem is mostly caused by the characteristics of transition matrix, which Markov chains based predictors use in their algorithms. A conventional transition matrix is two-dimensional and includes transition probabilities from one state to another, and thus whenever a user traverses from one cell to another, the state with the highest probability is predicted as the next state. Therefore, if two states have very similar probabilities, which can occur when revisits happen in the user trajectories, conventional transition matrices are very prone to be confused, resulting in increased chances of incorrect predictions.

Another important issue with the conventional Markov-chains based HO predictors is that their prediction making criteria is quite primitive, such that they solely rely on holding the highest transition probability to make the next state prediction. This could be sufficient to some extent, but for larger networks sizes, where the number of HO possibilities increases—especially with the dense SC deployments in 5G, this is quite expected—, this only criterion would be inadequate owing to the fact that the probabilities in the transition matrix are likely to scale down. This results in making prediction with a lower confidence[2], which subsequently degrades the prediction performances. Therefore, in this chapter, these two problems (i.e., path-dependency and insufficient prediction criteria) of the conventional Markov-chains based HO predictors are formularised along with

[1] Revisit is defined as a situation whereby a user visits the same cell more than once.

[2] The transition probability is treated as a confidence level here, as they are strongly correlated to each other.

the proposed solutions.

4.1.2 Related Work

There are numerous studies performed in the literature related to Markov chains based predictors for wireless communications. For example, in [258], the authors employ Markov chains for mobility prediction. A classical Markov chain is employed, and the authors demonstrate the usability and suitability of the predictive mobility management for LTE networks with femtocell deployments.

In [126,133], the authors introduce a discrete-time Markov chains (DTMC) in order to manage HOs for CDSA in LTE networks. In the proposed method, they try to predict the next position of a user in order to reduce the signalling costs incurred by HOs. The LTE X2 HO procedure is assumed in their study, wherein it is indicated that a non-predictive conventional HO procedure is better than an incorrect prediction in terms of HO signalling cost. Their work, which will be referred to as DTMC hereafter, will be used in this present work for comparison purposes, since it is a good example of the traditional implementation of Markov chains based HO predictions.

A hybrid Markov based model for human mobility prediction is presented in [259], which suffers from requiring fast and effective algorithms in order to select the optimal available network. A Markov renewal process, which is based on a semi-Markov model, is proposed in [260] by considering both the location and sojourn time of a user. As a key point in their work is that rather than predicting only the next time slot, the authors are able to predict multi-step ahead locations of a user, boosting the future visibility that in turn helps in making more informed and efficient arrangements. In [261], a novel Markov chains based predictive mobility management algorithm is presented, in which the authors attempt to enhance the prediction performance of Markov chains for both new and high-random users[3].

4.1.3 Objectives and Contributions

Conventional Markov chains based HO predictors are quite primitive and vulnerable to producing errors due to the fact that, as seen from (2.13), they only rely on the transition matrix and the initial distribution while making predictions. This brings multiple limitations, which undermines their practical implementation especially for applications that are slightly tolerant (or intolerant) to errors. For the case of HO prediction, it is already obvious from Chapter 2 that making

[3] Users with high randomness/less regularity in their trajectories.

incorrect predictions incurs more HO signalling cost than that of conventional non-predictive HO process[4]. Based on that, it is quite crucial to minimise the number of incorrect predictions in order to make the predictive HO process more advantageous.

Due to their simplistic structure, Markov chains based predictors are subject to have the following problems:

- the transition matrix can output similar/same transition probabilities, making the predictor confused, thereby increasing the likelihood of errors;
- the algorithm identifies the maximum element in the output probability matrix as the next state without making further investigations/checks. This is problematic—especially for the scenarios where many states are available—owing to the fact that a state with a very small transition probability can be predicted as the next state as long as it is the maximum among the others.

In this study, these two problems are modelled and addressed independently in order to better observe the behaviours of each single solution. First, the aforementioned first problem (i.e., revisit oriented close transition probabilities) is formularised, followed by the proposed solution. In particular, revisits are found to be affecting the prediction performance of Markov chains based HO predictors employed in the literature, such as the one in [133]. In order to tackle this problem, a structural change for the Markov chains based HO predictions is proposed, such that instead of storing the traditional 2-D one, a 3-D transition matrix idea is introduced, where each layer in the third dimension is a conventional 2-D transition matrix associated to a particular HO within a day. In other words, each HO—in terms of the order; e.g., first, second HO, etc.—has its own 2-D transition matrix, in which the HOs with the corresponding order are processed. Therefore, since an additional contextual information; i.e., the orders of HOs, is made available to the predictor, they start making more informed predictions, which subsequently improves the prediction accuracy.

This is a novel idea given that, to the best of the author's knowledge, this kind of transition matrix style has not been proposed in the literature. Although higher-order Markov chains would help in addressing the same problem, the size of their transition matrix grows exponentially with the increasing order. The proposed method, on the other hand, is more scalable provided that the size of the transition matrix increases linearly with increasing number of states. In order

[4] Correct HO process in the best among these three cases (i.e., correct prediction, incorrect prediction, and no-prediction) in terms of HO signalling cost.

to demonstrate the performance of the proposed method, numerical simulations are conducted, where a HO predictor with the proposed 3-D transition matrix is compared to the 2-D one. The algorithms are tested with and without revisits inclusions in the trajectory, and the obtained results confirm that the proposed method is more robust to the changes in the user trajectory, as its prediction performance is affected less from the revisit inclusions.

Second, the insufficient prediction criteria problem is modelled and a corresponding solution is proposed. As such, when the number of states increases, the transition probabilities are likely to scale down, which makes Markov chains based predictors prone to produce errors while making HO predictions. This is because conventional predictors rely on the maximum element in the probability matrix, which might not necessarily be high enough. In other words, for the conventional Markov chains based HO predictors, as in [133], holding the maximum probability is the only condition to be predicted as the next state. However, as mentioned earlier in this section, making incorrect prediction is more expensive that making no prediction in terms of HO signalling cost, yielding that there should be a confidence level at which the predictive HO is triggered.

In this regard, a threshold based prediction concept is presented in this work, that is, the conventional non-predictive HO procedure in triggered in case none of the states holds the developed threshold. This is done to protect the predictive HO case from being disadvantageous over the non-predictive one. Towards that end, an optimisation problem is developed for determining the threshold by considering the HO signalling cost. After analytical threshold derivations, numerical simulation campaigns are conducted to test the proposed method. The obtained results show that the proposed approach outperform the conventional Markov chains based HO predictor in terms of HO signalling cost due to the fewer number of predictions: the non-predictive HO procedure is processed if none of the candidate states could meet the threshold confidence level.

4.1.4 Problem Formulation

Before detailing the identified problems with conventional Markov chains based mobility predictors, it is better to give a brief recap on their characteristics. In general, Markov chains—fundamentally defined as a stochastic process—construct a finite number of states and corresponding transition probabilities that represent the likelihood of moving one state to another. These probabilities are stored in a matrix that is commonly named as transition matrix. As mentioned earlier, Markov chains based predictors are linked to the probability theory because of their stochastic nature, as such, they can be employed as predictors, since the transition probabilities already provide the chances of traversing among different states. Based on that, using (2.13), the predictor first calculates a probability vector, which depends only on the transition matrix and the initial distribution, and then estimates the next state by observing the resulting probabilities and selecting the one holding the highest transition probability. Therefore, Markov chains based predictors are easy to implement and have proven capabilities in cellular networks as shown in [126, 133, 258–261].

However, despite their success, Markov chains have an inherit problem associated to them, that is, relying on a simple matrix in order to predict the next movement of a user. Because this matrix is constructed through a Markov process, it suffers from so-called memoryless property, which states that the conditional probability distribution of future states depends only upon the present state [262]. In other words, Markov chains based predictors are very prone to make incorrect predictions owing to the simplicity in their model. This is mainly because the type of information exploited during the training phase is only transition probabilities, which are built on historic transitions between the states, resulting in making predictions with a limited context/knowledge. This characteristics of Markov chains are susceptible to produce closer transition probabilities, which in turn increases the likelihood of making incorrect predictions. As such, the problem is twofold:

1. the algorithm itself is poorly immune to having similar/close probabilities due to multiple effects including ping-pong and revisit;

2. predictions are made without considering any confidence level; predictors simply select the state with the highest probability as the next state regardless of a confidence level.

In the following paragraphs, these two aspects will be discussed in a detailed manner.

Close Probabilities Due to Revisits

Due to the memoryless property of Markov chains, issues can arise whenever there are revisits in a given user's path. For example, given a user visiting the same cell twice—but performs the next HO to different BSs—thereby the probability of going to either one of the cells is of 50% at the end of the learning stage. As such, it can be seen that despite utilising a Markov chain to learn this transition probability, there is no meaningful information learned and its performance would be the same as that of a random prediction. Towards that end, it is clear that although Markov chains can work very well for mobility prediction whenever the trajectory of a given user does not include revisits, problems can arise if revisits occur, whereby additional solutions become necessary [250].

Lack of a Confidence Level

While making predictions, the Markov chains based methods first trigger (2.13), and they identify the state with the highest probability as the next state. In other words, the selection of the next state is performed by using only the max function, such that $\max(\vec{p}_m)$, thereby the predictions are made blindly to the confidence level[5] of the predictor, which can play an important role on the overall performance [251].

This concept would be easier to explain this concept with an example. Assume that a user is currently located in State A, while States B, C, D, and E are being its neighbouring states[6]. Let's have two different cases [251]: in the first case, imagine the following probability matrix is obtained after employing (2.13) (the order of the elements goes from State B to State E):

$$\vec{p}_m = [0.90\ 0.08\ 0.01\ 0.01]. \tag{4.1}$$

According to the common practice, as it is done in [126, 133] for example, the predictor selects State B as the next state, since the transition probability from A to B is the highest.

In the second case, let the probability matrix be different from the first case as follows [251]:

$$\vec{p}_m = [0.30\ 0.25\ 0.25\ 0.20]. \tag{4.2}$$

The predictor selects State B in this case as well given that it holds the highest transition probability. Nonetheless, in this case, the transition probabilities from State A to other states are very close to each other. Moreover, the probability of going to State B was 0.9 in the first case, whereas it is 0.3 for the second case. The point here is that although the predictor outputs State B as the next step in both cases, the confidence of the prediction is much higher in the first case. Therefore, the predictor is supposed to be designed in a way that it takes into account not only the state with the highest probability but also the confidence level.

[5] The maximum probability in $\vec{p}_m$ resulted from (2.13) is considered as the confidence level in this work, since it derived from the transition matrix.

[6] In cellular networks, due to the spatial correlation, a user can move from one state to only its neighbours. Therefore, that is the reason why the neighbouring state concept is adopted here.

4.1.5 System Model

As shown in Fig. 4.1, a cellular network environment with multiple cells—each served by a BS—are considered in this work. The users within this environment commute based on their daily life routines. As seen from Fig. 4.1 that each cell has multiple neighbouring cells, which is defined as being adjacent to each other. Therefore, let $\mathcal{B}_{s,t}$ from $\mathbb{B}$ denote the serving BS of a user at time t, at which the user is in cell i; such that $\mathcal{B}_{s,t} = i$. Moreover, let cell i have a neighbour list that is stored in a vector of $\mathfrak{N}_i$. Therefore, as it is done in [133], the conditions for moving to a next cell are as follows. First, the user is only allowed to move to a neighbour cell of the current serving cell:

$$P(\mathcal{B}_{s,t+1} = x|\mathcal{B}_{s,t} = i) = 0, \quad \forall x \notin \mathfrak{N}_i, \tag{4.3}$$

where $\mathcal{B}_{s,t+1}$ denotes the next cell of the user. Second, moving from and to the same cell is not allowed:

$$P(\mathcal{B}_{s,t+1} = i|\mathcal{B}_{s,t} = i) = 0, \quad \forall i \in \mathbb{B}, \tag{4.4}$$

where $\mathbb{B}$ is the set of all the considered cells/BSs. The entries in the transition matrix, $\vec{\mathcal{T}}$, is updated according to an online learning process detailed in [133]. For the sake of avoiding repetition, the online learning process will not be detailed in this thesis, but the interested readers are referred to [133] and the equations from (7) to (11) therein. In brief, after each move, the transition probability of the destination cell is incremented by the accumulated transition probabilities of the other non-zero neighbouring cells. However, this obtained accumulated number is controlled by a trajectory dependent parameter, α, which basically acts as the learning rate of the system (i.e., a smaller value of α updates the transition matrix slowly, while a larger value of α updates it more quickly). Then, the transition probabilities of the rest of non-zero neighbouring cells are decremented in a similar fashion[7].

Assuming a user is moving randomly, such as random walk mobility model, the next cell of the user would be uniformly distributed as

$$P(\mathcal{B}_{s,t+1} = y|\mathcal{B}_{s,t} = i) = \frac{1}{|\mathfrak{N}_i|} \quad, \forall y \in \mathfrak{N}_i, \tag{4.5}$$

where $|\mathfrak{N}_i|$ denotes the cardinality of the set $\mathfrak{N}_i$.

[7] See equations (7) to (11) in [133].

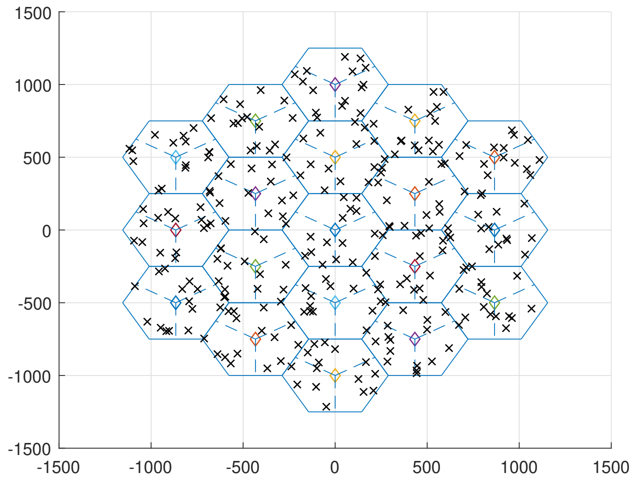


Figure 4.1: Considered system environment where there are multiple cells and users are moving around. Note that diamonds (◊) represent BSs, while crosses (×) are for users.

4.1.6 Proposed Methodology

As mentioned in Section 4.1.4, two different problems can arise from the simplistic characteristics of the conventional Markov chains:

- close probabilities due to revisits, and
- ignoring the confidence level while predicting.

Therefore, two independent solutions will be proposed for these two problems.

Three-Dimensional Transition Matrix

Two different approaches can be considered in order to mitigate the effects of revisits in Markov chains based mobility predictions: one is to include information about previous states, such as k-order Markov chains as in [263], while another is to include information about the order of states (i.e., the order of HO occurrences). Although k-order Markov models are good solutions to such problems, they usually suffer from huge computational complexity for higher values of k, where the prediction accuracy is prone to be degraded owing to the lower convergence rates [264].

Considering the case of adding the HO order information to a conventional Markov chain: this can be done by building a 3-D transition matrix, in which each row and column represent the probabilities between BSs, while the third dimension represents the order of the HO, such that the size of proposed 3-D transition matrix is $\vec{\mathcal{T}}_{\text{3-D}}^{m} = |\mathbb{B}| \times |\mathbb{B}| \times N_{\text{HO}}$, where $|\mathbb{B}|$ is the cardinality of $\mathbb{B}$ and

N_{HO} is the number of HOs per day [250]. In a more general manner, this idea can be described as a system that builds individual 2-D transition matrices for each HO in a day, and then all matrices are combined at the end to build a 3-D transition matrix. Thus, whenever revisits occur, a completely new 2-D matrix is analysed, and since the same BS cannot be revisited in a consecutive manner (otherwise there is no HO), the problem of revisits is solved.

Threshold Based Prediction Triggering

It is now obvious that there is an important issue with classical Markov chains based predictors when the highest transition probability is not sufficiently high. In order to mitigate this problem, a threshold is introduced in this study, where the predictors are allowed to make no prediction when the transition probabilities are low; i.e., the state with the highest probability fails to meet the pre-defined threshold. In other words, while holding the highest probability would be sufficient as a condition to make the next state prediction for conventional Markov chains based predictors, the proposed model requires two conditions [251]:

- holding the highest probability (as in the conventional methods), and
- meeting the threshold (the novel criterion).

If one of these condition is not met, the conventional non-predictive HO procedure is triggered. It is already known from Fig. 2.4 that, in terms of signalling cost, making no prediction is better than incorrect predictions, with the correct prediction is being the best [126, 133]. Let H_{n}, H_{i}, and H_{c} denote HO signalling costs for no-prediction, incorrect prediction, and correct prediction cases, respectively. Based on that, in [133], the expected total HO signalling cost is given as

$$E[H_{\mathrm{t}}] = A_{\mathrm{c}}H_{\mathrm{c}} + (1 - A_{\mathrm{c}})H_{\mathrm{i}}, \tag{4.6}$$

where H_{t} is the actual HO signalling cost, and A_{c} is the prediction accuracy.

Nevertheless, no-prediction case in not considered in (4.6). Given that the proposed approach results in no predictions in cases where the greatest transition probability does not meet the introduced threshold, the no-prediction case should be incorporated into (4.6). However, there is another modification needed in (4.6) before integrating the no-prediction case: the prediction accuracy is a measure of the confidence of the predictor, and it can be obtained through evaluating the prediction performance, but only after making some predictions. Moreover, according to the law of large numbers, measuring the accuracy with small size of measurements is not a good idea due to characteristics of random processes, where many observations (ideally infinite) are needed to obtain the underlying distribution. Similarly, in order to find the accuracy level in a correct way, the prediction performance should be measured over many observations[8]. Because the proposed threshold approach is to be implemented at each prediction instance, using of the accuracy as a confidence measurement is not a good idea. To this end, the maximum probability of the probability matrix, $\vec{p}_m$, is used in this study as a confidence level instead of the prediction accuracy, such that $A_\mathrm{c} \approx \vec{p}_{m,\mathrm{m}}$, where $p_{m,\mathrm{m}} = \max(\vec{p}_m)$. Thus, (4.6) is converted to

$$E[H_\mathrm{t}] = p_{m,\mathrm{m}} H_\mathrm{c} + (1 - \vec{p}_{m,\mathrm{m}}) H_\mathrm{i}. \tag{4.7}$$

Assume that the transition matrix of a Markov chain is known, and the probability matrix is obtained through (2.13) as $\vec{p}_m = [p_{m,1}, p_{m,2} \cdot, , , .p_{m,|\mathbb{B}|}]$. Since the predictor is now allowed to decide either making or not making predictions according to the threshold, both cases will occur with a correspondent probability. Thus, let p^+ and p^- be the probabilities of making and not making predictions, respectively, where $p^- = 1 - p^+$. Then, after incorporating this no-prediction case, (4.7) can be written as:

$$E[H_\mathrm{t}] = p^+ p_{m,\mathrm{m}} H_\mathrm{c} + p^+ (1 - p_{m,\mathrm{m}}) H_\mathrm{i} + p^- H_\mathrm{n}. \tag{4.8}$$

Based on that, the optimisation problem can be formulated as follows:

$$\begin{aligned} \min_{p_\mathrm{t}} \quad & E[H_\mathrm{t}] \\ \text{s.t.} \quad & 0 \leq p^+ \leq 1, \\ & 0 \leq p^- \leq 1, \\ & 0 \leq p_{m,\mathrm{m}} \leq 1, \end{aligned} \tag{4.9}$$

where p_t is the probability threshold for making predictions.

Since $\dfrac{d^2(E[H_\mathrm{t}])}{d(p^+)^2} = 0$, the optimisation problem in (4.9) is neither convex nor concave; $E[H_\mathrm{t}]$ is rather a linear function owing to the fact that $p_{m,\mathrm{m}}$ is known and constant. Therefore, the following expression is obtained when the first derivative of $E[H_\mathrm{t}]$ is taken:

$$\frac{d(E[H_\mathrm{t}])}{d(p^+)} = p_{m,\mathrm{m}} (H_\mathrm{c} - H_\mathrm{i}) + H_\mathrm{i} - H_\mathrm{n}. \tag{4.10}$$

[8] The more the number of observations is the better to convergence to the actual accuracy.

From (4.10), it is obvious that $\frac{d(E[H_{\rm t}])}{d(p^{+})} \leq 0$ when $p_{m,{\rm m}} \geq \frac{H_{\rm i} - H_{\rm n}}{H_{\rm i} - H_{\rm c}}$, which yields

$$p_{\rm t} = \frac{H_{\rm i} - H_{\rm n}}{H_{\rm i} - H_{\rm c}}. \tag{4.11}$$

In this regard, in order to make a prediction, the following condition should be satisfied:

$$p_{m,{\rm m}} \geq p_{\rm t} = \frac{H_{\rm i} - H_{\rm n}}{H_{\rm i} - H_{\rm c}}, \tag{4.12}$$

otherwise the conventional non-predictive HO procedure is triggered.

4.1.7 Performance Evaluation

Two different simulation campaigns are carried out in order to reflect the performance of the solutions to the aforementioned two problems. In order to observe the behaviours of the solutions in a better and more detailed way, they are implemented independently and results are analysed accordingly. However, there are some generic properties of the created simulation environment: a network with 19 cells is simulated and the movements of a single user over a period of 100 days, with 10 HOs per day, are observed.

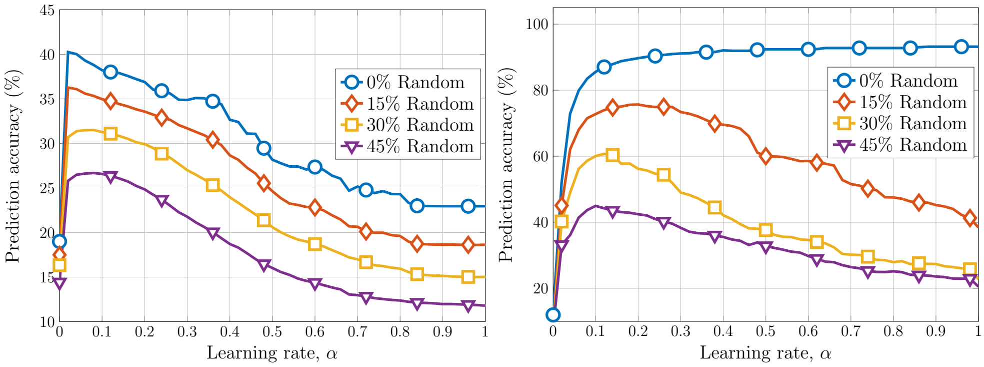
Moreover, four scenarios with different route randomness, mainly: 0%, 15%, 30%, and 45% randomness, are also investigated. More specifically, as in [133], in order to generate the synthetic data set for the user, the following data types are generated:

- a deterministic path is defined for the user in order to reflect their daily routine;

- random paths are generated with the aforementioned randomness levels in order to reflect the random behaviours[9].

Then, these two path types are combined to obtain the historic trajectory data of the considered user. The more randomness the user has in the overall data, the more unpredictable they become: 0% means that the user always takes the predefined deterministic path without any exception, while the user takes some (almost half) random paths in the case of 45% randomness. The performances of the proposed methods are compared with DTMC [133], in which a classical 2-D transition matrix is employed without having any threshold.

[9] The random behaviour can be anything unpredictable or unexpected. For example, a user mostly spends their time during weekends at home, but on a special occasion, they go to a music concert at one of the weekends. This behaviour is obviously unexpected, thereby unpredictable unless using other means of context.



(a) 2-D transition matrix (DTMC).

(b) Proposed 3-D transition matrix.

Figure 4.2: Prediction accuracy performances of DTMC (with 2-D transition matrix) and proposed method (with 3-D transition matrix) across different learning rates. Note that the results are the averages of 30 repeats.

Results for 3-D Transition Matrix based Markov Chains

Two different cases are considered in order to analyse the effects of revisits, namely with 40% revisits and without revisits. Moreover, four scenarios with different route randomness are also investigated, mainly: 0%, 15%, 30%, and 45% randomness. The prediction accuracy—the ratio between the number of correct predictions and total number of predictions—of the two proposed methods are evaluated against the aforementioned learning rate, α. Furthermore, after choosing the most appropriate α for each scenario, the two methods are compared in terms of HO signalling cost, defined as the number of bytes exchanged during HO control messages, according to [133].

The prediction accuracy performances of both DTMC and the proposed 3-D transition matrix are shown in Fig. 4.2, in which Fig. 4.2a is for DTMC while Fig. 4.2b reflects the results for the proposed method. From these two figures, it is clear that the performance of the proposed 3-D transition matrix algorithm largely outperforms the conventional 2-D transition matrix based Markov chains. It can be seen that when revisits occur, 2-D Markov chains are not able to make correct predictions, achieving its best performance of only around 42% when no randomness is included in the user's route. On the other hand, when a 3-D matrix is introduced, the performance approaches to 100% in the no randomness scenario.

Another interesting finding is the effect of parameter α in the performance of the system. In general, it can be concluded that a larger α translates to a worse prediction performance, with the sole exception to this rule being the case of the 3-D matrix with no randomness. This occurs because whenever α increases, the matrix remembers only the most recent transitions performed by the user, whereas when α is smaller, previous transitions are taken more into account. As such, whenever there is randomness in the user's route, more prediction errors are found to occur for a larger α. On the other hand, if the route is fixed, a larger α definitely helps, as the matrix is able to converge faster to the learned values. Therefore, it can also be seen that α plays an important role in the system, thereby should be chosen properly.

Based on that, for the evaluation in terms of HO signalling costs, the best values of α for each curve are chosen and results are shown in Fig 4.3, wherein it can be seen that the proposed 3-D matrix achieves a better performance than the conventional 2-D approach in all considered scenarios. This occurs because the 3-D matrix is able to make less mistakes in the mobility prediction, as such, it is able to allocate resources in advance at the correct BS, minimising the HO signalling costs. On the other hand, the conventional 2-D Markov method, as seen from Fig. 4.2a, is not able to correctly predict the next cell of a user, thus, HO signalling is sent to the wrong BS, increasing its total cost. In addition, it can also be seen that, due to the large amount of errors, the performance of the incorrect prediction case can be worse than the no-prediction scenario. Hence, the impact of correct and incorrect predictions plays an important role in the signalling exchange in mobility management.

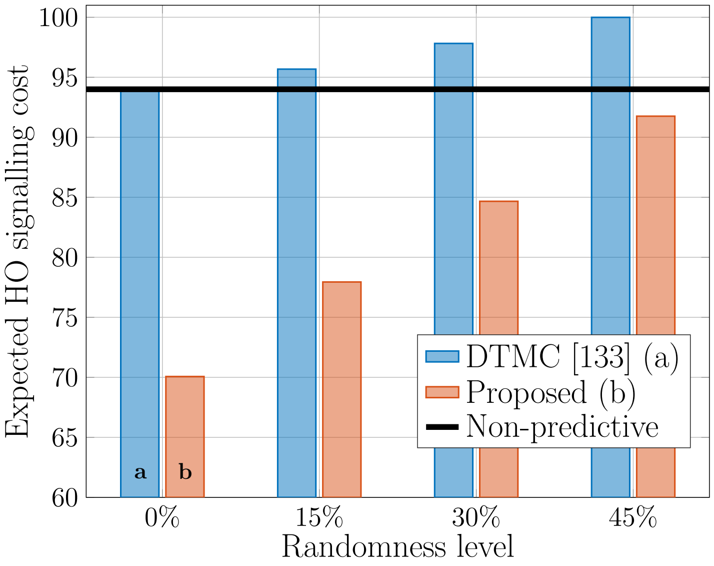


Figure 4.3: Expected HO signalling cost performance of DTMC (with 2-D transition matrix) and proposed method (with 3-D transition matrix) for different levels of trajectory randomness. The HO signalling costs for both methods are obtained from their best performing learning rates for each randomness level. Note that the results are the averages of 30 repeats. The indexing from (a) to (b) in the legend and the first set of bars are done for identification, and the same order follows for all sets of the bars.

Results on Threshold based Markov Chains

The prediction performances of both DTMC [133] and the proposed threshold based Markov chains are presented in Fig. 4.4, in which the prediction accuracy is used as a performance metric. Note that the total number of predictions is equal to the total number of HOs for DTMC, however, for the proposed method, the total numbers of HOs and predictions do not necessarily equal to each other owing to its no-prediction option.

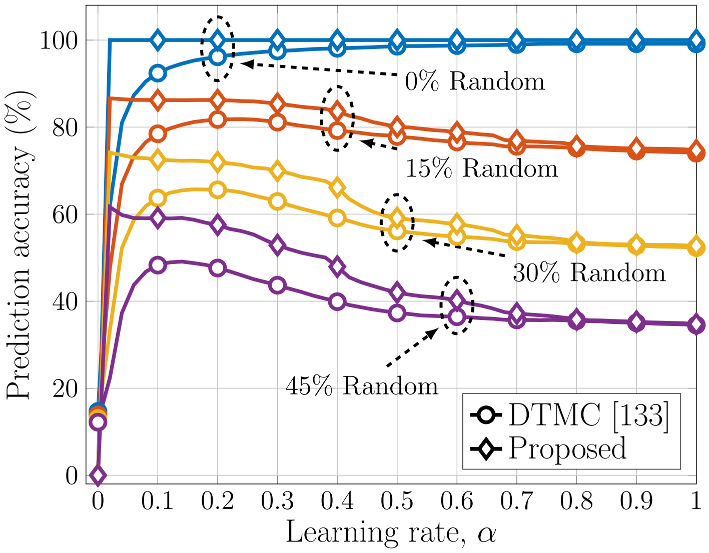


Figure 4.4: Prediction accuracy performances of the proposed threshold based method and DTMC across learning rates and various randomness in the trajectory. Note that the learning rate, α, is equivalent to the the trajectory dependent parameter, R_d in [133], which is used as a reference work.

The results in Fig. 4.4 reveal that the proposed method outperformed DTMC with various scales, and there are three important takeaways from these results:

- **prediction accuracy versus randomness:** for both methods, the prediction accuracy tends to decay with increasing randomness in the user trajectory. This observation is quite expected, as randomness cannot be predicted, and more randomness in the data set makes it harder to predict. Therefore, when the randomness in the data set is 0%, meaning the user always follows the pre-defined deterministic path, the prediction accuracy is equal (for the proposed method) or very close (for DTMC) to 100% for certain α values. For 45% randomness, on the other hand, the prediction accuracy drops to around 28% when $\alpha = 1$ for both DTMC and the proposed method. From this trend, it is expected that the prediction accuracy would drop further when the randomness in the user trajectory is increased.

- **prediction accuracy versus learning rate:** except for the 0% randomness case, a similar pattern is observed when the prediction accuracy is evaluated across various α values: to some extent of learning rate, the prediction accuracy improves with increasing learning rate, but then it changes the behaviour and starts decaying with increasing learning rate. The underlying rationale behind it is that, for example, when $\alpha = 0$, the transition matrix is not updated and kept the same as it was initialised. Given that the uniform distribution is followed in the initialisation phase, such that all the neighbouring cells have the same transition probability:

  - DTMC makes prediction randomly, and that's why it results in very low prediction accuracies when $\alpha = 0$. Due to the nature of the uniform distribution, this obtained accuracy would drop for larger number of neighbours, or vice versa.
  - the proposed method does not make predictions due to the fact that the probabilities in the transition matrix are very low; i.e., not satisfying the developed threshold. This is the reason why the proposed method starts from 0% accuracy when $\alpha = 0$, while DTMC starts from 13%-15%.

  The prediction accuracies start rising once α increases from 0 to some small positive numbers, because the algorithms starts learning from the experience they gained. Nonetheless, although the algorithms keep a good level of prediction accuracies to some level of learning rate (around $\alpha = 0.2$), the performances begin to degrade once the learning rate further increases ($\approx \alpha > 0.2$). This is because the algorithms start valuing the new data more, which in turn limits their learning capabilities. Furthermore, the degree of the performance degradation with the learning rate is also related to the randomness level: for DTMC, for instance, the prediction accuracy decreased by 8% for 45% randomness when α is incremented from 0.3 to 0.4, whereas the decrease is only 2.31% for 15% randomness. The reasoning behind this finding is that the algorithms value new data more for increasing learning rate, and the new data is more likely to be random for a higher degrees of randomness, which subsequently damages the prediction accuracy.

  Based on that, the prediction performances of both methods for the 0% randomness level is also worth discussing. As seen from the findings, the 0% randomness case does not follow the common trend as the rest of the cases. As such, unlike the other randomness levels, the performance always increases with the learning rate for DTMC, since it is better to learn fast;

i.e., value new data more, for the case of no randomness in the trajectory. This is because the new value is more accurate than the existing experience, which is based on the uniform initialisation. In other words, the algorithm tries to converge to the correct values starting from the initial transition matrix, which is more likely to be inaccurate, thereby the lower learning rates is more prone to take more time to converge, making them susceptible to incorrect prediction due to prolonged learning phase. For the proposed method, it seems it hits 100% prediction accuracy from the very beginning (when $\alpha = 0.02$), and its performance is no longer affected. However, this statement is partially true, such that for increased learning rates the proposed algorithm makes more correct predictions. Thus, even though the prediction accuracy remains the same, the number of correct predictions rises with increasing learning rate.

- **performance difference versus randomness:** although similar patterns are observed for both the proposed method and DTMC, the performance difference between them vary for different randomness levels. For example, when their best performing learning rates are compared[10], the proposed method outperformed DTMC by 8.9% and 0.99% for 45% and 0% randomness, respectively. The underlying idea for this observation is that the proposed method becomes more significant for users with higher randomness levels, as they are harder to predict and the proposed method protects the algorithm from making incorrect predictions.

The HO signalling cost performances of the proposed method and DTMC for various learning rates are demonstrated in Fig. 4.5, where the calculation of the total HO signalling cost is performed as follows:

$$H_{\mathrm{t}} = N_{\mathrm{ho,c}} H_{\mathrm{c}} + N_{\mathrm{ho,i}} H_{\mathrm{i}} + N_{\mathrm{ho,n}} H_{\mathrm{n}}, \tag{4.13}$$

where $N_{\mathrm{ho,c}}$ and $N_{\mathrm{ho,i}}$ are the total number of HOs that are correctly and incorrectly predicted, while $N_{\mathrm{ho,n}}$ is the number of non-predictive HOs (i.e., conventional HO). For the proposed method: $\{N_{\mathrm{ho,c}}, N_{\mathrm{ho,i}}, N_{\mathrm{ho,n}}\} \geq 0$, while for DTMC: $\{N_{\mathrm{ho,c}}, N_{\mathrm{ho,i}}\} \geq 0$ and $N_{\mathrm{ho,n}} = 0$. For the non-predictive case, on the other hand, $N_{\mathrm{ho,n}} = N_{\mathrm{ho,t}}$, where $N_{\mathrm{ho,t}}$ is the total number of HOs that yields $\{N_{\mathrm{ho,c}}, N_{\mathrm{ho,i}}\} = 0$.

As seen from the results, except for some instances in 45% randomness, both methods performed better than the conventional HO mechanism. Besides, the

[10] The best performing; i.e., the one with the highest prediction accuracy, is selected for comparisons for each randomness level.

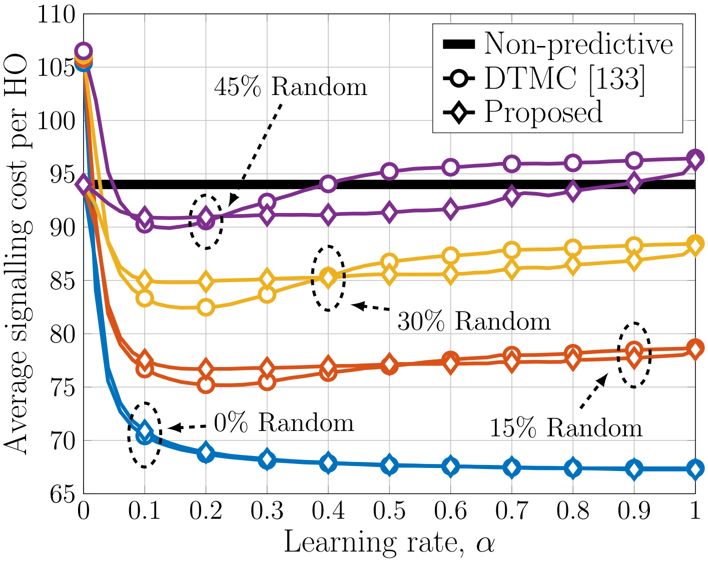


Figure 4.5: Performance of the proposed method and DTMC in terms of HO signalling cost across different learning rates. The average signalling cost per HO is calculated by dividing the total obtained HO signalling cost by the total number of HOs. Non-predictive case reflects the conventional HO procedure that does not include any prediction.

trends against the randomness level and the learning rate are quite similar to the findings in Fig. 4.4, because the prediction accuracy and the HO signalling costs are strongly correlated to each other. The model provided in (4.13) includes the number of correct and incorrect HOs, from which the prediction accuracy is derived, thereby (4.13) confirms the aforementioned relationship between the prediction accuracy and HO signalling cost.

The HO signalling cost grows with increasing randomness level in the data given that the less randomness in the HO history of the user results in more prediction accuracy. However, unlike the prediction accuracy results given in Fig. 4.4, for 0% randomness, the HO signalling cost almost continuously reduces with increasing learning rate albeit being slightly. Still considering the 0% randomness level, the proposed method makes more correct predictions with increasing learning rate that did not have an impact on the prediction accuracy—the ratio between correctly predicted HO and total predicted HOs. However, from (4.13), the total HO signalling costs decays with increasing number of correctly predicted HOs, since $H_\mathrm{c} < H_\mathrm{n} < H_\mathrm{i}$.

4.1.8 Summary

Two different problems associated with conventional Markov chains based HO predictors were identified and corresponding solutions were developed in this work. The identified problems are diagnosed to arise from the limited contextual information of conventional Markov chains based predictors, such that the algorithm is built solely on the HO history of users. Even though this kind of HO predictors are preferable in some cases due to their simplicity, the scenarios that they can work is also limited, making them impractical for some other cases. First, it is found in this study that when revisits are included in the user path, the performances of predictors degrade significantly due to the fact that revisits make multiple destination states have similar—or even the same—transition probabilities. This, in turn, affects the accuracy of the HO predictions. Therefore, in this work, a 3-D transition matrix based Markov chains HO predictor was designed, where the orders of the HOs also matter. In other words, the orders of the HOs are treated as additional contextual information, helping the algorithm make more informed and accurate predictions, and the results suggested that the proposed 3-D transition matrix structure enhanced the predictor's immunity to the revisits.

Second, the criterion of holding the highest probability to be predicted as a next state was identified as another problem, especially for users with large number of HOs—each constitute a state—within a day. When the size of the state space increases, it is more likely that the transition probabilities get smaller due to the nature of the random process. As such, the conventional predictors perform their prediction regardless of how confident they are: they just check for the state with the highest transition probability. This would increase the chances of making inaccurate HO predictions, which is more costly than the non-predictive HO process in terms of HO signalling. Therefore, it is vital to minimise the inaccuracies in the predictions, but the single criterion process is vulnerable to possible errors. In this regard, a threshold based HO prediction was introduced in this work, where the HO prediction is only triggered when the confidence of the predictor is sufficient; i.e., meet the threshold. An optimisation problem was developed and the threshold was analytically derived, and the numerical simulation results demonstrated that the HO signalling cost can be reduced with such a thresholding mechanism.

4.2 Clustering Based UAV Positioning for Capacity Enhancement

4.2.1 Introduction

UAVs are expected to play a fundamental role in future mobile networks, due to their mobility and adaptability, which allow them to provide network services on demand [94, 176]. Thus, UAVs are envisioned to be deployed as aerial BSs in order to provide a wide range of services in several situations, such as in quickly restoring service in emergencies, providing connectivity to remote areas, cache in the air and capacity enhancement, to name a few [94, 176, 265–267].

In particular, in the realm of capacity enhancement, UAV BSs can be utilised as complementary solutions to ground networks, when temporary or big events happen, such as open markets, fairs or music concerts. In such cases, the capacity offered by the ground network might be insufficient, causing many users to be in outage. Thus, UAVs can be a vital solution to this mobile networks use-case. The advantages of using UAV-mounted BSs over conventional terrestrial fixed BSs are already mentioned in Section 2.4.3, however, it is also worth discussing the benefits of UAVs against terrestrial mobile BSs mounted on vehicles, such as cars, trucks, etc. Although the mobility and on-demand availability are keys for both cases, UAVs come with extra benefits [176–178, 268]: first, the mobility of UAVs are easier and more flexible compared to terrestrial vehicles, which need to follow existing roads, traffic regulations, etc. UAVs, on the other hand, can move more freely, since they are subject to less restrictions. Second, because UAVs are placed in the air, they are most likely to construct LOS links with users, which in turn enhances the communication performance by improving the received signal quality. Moreover, UAVs are able to alter their altitudes, thereby they become capable of adjusting their coverage and link according to changing circumstances. Third, in the cases where the users are mostly mobile, the positioning of the mobile BSs should be continually optimised, and terrestrial mobile BSs are disadvantageous on this owing to their aforementioned movement restrictions. Fourth, UAV BS deployments are more cost-effective than dedicating terrestrial vehicles for BS deployments. Lastly, the deployment of terrestrial mobile BSs would be very hard in disaster scenarios, where the existing infrastructure, including communication networks and roads, are destroyed, and thus arriving at the required locations is less likely to be feasible for terrestrial BSs.

However, despite the recent popularity of UAVs and all of the potential applications, integrating UAVs in mobile networks is still a challenging topic [176].

In this context, several issues still remain, such as determining the optimal deployment location of multiple UAVs, designing their optimal trajectory, how to minimise the interference between the aerial and ground networks, how to handle resource allocation, HO and the backhaul of UAVs, etc. [176, 269]. As such, in this work, a low complexity solution is proposed to determine the optimal 3-D placement of multiple UAVs in a scenario of network capacity enhancement, such as an event happening in an urban area, in what is known as a pop-up network. In order to tackle this problem, a two-step solution is proposed, in which the 3-D placement problem is divided into two parts. First, the utilisation of k-means clustering is proposed to find the optimal 2-D placement of multiple UAVs. After that, the optimal heights of the aerial BSs are found such that each UAV can cover all users in its cluster. In addition, due to the intelligent and online nature of the proposed solution, it is applicable in different scenarios, such as in emergency networks, since this approach relies only on data, rather than problem specific constraints.

In the context of pop-up networks, a simulation scenario is built, in which an urban area is considered and user traffic demands are generated. In order to simulate real traffic conditions, the real CDR data provided by Telecom Italia, for the city of Milan, Italy is used as the data traffic[11]. It is assumed that the original ground network can cover part of that, but the additional capacity needs to be provided by the UAVs. The proposed solution is compared with two baseline methods, which consist of deploying the UAVs in a symmetric and in a uniformly random manner around the grid. Results show that the proposed method is more robust and capable of performing an online optimisation of the 3-D position of multiple UAVs and that it outperforms the other baselines in terms of users covered.

4.2.2 Related Work

In terms of UAV positioning, several recent works have tackled the issue. For example, in [267], the authors optimise the number and position of multiple UAV BSs in the presence of a ground network in order to achieve a particular QoS target. In contrast, in [270], the authors develop a mathematical solution to find the optimal position of a single UAV in order to minimise its energy consumption. Sun et al., in [233] propose two different methods to position multiple UAV BSs and achieve user coverage maximisation, one based on a mathematical approach and another based on k-means. However, the solutions proposed assume a very

[11] See Section 5.5 in Chapter 3 for the details about the dataset, which is available online at https://dandelion.eu/datamine/open-big-data/.

simple network scenario and also that the UAVs cannot have overlapping coverage regions, in order to limit inter-UAV interference. On the other hand, other approaches utilise ML methods to position multiple UAVs. In [271], for example, the authors utilise RL to position a single UAV in a scenario where the ground network is operable in order to enhance the QoS of the network. Similarly, [94] also proposes an RL approach to solve the positioning problem of UAV BSs. This time, however, a multiple UAV solution is proposed and the authors consider an emergency scenario, in which the previous ground network was totally destroyed. Lastly, [272] proposes a Gaussian mixture model to determine the optimal deployment of multiple UAVs considering a minimum power consumption.

4.2.3 Objectives and Contributions

Although there are numerous positioning methods available in the literature, solutions, such as [233, 267, 270] rely on a mathematical approach, which, in general, work in an offline manner—and can be quite limiting [52]. On the other hand, other approaches utilise ML methods, such as [94,271,272], however, these solutions require a lot of complexity and memory, which hinders the effect of these solutions.

In this work, the objective is to design a low-cost—in terms of computation—algorithm that can accurately position the UAV BSs. To this end, a clustering based positioning algorithm is proposed, where the 2-D (latitude and longitude) of the UAVs are determined. In particular, 2-D UAV positioning problem is converted to the clustering problem due to their similarities. Thus, k-means clustering—one of the most common and promising clustering algorithms—can be applied to determine the 2-D positions of the UAVs. More specifically, the objective of positioning is to deploy the UAVs in a way that they can have a close proximity to the users, since the closer users are to BSs, the larger the received SINR they might get due to the less path-loss and increased possibility of LOS. For k-means algorithm, on the other hand, the objective is to locate centroids for each cluster in a way that each data point is closer to its associated centroid than the other ones. Therefore, k-means algorithm is selected as a methodology to place the UAVs in terms of latitude and longitude. The use of an ML algorithm has its inherent advantageous; for example, the solution does not need to be trained/implemented from scratch as it becomes adaptive to the changes in the environment. Furthermore, an analytical approach is followed for the altitude, where the UAVs are situated at an altitude that they can cover all their associated users, which was determined via k-means implementation.

Moreover, the effects of the altitude on the received SINR is analysed compre-

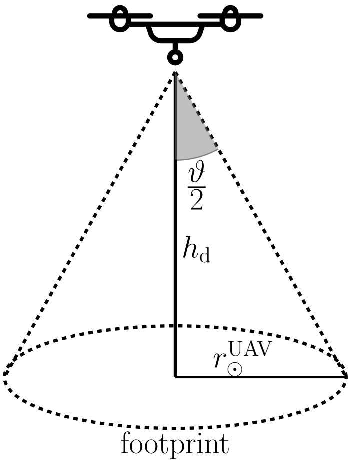


Figure 4.6: Directivity angle of the SC carried by the UAV [94].

hensively. This is an important contribution, since the selection of altitude would result in overlapping footprints for two or more UAV BSs, which in turn inflates the interference observed by the users within the overlapping regions. This subsequently degrades the user throughput, and it violates the idea behind UAV BS deployments given that the objective is to improve the capacity of the network. Therefore, this work not only provides a low-cost ML based solution to the UAV positioning problem, it also—in case of multiple UAV deployments—discusses the impacts of the altitude from the throughput/SINR perspective.

4.2.4 System Model

Environment

In this work, an urban scenario is considered with a fixed terrestrial infrastructure which provides cellular connectivity. Due to an increased demand in capacity, the network is strained and therefore the QoS experienced by its users is degraded. In this regard, a strategy is proposed for mitigating this degradation by deploying UAVs equipped with SCs, such that additional capacity can be offered.

UAV Small Cell

Radio access network: The UAV is carrying a SC with a directional antenna, and therefore it has a coverage footprint [94]. This is illustrated in Fig. 4.6, where ϑ is the directivity angle, h_{d} is the altitude of the UAV and $r_{\odot}^{\mathrm{UAV}}$ is the radius of coverage.

The SINR at the user is obtained using the same model as in [273], which considers free space path-loss between the user and the UAV. Following [273],

$PL_{i,j}$, the path-loss in dB, for the link between user i and UAV j is obtained as [94]

$$PL_{i,j} = 20\log_{10}\left(\frac{4\pi f_{\mathrm{c}} d_{i,j}}{c}\right), \tag{4.14}$$

where f_{c} is the carrier frequency, $d_{i,j}$ is the link distance, and c is the speed of light. Note that fading is not considered in the propagation modelling.

Next, the SINR for the link in question, $\gamma_{i,j}$, is obtained via [94]

$$\gamma_{i,j} = \frac{P_{i,j}}{\mathcal{N}_0 + \sum\limits_{k=1,k\neq j}^{N_{\mathrm{UAV}}} P_{i,k}}, \tag{4.15}$$

where N_{UAV} is the number of UAVs deployed, $\mathcal{N}_0$ is the additive white Gaussian noise (AWGN) power and

$$\mathcal{P}_{i,j} = \mathcal{P}_{\mathrm{t},j} - PL_{i,j} \tag{4.16}$$

is the received signal strength at user i, where $\mathcal{P}_{\mathrm{t},j}$ is the transmit power of UAV j. Note that all the powers are measured in dB.

Furthermore, the throughput is measured using Shannon's capacity formula, as in [94], such that the throughput for the link in question is determined via [146]

$$\mathfrak{T}_{i,j} = W\log_2(1+\gamma_{i,j}), \tag{4.17}$$

where W is the communication bandwidth.

Moreover, since the BS carried by the UAV is compliant with the cellular network standard, it utilises orthogonal frequency-division multiplexing (OFDM) for scheduling RAN resources, and therefore it has a limited number of resource blocks (RBs) to provide connectivity to users.

Backhaul: The backhaul connectivity for the UAV SCs is essential for the proper operation of the proposed solution to function properly. With that in mind, it is proposed that this should be done via a microwave link between the UAVs and the terrestrial BSs. In order to keep the interference to a minimum, this connection would be in the form of an out of band backhaul [94], thus requiring an additional spectrum, and leveraging OFDM to avoid interference between the connectivity of multiple UAVs.

User Allocation

User allocation is performed by the users ranking the available BSs by received SINR and choosing the one which provides the highest value. A UE is allocated to its highest ranked BS that has available RAN resources and which can provide a minimum signal strength criteria[12]. If there is no BS which meets the criteria, the user is not allocated and is considered out of coverage (i.e., in outage).

4.2.5 Proposed Solution

With this scenario in mind, a strategy is proposed to find a position to deploy the UAVs in order to provide the necessary enhanced capacity in the crowded scenario.

Firstly, N_{UAV}, the number of UAVs to be deployed, must be determined. It is proposed to find it by computing the necessary increase in capacity and providing that capacity with the UAVs, such that

$$N_{\mathrm{UAV}} = \frac{C_{\mathrm{D}} - C_{\mathrm{E}}}{C_{\mathrm{UAV}}}, \tag{4.18}$$

where C_{D} is the demanded capacity by the users, C_{E} is the existing network capacity, and C_{UAV} is the capacity that each UAV can provide, in terms of RAN. Note that C_{UAV} is the full capacity of a UAV, meaning that it is the capacity before undesired effects, such as interference, take place. However, intuitively, with the inclusion of interference, C_{UAV} is expected to decrease, which subsequently increases N_{UAV} according to (4.18). In this work, on the other hand, the planning on the number of required UAVs is performed by considering the full capacity of the UAVs, since the amount of interference cannot be estimated in advance.

Next, the solution consists of i) finding the best (x, y) position[13] to deploy the UAVs using unsupervised learning; then ii) determining the altitude for the UAVs that provides the best QoS both in terms of throughput as well as number of users served. The first task is accomplished by first obtaining the position of the users using a localisation technique, such as the one presented in [274], and then performing k-means clustering using the users' (x, y) position as features in order to determine the cluster centres.

Then, the UAVs are positioned at the cluster centres and their altitude is determined such that QoS metrics are optimised. From Fig. 4.6, $r_{\odot}^{\mathrm{eNB}}$, the footprint coverage radius, can easily be determined as a function of the flight altitude and

[12] In the simulations, it is considered that this minimum signal strength is 3 dB below the required by the user.

[13] x, y: latitude, longitude.

the antenna directivity, such that

$$r_{\odot}^{\mathrm{eNB}} = h_{\mathrm{d}} \tan\left(\frac{\vartheta}{2}\right). \tag{4.19}$$

Due to the nature of the UAV BS with the well-defined footprint, it is possible to regulate the UAV altitude depending on the desired area to be served. Moreover, due to the expensive nature of spectrum licenses, it is assumed that all the UAVs share the same spectrum, thus causing interference on each other. Thus, interference and coverage area create a trade-off in terms of altitude. In other words, the higher a UAV is positioned the larger the coverage footprint is, however at the same time it causes more interference on neighbouring UAVs. This can be viewed as an overlap in coverage footprints.

In order to study this trade-off effectively taking into account the user distribution, $\varrho \in [0, 1]$, a parameter that can regulate the amount of overlapping footprint, is introduced. Therefore, the flight altitude of the UAV is obtained as a function of ϱ and can be tuned online according to the QoS performance, such that

$$h_{\mathrm{d}} = 2\varrho r_{\odot,\mathrm{x}}^{\mathrm{eNB}} / \tan\left(\frac{\vartheta}{2}\right), \tag{4.20}$$

where $r_{\odot,\mathrm{x}}^{\mathrm{eNB}}$ is a radius that depends on the adopted strategy. For the proposed solution, it is equal to the distance of the furthest user in the cluster to the cluster centre. When (4.19) considered together with (4.20), it can be seen that ϱ is a parameter that is used to control the altitude of a UAV, which in turn affects the footprint of the UAV SC, such that: because (4.20) and (4.19) yield $\varrho \propto h_{\mathrm{d}}$ and $h_{\mathrm{d}} \propto r_{\odot}^{\mathrm{eNB}}$, respectively, then it can be inferred that $\varrho \propto r_{\odot}^{\mathrm{eNB}}$. In other words, even though ϱ is a control parameter for the altitude, it has a direct impact on the footprint of the UAV BS owing to the relationship between the altitude and footprint, as can be seen in (4.19).

From (4.20) it is possible to observe that, when $\varrho = 0.5$, all the users in each cluster are in the coverage range of an UAV positioned at the cluster centre, while when ϱ increases the UAVs serve larger areas, possibly serving more users (and increasing the interference into neighbouring UAVs), and lastly, when ϱ decreases, the UAVs serve smaller areas, but interfere less amongst themselves.

4.2.6 Simulation Results

Benchmarking and Metrics

In order to compare the performance of the proposed solution, two different benchmark UAV deployment methods are developed. First, a *symmetric* deployment is developed, where the UAVs are deployed symmetrically such that the differences between the consecutive UAVs are the same on both x and y axes. The altitude of the UAVs are calculated through (4.20), by setting $r^{\mathrm{eNB}}_{\odot,\mathrm{x}}$ as the largest radius which does not result in any overlap between the footprints of neighbouring UAVs. Moreover, due to the symmetric nature of the deployment, the altitude is kept the same for each UAV.

Second, a *random* deployment is developed, where the UAVs are uniformly distributed across the region of interest. There is no regular pattern for the distance between the UAVs, and thus the proximity of the UAVs can be small for some, while others are located far away from each other. Similar to the symmetric distribution case, the altitude of the UAVs are determined via (4.20). Here, the same $r^{\mathrm{eNB}}_{\odot,\mathrm{x}}$ value that was calculated for the symmetric distribution is adopted, since any reference distance value is not available for the random deployment method.

To evaluate and compare the performance of the developed methods, two different metrics are introduced. First, the user perceived SINR values—calculated using (4.15)—are measured. This becomes a utilitarian metric demonstrating the signal quality that is received by the users, which subsequently affects the throughput as evaluated in (4.17). Given the stringent peak data rate requirements for the eMBB scenario in 5G NR [13], improving user experienced throughput is a vital task.

Second, the number of user in outage is also counted in order to investigate the link failure performance of the developed methods. In this regard, a certain threshold value, γ_{T}, is selected for SINR values, such that the users are counted as in outage if their received SINR values are below γ_{T}, while they are treated as covered when the SINR is at least equal to γ_{T}. Further, the percentage of the users that are in outage is also calculated using $\dot{N}_{\mathrm{u,out}} = \dfrac{N_{\mathrm{u,out}}}{N_{\mathrm{u,t}}}$, where $N_{\mathrm{u,out}}$ and $N_{\mathrm{u,t}}$ are the number of users in outage and total number of users in the region of interest, respectively.

Simulated Environment

Data Set: As also explained in Chapter 3, the CDR data set is provided by Telecom Italia for the city of Milan, Italy[14]. As such, the city of Milan was divided into $10,000$ square-shaped grids, in which each grid has a side of 235 m. Then, call, text message, and internet activity levels, which reflect the amount of user-network interactions, were logged for each grid for 2 months. Furthermore, the resolution of the data set is 10 minutes, meaning that the activity levels were aggregated into 10-minute time slots. However, the provided data is unitless, and thus reflects merely a relative user activity levels.

User Positioning: Since the data set itself does not provide much information, a pre-processing and further assumptions are needed to make it more meaningful. In this regard, first, the user activity level in a grid is considered as the throughput demand from the users located in that grid. After that, the call, text message, and internet activities are combined in order to estimate a total throughput demand from each grid. As the data set is from the year 2013, in order to reflect the increase in data demand since then, the total throughput demands are then multiplied with a coefficient, ν. Later, it is assumed that the existing ground network also supplies for certain amount of the demand per grid, $Z_{\mathrm{e,g}}$, before the UAV deployments, which is deducted from the overall throughput demand in order to find out the users that are out of the coverage with the existing network. A certain throughput demand per user, Z_{u}, is assumed so that grid-wise number of users are obtained by dividing the total throughput demand in a grid by Z_{u}. Lastly, the resulting number of users are distributed uniformly across a given grid.

Results

Numerical simulation campaigns are performed to evaluate the proposed approach using parameters from Table 4.1. Moreover, Fig. 4.7 shows a snapshot of the obtained positions when $\varrho = 1$.

Fig. 4.8 demonstrates the perceived user SINR performances of the developed methods when $\varrho = 0.1$. The first point that can be inferred from these results is that the symmetric and random deployments performed very close to each other. The rationale behind this is that the altitude of the UAVs are comparatively less when $\varrho = 0.1$, so are the coverage areas. Therefore, there is a very small room for interference to be effective, resulting in a kind of interference-free communi-

[14] Data is available online at https://dandelion.eu/datamine/open-big-data/.

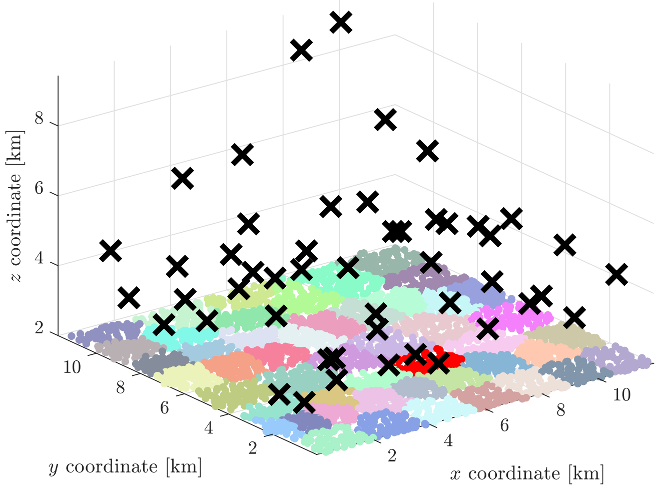


Figure 4.7: Distribution of the UAVs using the proposed methods when $\varrho = 1$. Dots (●) in different colour represent the users in different clusters that the k-means algorithm found, while black crosses (×) represent the UAVs.

Table 4.1: Simulation parameters

Parameters	**Value**
UE height	2 m
UAV SC EIRP	0 dbW
UAV SC antenna directivity angle	60°
Carrier frequency, f_c	1 GHz
Bandwidth per RB	180 kHz
Number of RBs per UAV	100
SINR threshold, γ_T	-3 dB
Data demand increase coefficient, ν	10
Terrestrial throughput supply per grid, $Z_\mathrm{e,g}$	20 Mbps
Number of grids considered	2500
Area of the region of interest	50 × 50 grids
Dimension of each grid	235 m
User throughput demand, Z_u	50 Mbps

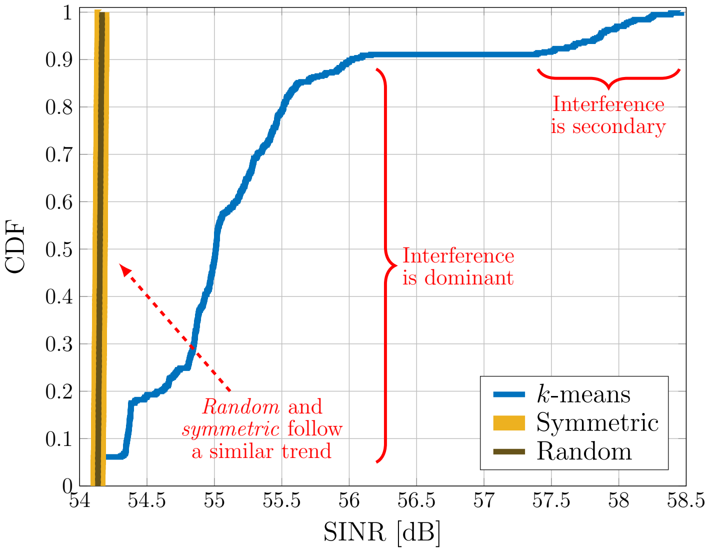


Figure 4.8: Received user SINR performances of the developed methods when $\varrho = 0.1$.

cation for the users. This subsequently makes the path-loss the only dominant factor that affects the link quality between the UAVs and their associated users, and thus there are small variations observed on the received SINR. In other words, considering (4.15), the SINR is dependent on received signal strength and the interference, where $\mathcal{N}_0$ is constant. Hence, when the interference becomes secondary, the only parameter having an effect on the SINR is the received signal strength. From (4.14), it is obvious that the path-loss is merely distance-dependent, since the other parameters, such as f_c, are kept constant. Having said all these, it is quite intuitive that when d gets smaller with decreasing ϱ, the interference becomes secondary and the path-loss causes only small variations.

On the other hand, compared to the symmetric and random distributions, the proposed k-means based UAV distribution method resulted in broader range of SINRs. Two separable regions are observed in this case: a region where the interference is dominant and secondary, respectively. One can question why the dominant interference region occurred in this case while it did not exist for the symmetric and random deployments. The altitudes of the UAVs in the symmetric and random cases are identical, whereas they are different from each other in the proposed deployment method. Accordingly, while there is a clear separation between the footprints of the UAVs in the symmetric and random cases, they are more likely to be overlapping in the proposed method. Therefore, the users, who are located in the overlapping areas, experience interference and constitute the *interference is dominant* region in Fig. 4.8. The users in non-overlapping regions, on the other hand, receive better signal quality and constitute the *interference is*

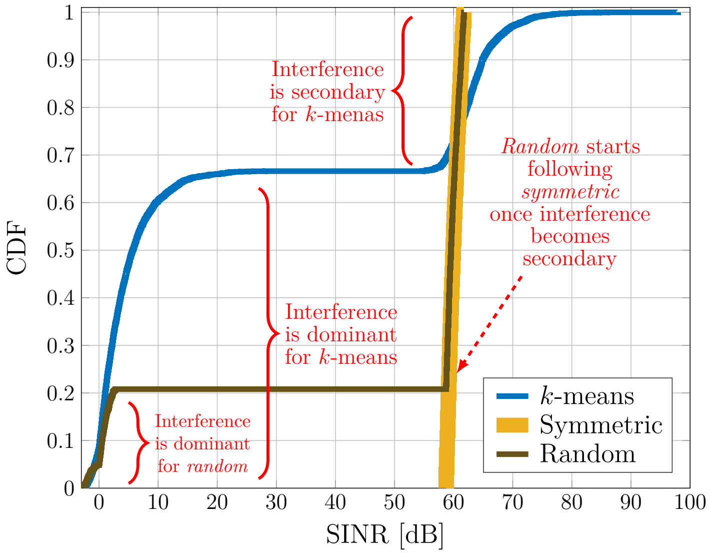


Figure 4.9: Received user SINR performances of the developed methods when $\varrho = 0.5$.

secondary region in Fig. 4.8.

When $\varrho = 0.1$, regardless of the deployment method, the scale of the SINR is quite high (around 54 to 58 dB) as seen in Fig. 4.8, since the interference is less effective and the distance between the UAVs and associated users are comparatively less. Nevertheless, Fig. 4.11 reveals the number of users in outage for three different deployment scenarios, and the outage performances are quite poor for all deployment methods; they resulted in around 87% to 98% of outage. These results are obviously unacceptable owing to the fact that the vast majority of the users are out of service. The under-performance of the developed methods again arises from the lower values of ϱ, which subsequently results in reduced footprints for the UAVs. Moreover, the proposed deployment method outperformed the other two benchmarking methods, because it focuses on minimising the Euclidean distance between the users and UAVs, which makes the UAVs inclined towards the locations where there are more number of users.

The results in Figs. 4.8 and 4.11 can be summarised as follows: when the coverage areas of the UAVs are smaller, so is the interference between them. This also makes the associated users closer to the UAVs. Therefore, the takeaway from these results is that the smaller altitudes of the UAVs—manipulated by ϱ parameter—improves the received SINR values for the user at the expense of connecting much less users. Furthermore, the distinct altitudes of the UAVs for the proposed method render it to be more vulnerable to interference.

Fig. 4.9 demonstrates the SINR performances of the proposed method and two benchmark methods when $\varrho = 0.5$. It is first observed that the received SINR

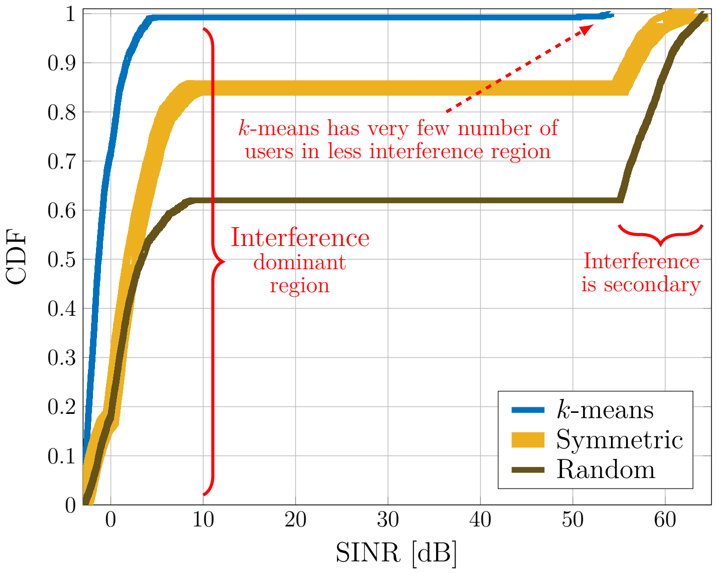


Figure 4.10: Received user SINR performances of the developed methods when $\varrho = 1.0$.

values are scaled down compared to the $\varrho = 0.1$ case. Second, the dominant interference region for the proposed method expanded due to the fact that the footprint for each UAV is enlarged with increasing ϱ value. When the footprint for each single UAV increases, the overlapping areas among UAV footprints also expand, which in turn leaves more users in the dominant interference regions. The symmetric deployment case, on the other hand, is still immune to the interference given that no overlapping region occurs, since $\varrho = 0.5$ means that the footprints of the UAVs are just tangent to each other. For the random case, however, some small portion of the dominant interference region occurs, as the occurrence of overlapping areas is likely due to the uniform distribution, making some users experience considerable level of interference.

As seen in Fig. 4.11, the number of outage users also scaled down compared to the $\varrho = 0.1$ case, since the footprints of the UAVs are now increased, which results in more users to be connected. It is also observed that the proposed method significantly outperformed the symmetric and random distribution methods. As aforementioned, this happens because the proposed method uses k-means algorithm, which employs the Euclidean distance as a cost function to determine the locations of the UAVs. Therefore, the proposed method focuses on reducing the overall Euclidean distance between the UAVs and the users, which positions them closer to a greater number of users.

Lastly, Fig. 4.10 shows the received user SINR results for the three UAV deployment methods while $\varrho = 1$. The region, where interference is dominant, for the proposed method is observed to be expanded dramatically, since around

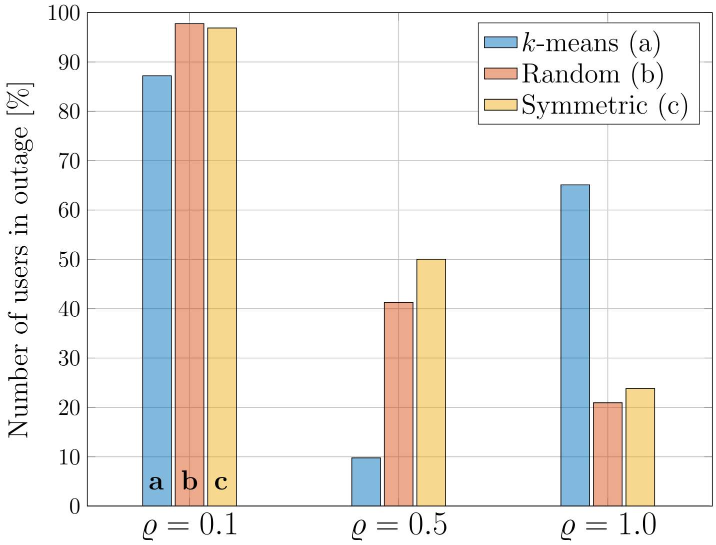


Figure 4.11: Number of users in outage for various α values. The indexing from (a) to (g) in the legend and the first set of bars are done for identification, and the same order follows for all sets of the bars.

99% of the users were found to be in this region. In addition, in this case, the dominant interference region also occurred for both symmetric and random deployment methods. The reason for this, again, is the increased footprints for the UAVs that causes the overlapping areas to become greater, which subsequently increases the number of users experiencing considerable amount of interference. Another interesting point that is worth discussing is that the random deployment method outperformed the proposed and symmetric methods in terms of SINR. This arises from the fact that it results in less overlapping areas at some locations due to the nature of the uniform deployment, whereas the overlapping areas are the same for all the UAVs in the symmetric case. In other words, while the overlapping areas are large for the UAVs located close to each other, it is comparatively less for the UAVs that are separated by a considerable distance.

From the results in Fig. 4.11, the number of users in outage decreased for the symmetric and random deployments compared to the cases where $\varrho = 0.1$ and $\varrho = 0.5$. Once again, the reason for this is the increased footprints of the UAVs, where the UAVs are able to serve more users. Nonetheless, this behaviour did not happen for the proposed method, where interference is much more severe making more users fall below γ_{T}.

4.2.7 Summary

In this work, a k-means algorithm based UAV positioning method was proposed, where (x, y) coordinates of the users were considered as features. Then, the UAVs were deployed at the centroid positions for each cluster. The obtained results reveal that the proposed UAV positioning method is mostly good at reducing the number of users in outage due to the nature of the k-means algorithm, where the Euclidean distance was employed as a cost function. However, in terms of the user perceived SINR values, the proposed method is more vulnerable to interference owing to the differences in the altitudes of the UAVs. ϱ parameters was observed to have crucial impacts of the performances of the developed methods; the greater ϱ is the larger footprint for the UAVs, which in turn increases the number of connected users whereas scaling down the user SINR values. Future works can include the user positioning and limited backhaul capacity conditions, where relay UAVs and/or ground BSs would be needed. Furthermore, regulations on flight altitudes were not considered in this work, but they can play significant roles on the communication performance, since the altitudes of the UAVs are crucial in providing a good coverage and signal quality—as also confirmed with the results obtained in this work. Therefore, future works also include adding the flight regulations as a constraint for the UAV positioning problem.

Chapter 5

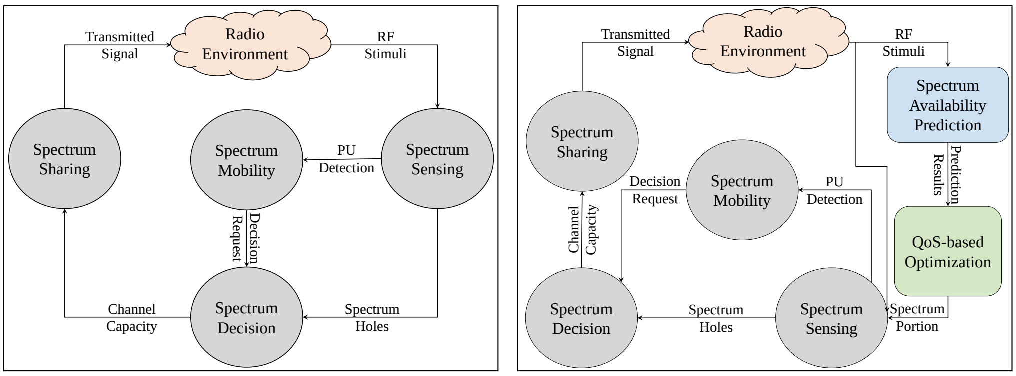
QoS-Aware Dynamic Spectrum Access with Cognitive Radio

A predictive and context-aware methodology is proposed in this Chapter for the concept of dynamic spectrum accessing, as in [275]. In particular, a system model with different types of RATs is considered and the problem formulation is detailed, followed by comprehensive discussions on the ANN and Q-learning assisted proposed methodology. Lastly, the concluding remarks will be provided after evaluating the performance of the proposed approach through numerical simulations.

5.1 Introduction

As shown in Fig. 5.1a [275], four fundamental phases are included in the conventional CR spectrum access process, namely: *spectrum sensing*, *spectrum decision*, *spectrum sharing*, and *spectrum mobility* [184]. In the spectrum sensing phase, SUs sense the spectrum continuously to find an idle channel to allocate, while, in the spectrum decision phase, they choose a channel to associate with (in case of multiple channels being available). Spectrum sharing refers to the process of sharing the available frequency band with other users, and evacuating the allocated channel in the presence of any PU in order to avoid interference is referred to as spectrum mobility [60].

Further, the spectrum sensing phase can be divided into two main categories according to the size of the bandwidth to be sensed, namely NB and WB sensing. The former refers to the case when the bandwidth to be sensed is smaller than the coherence bandwidth of the channel, while the latter happens when the sensed bandwidth is larger than the coherence bandwidth [196]. Many NB spectrum sensing methods, such as energy detector [276], cyclostationarity based [277],



(a) Conventional CR spectrum access process.

(b) Proposed CR spectrum access process.

Figure 5.1: (a) Conventional and (b) proposed CR spectrum access processes. The proposed method puts two additional phases (WB spectrum prediction and QoS-based optimisation) before sensing in order to enhance SU's satisfaction by selecting the network that suits best with the requirements of SU.

matched-filtering [278], etc., have been proposed in the literature [47]. However, the main drawback of NB sensing is that the bandwidth to be sensed is limited, so is the spectral opportunity. As such, NB sensing methods have to compromise on the greater number of available bands by focusing on a certain bandwidth. Furthermore, their implementation to WB is challenging owing to their inherent binary decision approaches [196], thereby dedicated methods have been proposed for WB sensing, including Nyquist based [106, 107, 197] and sub-Nyquist based [109, 110]. Nonetheless, the WB sensing methods are often more complex to implement [196], making them prone to higher latency due to prolonged sensing times [279].

As latency is a key parameter to consider in future mobile networks, especially when SUs run real-time applications, a large amount of sensing attempts can result in SUs being dissatisfied. Towards that end, in order to combat the increased sensing latency issue, predictive spectrum sensing methods have been proposed in the literature [60, 99, 142, 206, 280–287] with an objective of producing an interface between WB and NB sensing by predicting future occupancy states of spectrum bands in WB to enable NB sensing to focus only on the bands that are predicted to be available. However, these prediction-based methods are not a contender of existing sensing techniques, but rather complementary, as the predicted availabilities of spectrum bands are exploited before going through the conventional sensing process to reduce the sensing latency [60]. The fundamental idea behind this approach is that the number of required sensing attempts decays by decreasing the bandwidth of interest, which in turn reduces the resultant latency [275].

Nonetheless, the methods employed in [60,99,142,206,280–287] could be unrealistic, given that they rely on specific assumptions, such as having historic data set on the occupancy of individual channels, that is, impractical due to the fact that acquiring such data set for each individual channel is quite hard. In addition, even if such data is available, this type of implementation would be very costly in terms of both computation and memory, as processing and storage requirements increase with the data volume. Furthermore, most of the aforementioned works take merely the latency as a QoS parameter and ignore other SU requirements during the spectrum sensing phase.

In this study, as shown in Fig. 5.1b, a novel spectrum access approach is proposed, which includes a virtual predictive WB spectrum sensing and *QoS-based optimisation*, with the aim of enacting the satisfaction of SUs by meeting their user/application-specific requirements. In the virtual predictive WB sensing, instead of being interested in individual frequency channels, the traffic loads of the RATs are considered as a whole for predictions, as this approach necessitates significantly less memory and processing due to less amount of data to be handled. Moreover, it is more likely and easier to have historic traffic load data sets for RATs, and thus the proposed virtual predictive WB sensing method makes the process more realistic and practical [275].

Further, in the proposed *QoS-based optimisation* phase, two different decision strategies are introduced. The first strategy, which will be called WB Predictive Sensing (WBPS) hereafter, focuses only on the sensing latency as a QoS parameter. Particularly, the future traffic loads of the RATs, occupying different portions of the WB spectrum, are predicted. Then, the bandwidth of the RAT with the minimum relative traffic load is selected to be sensed with NB sensing, hence the probability of finding a spectrum hole is boosted by narrowing down the WB spectrum to the less utilised portion. In this way, higher spectral opportunity of WB sensing and easy implementation of NB sensing are both exploited without requiring a huge volume of data, reliving the computational and storage burden. In other words, given the aforementioned problems, WBPS is the upgraded and modified version of existing predictive spectrum sensing approaches.

The second strategy, which will be called Q-learning Enabled WBPS (QWBPS) hereafter, is a novel and more robust approach, in which the decision process in WBPS is consolidated by considering all the QoS requirements of SUs. Note that even though any set of QoS requirements can be considered with QWBPS, for the sake of clarification in the presentation, latency, coverage, and bandwidth are captured in this work. Furthermore, the SUs are allowed to prioritise the QoS parameters, rendering the proposed algorithm capable of adapting itself to dynamically changing scenarios and circumstances.

Normally, running a Q-learning algorithm also takes some time, which may lead to an additional latency. However, the proposed framework is proactive and timing is not an issue, since the Q-learning implementation is performed in advance by predicting the future traffic loads. Additionally, as it will be elaborated in Section 5.6.2, since the developed ANN algorithm is fed with time and day inputs, it is able to predict the traffic load not only for the next time slot but also for any given time and day.

However, despite QWBPS being more robust and dynamic, it comes at the expense of more computational cost due to the additional Q-learning implementation. In that regard, a trade-off arises, that is, it is wiser to use WBPS if the latency is the only concern, while QWBPS is a better choice in case the SU has additional concerns other than the latency.

5.2 Related Work

Spectrum prediction has been extensively studied in the literature by employing various techniques [60], including HMM [142,206], ANN [280,284], LSTM [99], autoregressive (AR) model [283], etc. In [280], the authors try to predict the future occupancy states of a channel by designing an MLP with backpropagation (BP). They generate a synthetic PU traffic for a single channel using Poisson process, while the channel's *on/off* times are determined using geometric distribution. It is observed that, through predictive sensing, the spectrum utilisation is boosted and that the sensing energy is decreased. An analytical model for SU's throughput is derived in [281] by considering both the imperfect spectrum prediction and protection to PUs. Some numerical studies are also performed to observe how the SU's throughput is affected by different parameters, such as prediction error and the number of channels to be sensed.

In [285], the performance of a BP ANN for spectrum prediction is improved by employing genetic algorithm at the training phase, as conventional BP ANN is very prone to be trapped into a local optimum [288]. An HMM based prediction of future channel states is presented in [282], wherein a channel selector, which includes a channel state predictor and a channel environment evaluator, is introduced so that the channel selection process becomes the combination of SINR level and the availability of the channel. Another HMM based spectrum prediction is presented in [206], where real data is collected by measuring Wi-Fi signals via the experimental set-up with four different antennas.

An ALOHA system is assumed in [286], in which a second-order AR and

Kalman filter are employed to predict occupancy states of the spectrum. A recent work in [287] study the predictive spectrum management in a comprehensive manner: first, SU's mobility is predicted using a second-order Markov model. Second, the spectrum prediction is also performed and combined with the mobility predictions. Lastly, a channel selection phase is executed in case of multiple channel availability. The authors also include a joint prediction cost model by considering the errors occurring at each stage.

Nevertheless, most of the predictive sensing works available in the literature have been performed for a single (or a few) channel(s) scenario, which is not applicable for WB sensing—albeit being practical for NB sensing—, as it is not possible to predict the future occupancy levels of numerous frequency channels in WB. Furthermore, they mostly rely on the availability of historic channel occupancy data sets for each individual channel, making them even more unrealistic, since it is hard to have such data sets for all channels at all possible locations and times. In addition, in such implementations, the memory and energy consumption increase with the bandwidth of interest (or number of channels), as more data will be needed to conduct ML training for the inflated number of channels.

Besides, as most of the existing studies are performed using synthetically generated occupancy states; i.e. 1 is busy and 0 is free, they are yet to be tested in more complicated and realistic scenarios, where multiple channels are available with different characteristics and limitations. Therefore, there is a significant need for an implementation of predictive spectrum management schemes in more complicated environments. In addition to the lack of realistic implementations, none of the studies aforementioned considered QoS requirements of the SUs before proceeding to the sensing phase.

5.3 Objectives and Contributions

Instead of predicting the occupancy states of individual frequency channels, in this work, traffic load prediction of RATs is proposed for predictive spectrum sensing, since it is easier to acquire/collect such data sets. As such, due to less data requirements, the proposed virtual predictive WB sensing method results in less memory and energy consumption. In addition, the requirements of SUs are also taken into consideration in order to augment their experienced QoS satisfaction. Lastly, prioritisation of QoS parameters are allowed to make the proposed framework more user/application-specific.

The main contributions of this work are as follows:

1. In order to make the model realistic:

- four different RATs with different frequency ranges are considered;
- a real CDR data set[1] is employed for RAT-A and RAT-B;
- the synthetic data generation for RAT-C and RAT-D is inspired by the real data measurements from [289].

2. Given that the data set considered for RATs A and B consists of many squared grids, a k-means algorithm is employed to cluster the grids according to their traffic load characteristics in order to avoid over-fitting and to reduce the computational cost.

3. Due to diversified characteristics of each assumed RAT, different ANN models are developed during future traffic load predictions.

4. Two different decision approaches (WBPS and QWBPS) are proposed to satisfy user-specific requirements. In particular, the WBPS approach is proposed for users with only latency concern, while QWBPS is developed for users who have other QoS requirements in addition to latency. In WBPS, future traffic load predictions are exploited to direct the SUs to the most available RAT for latency reduction purposes, while in QWBPS, future load predictions and QoS requirements—along with their weights—are exploited in order to satisfy the requirements of the SUs.

5. A weighting mechanism for QoS elements is developed to allow the SUs to prioritise their requirements. This enables the users to adjust the importance of their requirements for specific applications, making the proposed method both user and application centric.

5.4 System Model

As shown in Fig. 5.2, the system model considers four RATs around an SU. There is also a CR BS that is responsible to provide coverage and data transmission for the SU. Therefore, the SU in this environment searches for an available frequency band to initiate its connection. With regards to the QoS requirements, latency, coverage, and bandwidth are considered in this study, and they will be detailed individually in the following paragraphs.

[1] See Section 5.5 in Chapter 3 for the details about the dataset, which is available online at https://dandelion.eu/datamine/open-big-data/.

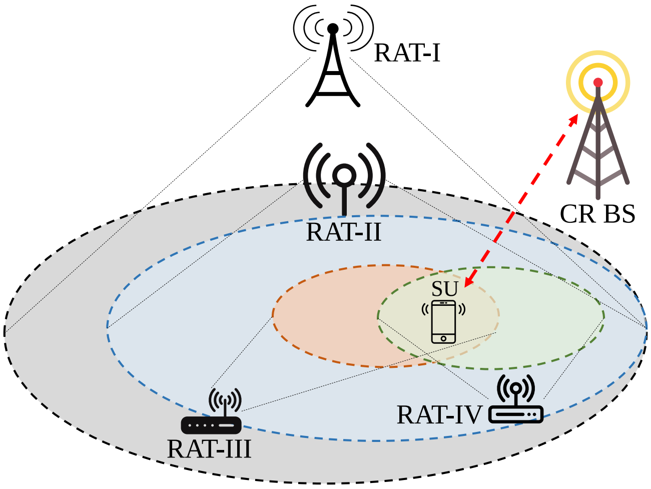


Figure 5.2: System model. An SU is surrounded by four different RATs. RAT-A, RAT-B, RAT-C, and RAT-D use different carrier frequencies.

5.4.1 User Requirements

The coverage requirement refers to the distance that the SU will be away from its current location. If the SU is mobile, for example, then it will need an RAT offering more coverage in order for its connection to last longer. As shown in Fig. 5.2,—in the case of SU with high mobility profile—since RAT-A offers the widest coverage area, it would preferably be selected as an RAT to be sensed.

Second, the bandwidth (or data rate) requirement is captured by how much bandwidth the SU demands to run a desired application. In the case of video-conferencing, for instance, huge data rates are required, hence an RAT with more available bandwidth is preferred.

Theorem 2. *Let x_f be a random variable that represents full-satisfaction, where SU's both coverage and bandwidth requirements are satisfied simultaneously, such that $x_f = 1$ if full-satisfaction is achieved, and $x_f = 0$ otherwise. Then, the expected value of full-satisfaction is*

$$E[x_f] = \sum_{i=1}^{N_R} p_{t,i} p_{s,i}, \tag{5.1}$$

where $p_{t,i}$ is the probability of full-satisfaction with RAT i, $p_{s,i}$ is the probability of selecting RAT i among all the available options, and N_R is the number of available RATs.

Proof. Firstly, a random search concept is developed for benchmarking purposes, where SUs select a random RAT, whose channels in the frequency spectrum are sensed in a random manner in order to find an available one to occupy. Therefore, the RAT selection is a random process following the discrete uniform distribution. Let $\mathbb{O}$ be the set of the available RAT options, such that $\mathbb{O} = \{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_{N_\mathrm{R}}\}$, whose index set is $\mathbb{I} = \{1, 2, ..., N_\mathrm{R}\}$.

Let $n \sim U(1, N_\mathrm{R})$ be a random variable used to select the index of RAT to be sensed from $\mathbb{I}$. Moreover, let p_s be the probability of being selected as an RAT to be sensed:

$$p_\mathrm{s} = \frac{1}{N_\mathrm{R}}, \tag{5.2}$$

where $N_\mathrm{R} = |\mathbb{I}|$ is the cardinality of $\mathbb{I}$.

SU's QoS requirements (Σ) are given as:

$$\Sigma = (\Sigma_\mathrm{c}, \Sigma_\mathrm{b}), \quad \{\Sigma_\mathrm{c}, \Sigma_\mathrm{b}\} \in \mathbb{N} \tag{5.3}$$

where Σ is a 2-tuple of coverage requirement (Σ_c) and bandwidth requirement (Σ_b).

Each RAT option can be equivalent to a tuple of its coverage and bandwidth capabilities, respectively, such that

$$\mathcal{O}_i = (\varpi_{\mathrm{c},i}, \varpi_{\mathrm{b},i}), \quad \forall i \in \mathbb{I}, \tag{5.4}$$

where $\vec{\varpi}_\mathrm{c}$ and $\vec{\varpi}_\mathrm{b}$ are the vectors of coverage and bandwidth capabilities, and

$$\vec{\varpi}_\mathrm{c} = [\varpi_{\mathrm{c},1}, \varpi_{\mathrm{c},2}, ..., \varpi_{\mathrm{c},N_\mathrm{R}}], \tag{5.5}$$

and

$$\vec{\varpi}_\mathrm{b} = [\varpi_{\mathrm{b},1}, \varpi_{\mathrm{b},2}, ..., \varpi_{\mathrm{b},N_\mathrm{R}}]. \tag{5.6}$$

Moreover, $p_{\mathrm{c},i}$ represents the probability of satisfying the SU's coverage requirement with RAT i:

$$p_{\mathrm{c},i} = P(\varpi_{\mathrm{c},i} \geq \Sigma_\mathrm{c}). \tag{5.7}$$

Let x_c be a random variable that represents the coverage satisfaction, where $x_\mathrm{c} = 1$ if coverage satisfaction is achieved, and $x_\mathrm{c} = 0$ otherwise, such that

$$x_\mathrm{c} = \begin{cases} 1, & \text{if} \quad \varpi_{\mathrm{c},i} \geq \Sigma_\mathrm{c} \\ 0, & \text{otherwise.} \end{cases} \tag{5.8}$$

Then, the expected value of coverage satisfaction is

$$E[x_\mathrm{c}] = \sum_{i=1}^{N_\mathrm{R}} p_{\mathrm{c},i} p_{\mathrm{s},i}. \tag{5.9}$$

Similarly, $p_{\mathrm{b},i}$ represents the probability of satisfying the SU's bandwidth requirement with RAT i:

$$p_{\mathrm{b},i} = P(\varpi_{\mathrm{b},i} \geq \Sigma_{\mathrm{b}}). \tag{5.10}$$

Let x_{b} be a random variable representing the bandwidth satisfaction, such that:

$$x_{\mathrm{b}} = \begin{cases} 1, & \text{if} \quad \varpi_{\mathrm{b},i} \geq \Sigma_{\mathrm{b}} \\ 0, & \text{otherwise.} \end{cases} \tag{5.11}$$

The expected value of bandwidth requirement becomes:

$$E[x_{\mathrm{b}}] = \sum_{i=1}^{N_{\mathrm{R}}} p_{\mathrm{b},i} p_{\mathrm{s},i}. \tag{5.12}$$

Finally, the probability of full-satisfaction with RAT i is the multiplication of satisfaction probabilities of both coverage and bandwidth requirements:

$$p_{\mathrm{t},i} = p_{\mathrm{c},i} p_{\mathrm{b},i}. \tag{5.13}$$

□

5.4.2 Sensing Latency

Latency in this work refers to the delay caused by unsuccessful sensing attempts, where the SU senses a frequency channel that is already being used by a PU. Another attempt with a different frequency channel is required after each failure, hence latency increases with the increasing number of failures.

Let $p_{\mathrm{a},i}$ be the probability of finding a frequency hole with RAT i in the first attempt:

$$p_{\mathrm{a},i} = \frac{N_{\mathrm{a},i}}{N_{\mathrm{f},i}}, \tag{5.14}$$

where $N_{\mathrm{a},i}$ and $N_{\mathrm{f},i}$ are the number of available and existing frequency channels for RAT i, respectively.

Therefore, it is obvious that a higher probability of finding an available channel is obtained when the total number of channels is small, being the main idea behind the predictive sensing approach.

Let x_{a} be a random variable that represents finding a frequency hole in the selected RAT, where $x_{\mathrm{a}} = 1$ is frequency hole is found, and $x_{\mathrm{a}} = 0$ otherwise,

Table 5.1: Glossary

Parameter	**Description**
p_{a}	Probability of finding a vacant channel in the first attempt
p_{s}	Probability of being selected as an RAT to be sensed
p_{c}	Probability of satisfying coverage requirement with the selected RAT
p_{b}	Probability of satisfying bandwidth requirement with the selected RAT
p_{t}	Probability of full-satisfaction
N_{R}	Number of available RATs
N_{a}	Number of available channels in the selected RAT
N_{f}	Total number of channels in the selected RAT
Σ_{c}	Coverage requirement of SU
Σ_{b}	Bandwidth requirement of SU
ϖ_{c}	Coverage capability of the selected RAT
ϖ_{b}	Bandwidth capability of the selected RAT
w_{a}	Prioritisation weight for latency
w_{b}	Prioritisation weight for bandwidth requirement
w_{c}	Prioritisation weight for coverage requirement
$\Lambda_{\mathrm{p},i}$	Predicted relative data traffic of the selected action
$\mathfrak{C}_{\mathrm{q},\mathbb{S}_{\mathrm{A}}}$	Cost of being in $\mathbb{S}_{\mathrm{A}}$ (State-I)
$\mathfrak{C}_{\mathrm{q},\mathbb{S}_{\mathrm{B}}}$	Cost of being in $\mathbb{S}_{\mathrm{B}}$ (State-II)
$\mathfrak{C}_{\mathrm{q,c}}$	Cost function for coverage requirement
$\mathfrak{C}_{\mathrm{q,b}}$	Cost function for bandwidth requirement
W_{u}	Unit bandwidth (200 kHz)
N_{W}	Number of unit bandwidth

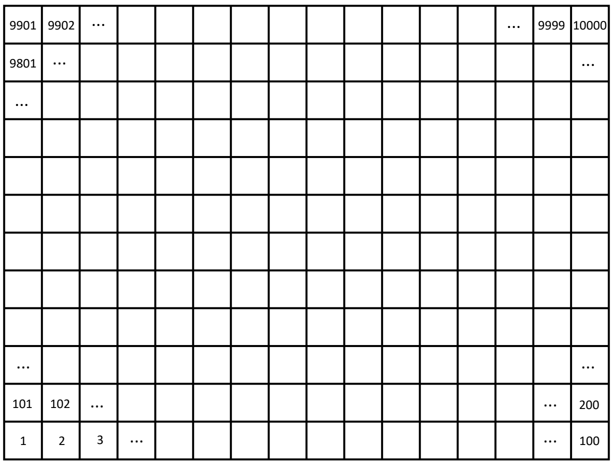


Figure 5.3: Milan city divided into square-shaped grids. Only the first $5,000$ grids are considered (the lower half).

then the expected value of finding a frequency hole is given as:

$$E[x_\mathrm{a}] = \frac{1}{N_\mathrm{R}} \sum_{i=1}^{N_\mathrm{R}} \frac{N_{\mathrm{a},i}}{N_{\mathrm{f},i}}. \tag{5.15}$$

5.5 Data Set & Preprocessing

There are two different data sets employed in this work: Milan city data set[2] for RAT-A and RAT-B, and a synthetic data set—inspired by [289] during generation—for RAT-C and RAT-D.

5.5.1 Data Set for RAT-A and RAT-B

The aforementioned CDR data set from Milan, Italy provided by Telecom Italia is employed to create the data set for RAT-A and RAT-B. Fig. 5.3 demonstrates the grid concept within the provided data set, whose call and text message activities are combined to create the historic data set for RAT-A, while the Internet activity is treated as the historic data set for RAT-B. Due to some missing data and for the sake of computational efficiency, the first 5,000 grids and first 3 weeks of November data are considered. The first two weeks of the three-week data set are used for training, while the remaining one week is used for testing.

[2] See Section 5.5 in Chapter 3 for the details about the dataset, which is available online at https://dandelion.eu/datamine/open-big-data/.

Table 5.2: User motifs for RAT-C and RAT-D

	Week Motif	**Day Motif**
RAT-C	Workday	Afternoon
RAT-D	Everyday	All day

5.5.2 Data Set for RAT-C and RAT-D

The historic data set for RAT-C and RAT-D is synthetically generated by being inspired by [289], in which real Wi-Fi data traffic was measured from 2147 wireless devices (196 residential gateways in 110 different cities) for 2 months. The primary objective of the authors is to capture Wi-Fi usage patterns of the users, and they extracted many user motifs (101 weekly motif and 112 daily motif), of which 14 weekly and 48 daily motifs have strong supports, meaning they are dominant over others [289].

In this study, two motifs are selected from [289] among the provided dominant ones, as shown in Table 5.2, to generate the data set for RAT-C and RAT-D, but the proposed study is not limited to any particular motif. The synthetic data generation is carried out for 8 weeks, where 7-week data is used for training purposes while 1-week data is used for testing.

5.6 Proposed Methodology

In the traditional CR spectrum access process, SUs perform either NB sensing or WB sensing methods in order to find an available frequency band to allocate. Both have some certain advantages and disadvantages over each other, such that spectrum opportunities are limited in the NB sensing with the narrower bandwidth of interest, leading to missing opportunities. In the WB sensing, on the other hand, albeit having more spectral opportunities, time spent for sensing is higher due to the larger size of the bandwidth.

In this study, as shown in Fig. 5.1b, a virtual WB sensing method is proposed, in which traffic load predictions for various RATs in WB are conducted in order to zoom into the most available spectrum portion, enabling NB sensing methods. More particularly, in case of multi-RAT availability, since each RAT may use different frequencies, the bandwidth of interest would most likely to be WB where NB sensing is no more applicable. In the proposed virtual WB sensing method, an RAT is chosen out of all the available ones through traffic load predictions and/or user requirements, and the spectrum sensing procedure is performed to the bandwidth of the selected RAT. As such, the WB is narrowed down to the

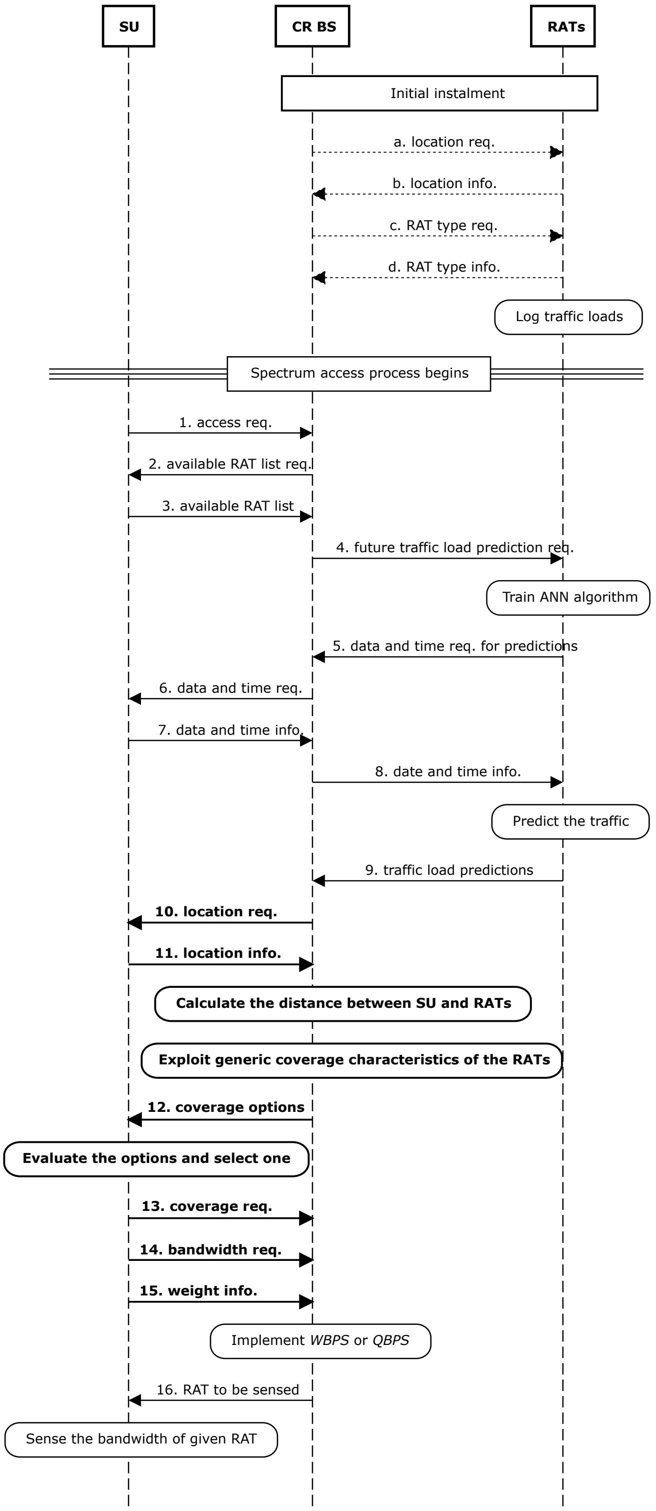


Figure 5.4: Sequence diagram showing the messaging among SU, CR BS, and RATs for the proposed methodology. Steps a-d (dashed) take place in case of new CR BS and/or RAT BS instalment. Steps 1-16 happen when the SU requests to initiate a connection. Steps 10-15 (bold) are executed only for QWBPS, while steps 1-9 and step 16 are executed for both WBPS and QWBPS.

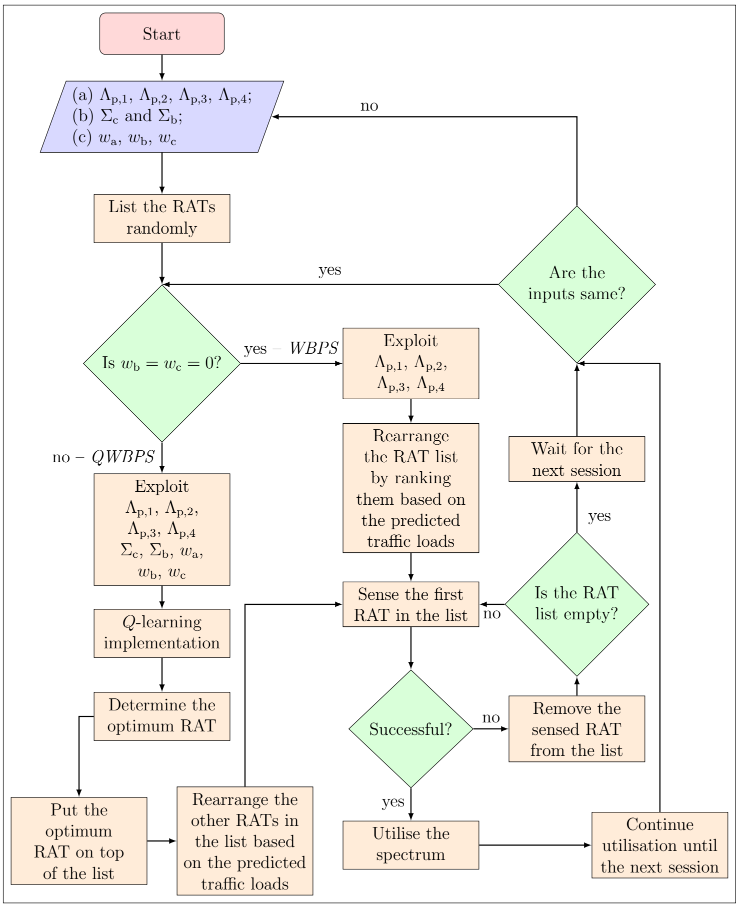

Start
(a) $\Lambda_{p,1}$, $\Lambda_{p,2}$, $\Lambda_{p,3}$, $\Lambda_{p,4}$;
(b) Σ_c and Σ_b;
(c) w_a, w_b, w_c
no
List the RATs randomly
yes
Are the inputs same?
Is $w_b = w_c = 0$?
yes – *WBPS*
Exploit $\Lambda_{p,1}$, $\Lambda_{p,2}$, $\Lambda_{p,3}$, $\Lambda_{p,4}$
no – *QWBPS*
Exploit $\Lambda_{p,1}$, $\Lambda_{p,2}$, $\Lambda_{p,3}$, $\Lambda_{p,4}$ Σ_c, Σ_b, w_a, w_b, w_c
Rearrange the RAT list by ranking them based on the predicted traffic loads
Wait for the next session
yes
Is the RAT list empty?
no
Sense the first RAT in the list
Q-learning implementation
Determine the optimum RAT
Successful?
no
Remove the sensed RAT from the list
yes
Put the optimum RAT on top of the list
Rearrange the other RATs in the list based on the predicted traffic loads
Utilise the spectrum
Continue utilisation until the next session

Figure 5.5: Flowchart for the two proposed methods. $\Lambda_{p,1}$, $\Lambda_{p,2}$, $\Lambda_{p,3}$, $\Lambda_{p,4}$ are the future traffic load predictions for RAT-A, RAT-B, RAT-C, and RAT-D, respectively. In case the optimum RAT determined by QWBPS does not have any available frequency channel, the process switches to WBPS.

bandwidth of the selected RAT, which in turn enables NB sensing approaches. Therefore, the proposed method uses the cooperation of WB and NB sensing by exploiting their inherent advantages. Fig. 5.4 demonstrates the messaging between the SU, CR BS, and RATs, and reveals how the proposed method might be implemented in real-life scenarios.

The second main contribution of the proposed method is to take the QoS requirements of the SUs into account, with which the objective function is adjusted accordingly. To that end, as illustrated in Fig. 5.5, the proposed *QoS-based optimisation* phase consists of two different decision strategies:

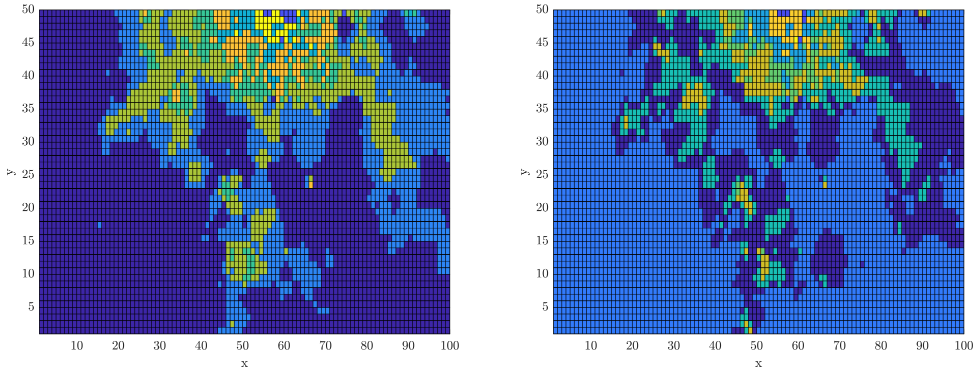
1. **WBPS:** as shown in Figs. 5.4 and 5.5, at each session (instance), when the SU wants to access the spectrum, the associated CR BS asks the RATs around the SU for future traffic load predictions. Then, the CR BS prepares an RAT list by ranking the RATs in ascending order according to their rel-

ative predicted traffic loads (percentage occupancy). Then, it directs the SU to the first RAT, whose traffic load is the minimum, to sense. If the SU cannot find a vacant channel in the selected RAT, it starts sensing the next RAT in the list, and this process continues until a free channel is found or there is no RAT left to sense. If the SU cannot find a vacant channel in any available RAT, then it counts the session as a fail and waits for the next session. This method targets only the latency minimisation using virtual WB sensing by making the bandwidth to be sensed narrower.

2. **QWBPS**: as a main contribution of this study, all the user requirements are taken into account in order to boost the satisfaction level of the SUs. Without loss of generality, coverage and bandwidth requirements are considered in addition to latency, but the proposed method is not limited to any specific requirement; any other requirement can easily be appended to the framework. As seen in Figs. 5.4 and 5.5, the CR BS receives the coverage and bandwidth requirements of the SU along with the QoS weights as inputs, and executes the developed Q-learning algorithm accordingly in order to determine the optimum RAT to sense. However, as Fig. 5.5 reveals, QWBPS switches to WBPS in case there is no frequency channel available in the determined optimum RAT.

5.6.1 Clustering for Milan City Data Set

Training a grid individually in Milan city data set is very prone to have an overfitting problem, as there is a limited number of samples for each grid. Therefore, it is a better approach to train the grids together for a better generalisation that can lead to better prediction performance, since many samples from various grids will be digested during the training. On the other hand, training only one ML algorithm for all the grids would decrease the prediction accuracy while keeping the algorithm well-generalised. In other words, the algorithm will try to obtain a model that somehow fits all the grids, however the model will be very unlikely to fit all the grids perfectly given that their characteristics are quite different from each other—while some are located at the city centre, some might be at more rural areas. Thus, there exist a trade-off between having a good generalisation and a good prediction accuracy. In this regard, clustering the grids based on their traffic loads can be an intelligent solution; the grids are clustered with their similar peers, hence generalisation can be provided by compromising less from the prediction accuracy due to the fact that there will be a different model for



(a) Clustering for RAT-A.

(b) Clustering for RAT-B.

Figure 5.6: Results of clustering the first 5,000 grids of the Milan city data set according to their average traffic load. (a) is for RAT-A, while (b) is RAT-B. Note that different colours (or shades in a black-and-white version of the figure) represent different clusters, and the colours in (a) and (b) are independent from each other. Values in x and y axes are for indexing purposes, and use $(y-1)50+x$ to find the index of a particular grid.

each cluster, consisting of elements with similar characteristics.

In this study, k-means clustering[3] is employed in order to cluster 5,000 grids according to their average traffic loads. k-means is an algorithm attempting to iteratively discover k different clusters in a data set with various samples. For each cluster there is a dedicated centroid [123], and the basic objective behind this algorithm is to place these centroids and associate the closest data points to them. In the learning phase, the positions of the clusters are altered by the average value of the associated data points in order to find an optimum clustering.

The elbow method [290] is used to find the optimum number of clusters by taking the percentage variation in the errors into consideration. In this regard, a 95% decrease in the error is determined as the stopping criterion for the employed elbow method. As such, the optimum number of clusters is found as 8 for the RAT-A, while it is 9 for RAT-B. The resulting clusters are shown in Fig. 5.6.

5.6.2 Future Traffic Load Prediction

Using the data sets elaborated in Section 5.5, the following future traffic load predictions are performed.

[3] Refer to Section 2.2.2 in Chapter 1 for details about the k-means algorithm.

Artificial Neural Networks

ANN is selected as a supervised learning method due to its easy implementation and high performances in terms of prediction accuracy [208]. Moreover, being independent of information about the underlying distribution of the available data set in order to obtain a model, ANN outclasses statistical models [280].

MSE is used as a cost function for all the developed ANN models, in which the error is given by

$$\mathfrak{C}_{\mathrm{D}} = \frac{1}{N_{\mathrm{s}}} \sum_{i=1}^{N_{\mathrm{s}}} (y_i - y_i')^2, \tag{5.16}$$

where N_{s} is the number of samples, y is the target value, and y' is the predicted value.

The aim of the training phase is to minimise $\mathfrak{C}_{\mathrm{D}}$ in (5.16) by properly arranging weights and bias values. The Bayesian regularisation [291] is employed as a training algorithm, since it is one of the strong BP training methods, preventing the network from over-fitting. More particularly, the Bayesian regularisation introduces an extra parameter to the cost function in (5.16) as follows:

$$\mathfrak{C}_{\mathrm{ANN}} = \dot{f}\mathfrak{C}_{\mathrm{D}} + \dot{g}\mathfrak{C}_{\mathrm{W}}, \tag{5.17}$$

where $\mathfrak{C}_{\mathrm{W}}$ is the cost implied by the ANN network weights, and $\dot{f}$ and $\dot{g}$ are the parameters to be determined. More specifically, $\mathfrak{C}_{\mathrm{W}}$ is given by

$$\mathfrak{C}_{\mathrm{W}} = \frac{1}{\|\vec{w}_{\mathrm{ANN}}\|} \sum_{j=1} \vec{w}_{\mathrm{ANN}}^2, \tag{5.18}$$

where $\vec{w}_{\mathrm{ANN}}$ is the neural network weight vector.

On one hand, in case $\dot{f} \gg \dot{g}$, the training phase will prioritise the error reduction and will be prone to over-fitting, while, on the other hand, the training response will be well-generalised, but the obtained error will be higher in the case of $\dot{f} \ll \dot{g}$ [292]. Therefore, the trade-off in $\dot{f}$ and $\dot{g}$ arises, and these parameters need to be carefully tuned in order to minimise $\mathfrak{C}_{\mathrm{D}}$ as well as having a good level of generalisation. In that regard, the Bayesian regularisation with Lavenberg-Marquardt optimisation, introduced in [292], is utilised as a part of the ML tool mentioned in Section 2.2.5.

As there are four different RATs included in the proposed system model, the future traffic load predictions have been performed separately due to their distinctive characteristics. First, as a generic model for all the RATs, a fully-connected feed-forward ANN with input, hidden, and output layers is employed. Then, the created generic model is customised individually for each RAT.

Table 5.3: Hyper-parametrisation for the different ANN models

	# HL Neurons	Data Split (Train-Validation-Test)(%)
RAT-A	7	50-25-25
RAT-B	14	50-25-25
RAT-C	7	50-25-25
RAT-D	3	50-25-25

Determining the number of hidden layer (HL) neurons is an important issue in ANN, and some methods are recommended in [293] for this issue, however they did not work well in the data sets in question, as all the methods suggest a small number of neurons which leads to under-fitting. Instead, an empirical approach is used to determine the optimal number of HL neurons. This approach considers gradually increasing the number of HL neurons starting from 1 to 20 in steps of 1 and evaluating the performance in terms of a cost function reflecting the obtained error. Similarly, an empirical method is followed in determining the data set split in terms of training, validation, and testing data. Basically, three different data splitting approaches are tested: 1) 50% training, 25% validation, 25% testing; 2) 60% training, 20% validation, 20% testing; and 3) 70% training, 15% validation, 15% testing.

After detailed hyper-parametrisation analyses, number of HL neurons and data splitting, as in Table 5.3, are determined for each RAT by considering the MSE performances of the ANN models. Significant MSE drop is investigated (mainly at least 95% drop is targeted) to select the number of HL neurons. Once at least 95% MSE drop is achieved, the number of HL neurons is no more incremented even if it causes better MSE performance by considering the generalisation of the model—the more neurons included in the network, the more the model is prone to over-fitting [294]. Similarly, 50-25-25 splitting is selected for all the ANN models, as there is no significant difference in terms of performance for the three data splitting approaches. The over-fitting problem is taken into account for this decision, since one of the reasons of over-fitting is over-training, leading to a lack of generalisation of the developed ANN model [294].

ANN models for RAT-A and RAT-B: The input layer consists of 3 nodes: 1) indices of the grids; 2) days of the week; and 3) time of the day. In order to convert days of the week and time of the day into numeric values, they are encoded to linearly separated numbers between 0 and 1. As such, for the days of a week, linearly separated 7 numbers generated between 0 and 1, and the days from Monday to Sunday are encoded to them respectively. Similarly, for the time

Table 5.4: List of possible states and associated costs

State	Description	Penalty Function ($\mathfrak{C}_Q$)
$\mathbb{S}_\mathrm{A}$	$\mathfrak{C}_\mathrm{q,c} = 1 \wedge \mathfrak{C}_\mathrm{q,b} = 1$	$w_\mathrm{c}\mathfrak{C}_\mathrm{q,c} + w_\mathrm{b}\mathfrak{C}_\mathrm{q,b} + w_\mathrm{a}\Lambda_{\mathrm{p},i} + \mathfrak{C}_{\mathrm{q},\mathbb{S}_\mathrm{A}}$
$\mathbb{S}_\mathrm{B}$	$\mathfrak{C}_\mathrm{q,c} = 0 \vee \mathfrak{C}_\mathrm{q,b} = 0$	$w_\mathrm{c}\mathfrak{C}_\mathrm{q,c} + w_\mathrm{b}\mathfrak{C}_\mathrm{q,b} + w_\mathrm{a}\Lambda_{\mathrm{p},i} + \mathfrak{C}_{\mathrm{q},\mathbb{S}_\mathrm{B}}$
$\mathbb{S}_\mathrm{C}$	$\mathfrak{C}_\mathrm{q,c} = 0 \wedge \mathfrak{C}_\mathrm{q,b} = 0$	$w_\mathrm{a}\Lambda_{\mathrm{p},i}$

of a day, linearly separated 144 numbers[4] are generated between 0 and 1, and time of a day is encoded to them accordingly. Only 1 output node, representing the data traffic loads, is included in the developed model for RAT-A and RAT-B.

ANN models for RAT-C and RAT-D: Two neurons, which are days of a week and time of a day, are employed in the input layer. Moreover, the data encoding procedure conducted for RAT-A and RAT-B is adopted for RAT-C and RAT-D, where the days of a week are encoded to 7 linearly separated numbers between 0 and 1, and the time of a day is encoded to linearly separated 144 numbers between 0 and 1. Similarly, there is only 1 neuron included at the output for the data traffic loads of RAT-C and RAT-D.

5.6.3 Proposed Q-Learning Framework

As seen in Fig. 5.1b, the proposed method includes a pre-decision process, where the portion of the spectrum (each RAT allocates different portions) to be sensed is determined based on either only the traffic load conditions (WBPS) or both the traffic load conditions and SU's requirements (QWBPS). WBPS relies on the predicted availability of all the portions, and ranks them in terms of their relative traffic loads to select the most available one. In QWBPS, however, the pre-decision phase takes the requirements and associated weights from SUs into account in addition to the predicted traffic loads. In order to accomplish this task, due to its model-free learning characteristics [55,228]—which learns from the environment by interacting with it[5]—the Q-learning algorithm [54] is employed, since it is one of the most outstanding RL methods that is capable of solving this kind of optimum policy-seeking problems.

Three different states are designed in this work based on the satisfaction status of the user requirements. The states and associated costs to be incurred for being in the states are shown in Table 5.4. w_a, w_b, and w_c in Table 5.4 are the prioritisation weights for latency, bandwidth, and coverage, respectively, and the

[4] The resolution of the data is 10 minutes, and there are 1440 minutes in a day, making 144 time slots.

[5] Refer to Section 2.2.3 for details about the Q-learning algorithm.

SU can tune them according to its preferences. $\Lambda_{\mathrm{p},i}$ is the predicted occupancy level of the taken action with RAT i. $\mathfrak{C}_{\mathrm{q},\mathbb{S}_\mathrm{A}}$ and $\mathfrak{C}_{\mathrm{q},\mathbb{S}_\mathrm{B}}$ are the penalties of being in $\mathbb{S}_\mathrm{A}$ and $\mathbb{S}_\mathrm{B}$, respectively, where $\mathfrak{C}_{\mathrm{q},\mathbb{S}_\mathrm{A}} > \mathfrak{C}_{\mathrm{q},\mathbb{S}_\mathrm{B}}$, encouraging the agent to move to the best possible state[6]. Hence, there is no such cost in $\mathbb{S}_\mathrm{C}$ as it is the best possible state. $\mathfrak{C}_{\mathrm{q,c}}$ is a cost function for the coverage requirement, where its value becomes 0 when the requirement is satisfied, and 1 otherwise:

$$\mathfrak{C}_{\mathrm{q,c}} = \begin{cases} 0, & \varpi_\mathrm{c} \geqslant \Sigma_\mathrm{c}, \\ 1, & \varpi_\mathrm{c} < \Sigma_\mathrm{c}. \end{cases} \tag{5.19}$$

Similarly, $\mathfrak{C}_{\mathrm{q,b}}$ is a cost function for the bandwidth requirement, where its value becomes 0 when the requirement is satisfied, and 1 otherwise, such that

$$\mathfrak{C}_{\mathrm{q,b}} = \begin{cases} 0, & \varpi_\mathrm{b} \geqslant \Sigma_\mathrm{b}, \\ 1, & \varpi_\mathrm{b} < \Sigma_\mathrm{b}. \end{cases} \tag{5.20}$$

Furthermore, the action list is provided in Table 5.5 where $\varpi_{\mathrm{b},1}$, $\varpi_{\mathrm{b},2}$, $\varpi_{\mathrm{b},3}$, $\varpi_{\mathrm{b},4}$ and $\varpi_{\mathrm{c},1}$, $\varpi_{\mathrm{c},2}$, $\varpi_{\mathrm{c},3}$, $\varpi_{\mathrm{c},4}$ are the available bandwidth and coverage capabilities of RAT-A, RAT-B, RAT-C, and RAT-D, respectively, where $\varpi_{\mathrm{c},1} > \varpi_{\mathrm{c},2} > \varpi_{\mathrm{c},4} > \varpi_{\mathrm{c},3}$. As seen from Table 5.5, there are basically four different actions that the agent can perform, which correspond to the RATs that the agent can choose to sense using NB sensing.

In order to develop an optimisation problem, first, a global cost function is defined as follows:

$$\mathfrak{C}(\Sigma, \mathcal{O}) = w_\mathrm{c}\mathfrak{C}_{\mathrm{q,c}} + w_\mathrm{b}\mathfrak{C}_{\mathrm{q,b}} + w_\mathrm{a} l, \tag{5.21}$$

where l is the sensing latency, which is modelled as $l \approx \Lambda_{\mathrm{p},i}$, since (5.14) implies that the probability of finding a vacant frequency channel is directly proportional to the occupancy level of the spectrum: the lesser probability of finding a vacant frequency channel, the more sensing latency it causes.

Then, the optimisation problem can formally be written as:

$$\begin{aligned} \min_{n} \quad & \mathfrak{C}(\Sigma, \mathcal{O}) && \text{(5.22a)} \\ \text{s.t.} \quad & \varpi_\mathrm{c} \leqslant \varpi_{\mathrm{c,m}}, && \text{(5.22b)} \\ & \varpi_\mathrm{b} \leqslant \varpi_{\mathrm{b,m}}, && \text{(5.22c)} \\ & n \leqslant N_\mathrm{R}, && \text{(5.22d)} \end{aligned}$$

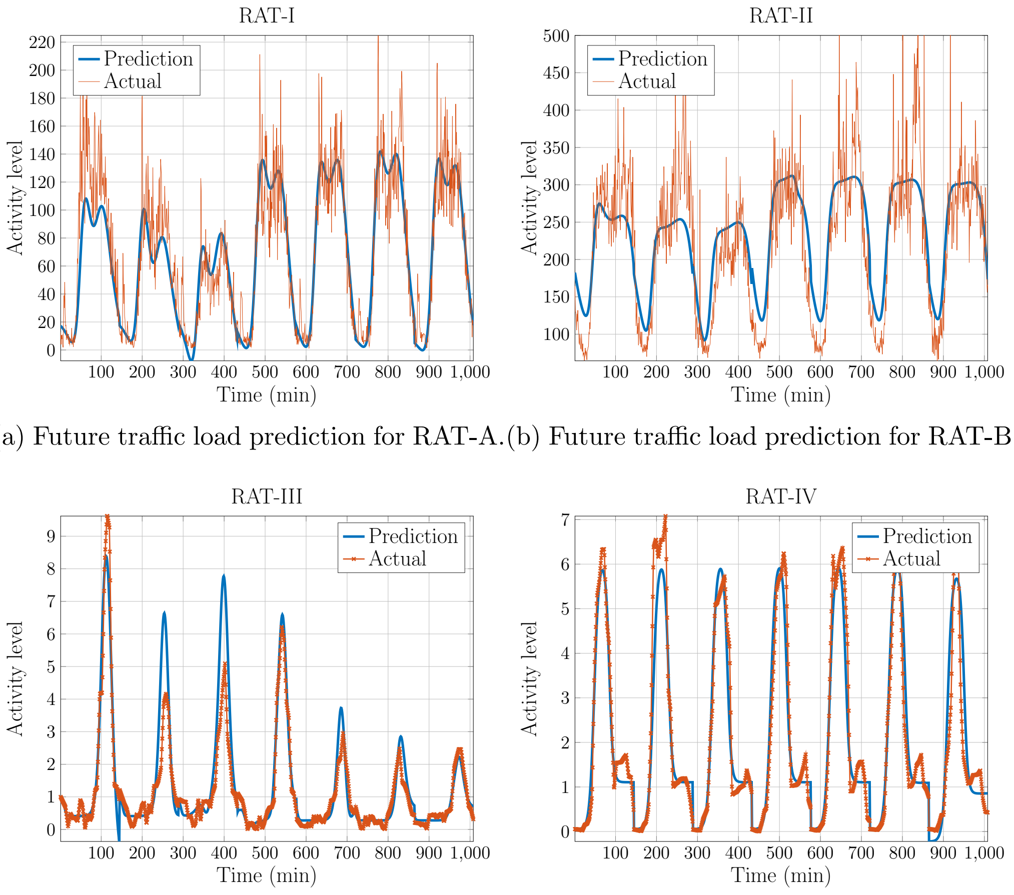
[6] A state is considered to be better if it satisfies more requirements, and thus the $\mathbb{S}_\mathrm{c}$ is the best state, while $\mathbb{S}_\mathrm{A}$ is the worst, with $\mathbb{S}_\mathrm{B}$ being in between.

Table 5.5: Q-Learning action list

Action	Description	Tuple ($\{\varpi_{\mathbf{c}}, \varpi_{\mathbf{b}}\}$)
ϖ_1	Sense $\mathcal{O}_1$	$\{\varpi_{\mathrm{c},1}, \varpi_{\mathrm{b},1}\}$
ϖ_2	Sense $\mathcal{O}_2$	$\{\varpi_{\mathrm{c},2}, \varpi_{\mathrm{b},2}\}$
ϖ_3	Sense $\mathcal{O}_3$	$\{\varpi_{\mathrm{c},3}, \varpi_{\mathrm{b},3}\}$
ϖ_4	Sense $\mathcal{O}_4$	$\{\varpi_{\mathrm{c},4}, \varpi_{\mathrm{b},4}\}$

where $\varpi_{\mathrm{c,m}}$ and $\varpi_{\mathrm{b,m}}$ are the maximum coverage and bandwidth supplies with the available RAT options.

Algorithm 2 in Chapter 2 is triggered to facilitate the proposed Q-learning approach, which SUs run for each session and determine the best action to take. The Q-learning algorithm allows the SUs to prioritise QoS components; i.e., cov-



(a) Future traffic load prediction for RAT-A.(b) Future traffic load prediction for RAT-B.

(c) Future traffic load prediction for RAT-C.(d) Future traffic load prediction for RAT-D.

Figure 5.7: Future traffic load predictions for (a) RAT-A, (b) RAT-B, (c) RAT-C, and (d) RAT-D. For (a) and (b), two weeks out of three weeks available data are used for training and one week is used for testing. For (c) and (d), seven out of eight weeks data are utilised for training purposes, while one week is used for testing. Note that the results are for a randomly selected grid.

erage, bandwidth, and sensing latency. If, for example, the SU prioritises latency the most (e.g., running a real time application), then the algorithm attempts to minimise the number of unsuccessful sensing attempts, as it is the main reason for the delay in the spectrum sensing phase. In case the SU is mostly mobile, for instance, it would prioritise the coverage requirement above all other components, as the RAT with the widest coverage area can keep the SU connected for a much longer time. Thus, the proposed QWBPS process is very strong and capable of dealing with various requirements, given that the algorithm is able to adjust itself according to the preferences of SUs.

5.7 Performance Evaluation

Table 5.6: Simulation parameters

Parameter	**Value**
Data rate of RAT-A	270.9 kbps
Number of channels for RAT-A	12
Data rate of RAT-B	2.85 Mbps
Number of channels for RAT-B	6
Data rate of RAT-C	100 Mbps
Number of channels for RAT-C	5
Data rate of RAT-D	51.85 Mbps
Number of channels for RAT-D	3
Grid indices used from Milan data set	$[1, 5000]$
Unit bandwidth (W_{u})	200 kHz
Number of unit bandwidth (N_{W})	$U \sim [1, 20]$
Days used from Milan data set	November, first 3 weeks
w_{a}, w_{b}, w_{c}	$[0, 5]$
$\mathfrak{C}_{\mathrm{q},\mathbb{S}_{\mathrm{B}}}$, $\mathfrak{C}_{\mathrm{q},\mathbb{S}_{\mathrm{A}}}$	5, 3
Number of iterations	100
Number of episodes	10

The model shown in Fig. 5.2 is used in this study to evaluate the proposed method, and the network is monitored for a week with 10 minutes resolution. All the parameters utilised in the simulations are provided in Table 5.6. In the evaluations, the proposed WBPS and QWBPS methods are compared with the developed random RAT selection approach, where SUs first select a random RAT to sense, and then channels in the frequency spectrum of the selected RAT are sensed also in a random manner.

The comparison is performed with the following metrics: (i) aggregated sensing latency, (ii) coverage satisfaction rate, (iii) bandwidth satisfaction rate, and

(iv) full-satisfaction rate. The aggregated sensing latency represents the total latency incurred during the sensing phase in a week. Note that as the sensing latency has a strong correlation with the number of unsuccessful sensing attempts, it is assumed that an unsuccessful sensing attempt causes a unit time (ut) of delay. The coverage and bandwidth satisfaction rates imply the percentage of instances in one week[7] that the coverage and bandwidth requirements are satisfied, respectively. The full-satisfaction rate, on the other hand, represents the percentage of instances that both coverage and bandwidth requirements are satisfied simultaneously. Note that since the existing predictive sensing methods, where the occupancy states of individual frequency channels are predicted, are conceptually different from the proposed methods, it is impossible to use them for comparison purposes. The proposed WBPS and QWBPS are more realistic and implementable for WB sensing, moreover QWBPS considers the QoS requirements of SUs as well.

Fig. 5.7 shows sample traffic load predictions for RAT-A, RAT-B, RAT-C, and RAT-D, respectively, for a random grid from a random cluster with 70-100 grids[8]. The results reveal that the proposed ANN manage to fit the data well, making the further phases implementable, since both pre-decision strategies (WBPS and QWBPS) use this predicted data as an input, thereby any significant error that occurs in this prediction phase can lead to massive errors at the output.

Fig. 5.8 demonstrates the obtained sensing latencies for WBPS and the random search for various data traffic levels. The purpose of this result is to reveal the behaviours of the random search and WBPS under different congestion levels, which is modelled by the data traffic loads; the more data traffic load an RAT experiences, the more congested it becomes. In particular, the load for each RAT is varied from 10% to 1000%—by assuming the available traffic is 100%—in order to obtain various data traffic loads (or congestion levels). As illustrated in Fig. 5.8, for the random search case, the obtained sensing latency increases tremendously when the congestion level becomes higher. Referring to the analytic model given in (5.15), the reason behind this result is that it is less likely to find a frequency hole to utilise when the network is more congested, leading to increasing number of unsuccessful attempts that causes increasing sensing latency. On the other hand, the sensing latency gradually grows for various congestion levels when WBPS becomes the adopted strategy, and WBPS decreases the la-

[7] An instance represents a 10 minute slot in a week simulation period. There is 1008 instances in total.

[8] The reason for putting this limitation to the cluster selection is just to make sure the cluster has sufficient number of elements for training while keeping it reasonable in order to avoid huge computational cost.

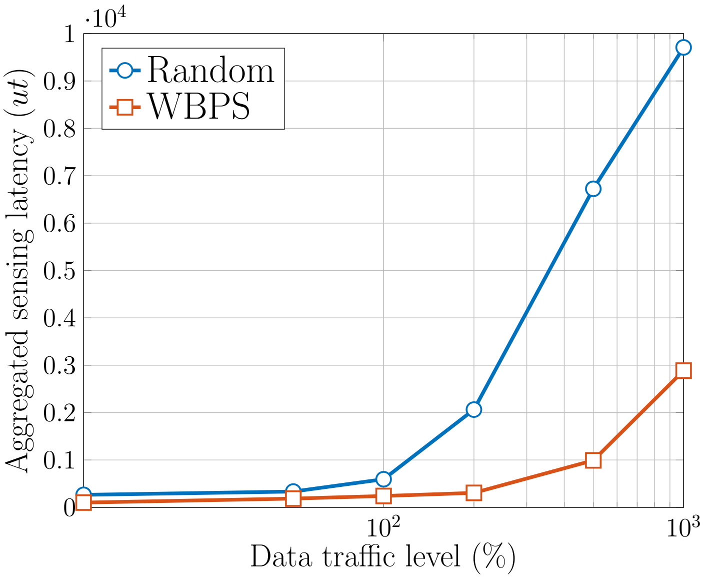


Figure 5.8: The sensing latency performances of WBPS and random search at various traffic levels for finding one frequency channel. The sensing latency is directly correlated to the number of unsuccessful sensing attempts, and it is assumed that an unsuccessful sensing attempt takes one unit time (ut) of delay. Note that the results are the average of 100 runs.

tency significantly (up to 85.25% when traffic load is 500%) by always choosing the RAT, with the minimum relative traffic load, to sense. Hence, it can clearly be seen that the probability of finding a vacant frequency channel is enhanced by WBPS, as it zooms on the least utilised portion of the available spectrum.

Fig. 5.9 shows the percentage satisfaction rate of the SU's coverage requirement. The QWBPS strategy managed to satisfy the SU at almost all the instances (99% success), while WBPS and random search satisfied the SU for around 546.7 (54.24% success) and 631.9 (62.69% success) times on average, respectively. These results also prove the superiority of QWBPS; it can focus on the SU's requirements and produce the output accordingly. The reason why QWBPS could not give 100% success in satisfying the coverage requirement is that there are cases where there is no vacant channel in the selected RAT. As seen in Fig. 5.5 that QWBPS switches to WBPS in such cases, meaning that it starts focusing on the sensing latency instead of any requirement. Random search, on the other hand, has no intention to satisfy the requirements, thereby the obtained results are—not surprisingly—the outcome of the random process. Appendix B.1 demonstrates the calculations for coverage satisfaction of the random search using the model presented in Section 5.4.1. It is straightforward from the calculations in Appendix B.1 and the findings in Fig. 5.9 that the mathematical model in Section 5.4.1 and the simulations are in line with each other owing to the very close

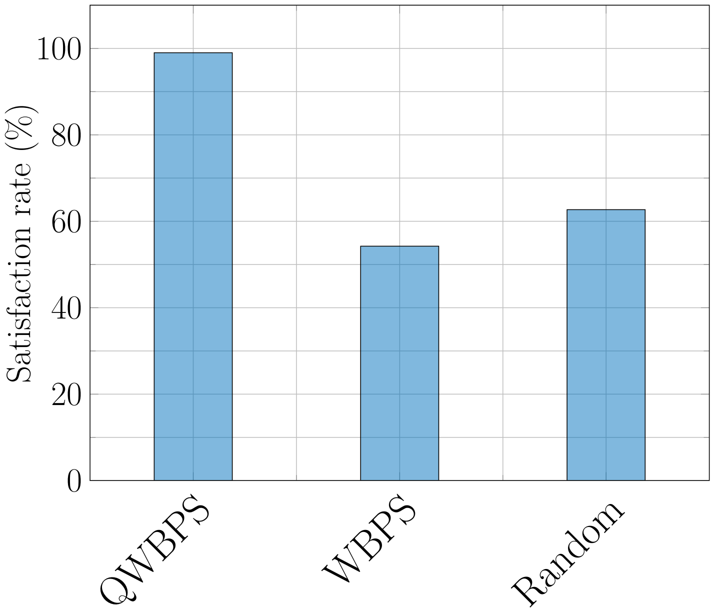


Figure 5.9: Percentage of coverage satisfaction for three different methods. The network is monitored for one week with 10-minute resolution. For the Q-learning part, $w_{\mathrm{a}} = w_{\mathrm{b}} = 0$ and $w_{\mathrm{c}} = 5$. Note that the obtained results are the averages of 100 runs.

results of both.

Besides, the WBPS strategy does not have any aim of satisfying the user requirements as well. However, the reason why it produced worse results than the random search is that it always focuses on the option which relatively has the most available resources. As RATs C and D, whose coverage capabilities are weak, mostly happened to offer the most relative resources, WBPS tends to select these options.

Fig. 5.10 demonstrates the results for the satisfaction level in terms of bandwidth requirements. It is observed from Fig. 5.10 that QWBPS (97.27% success) outperformed both random search (65.1% success) and WBPS (77.83%) by 49% and 25%, respectively, as it strictly prioritises the bandwidth requirement, enhancing the satisfaction level of the SU to a higher level. There could be two different reasons explaining why QWBPS could not give a 100% success rate: i) once QWBPS switches to WBPS when there is no available channel in the first selected RAT, the only focus becomes the sensing latency rather than specific requirements; ii) RAT determination is performed based on the predicted values, and there are some improper predictions, as seen in Fig. 5.7, affecting the performance in a negative way. However, the outstandingly good performance of QWBPS (97.27% success) would make these two issues tolerable. On the other hand, it is interesting that WBPS gives better results than the random search

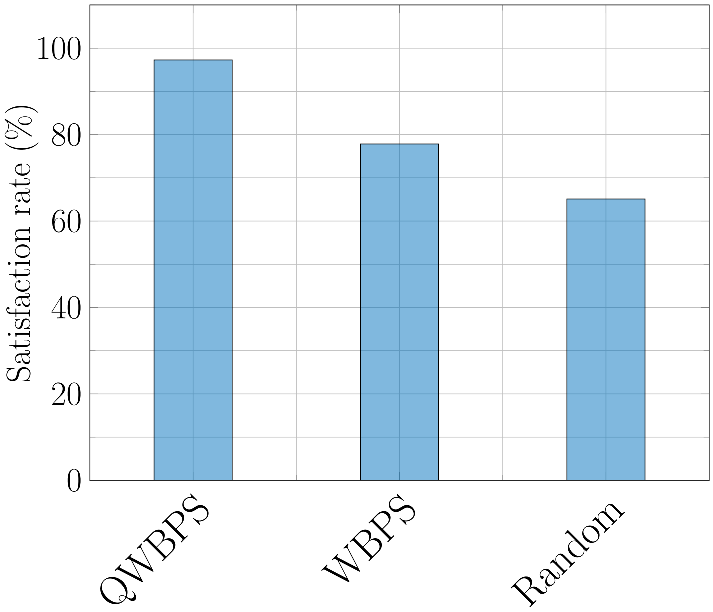
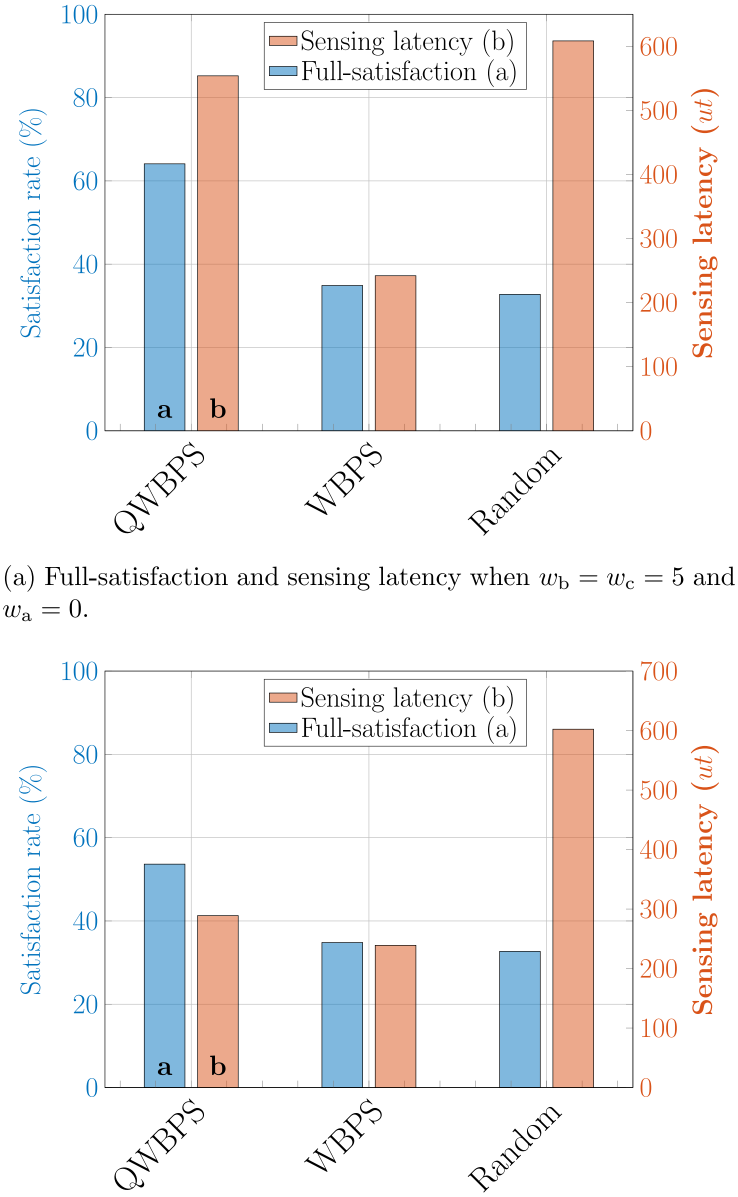


Figure 5.10: Percentage of bandwidth satisfaction for three different methods. The network is monitored for one week with 10-minute resolution. For the Q-learning part, $w_{\mathrm{a}} = w_{\mathrm{c}} = 0$ and $w_{\mathrm{b}} = 5$. Note that the obtained results are the averages of 100 runs.

without having any specific intention for that. The main rationale behind this outcome is that there is a strong correlation between minimising the number of unsuccessful sensing attempts and the bandwidth requirement, as the RAT option, with the least occupancy level, is likely to have sufficient bandwidth. Furthermore, the result for the random search case (65.1% success) is not surprising, and is the outcome of the random process similar to the results in Fig. 5.9. Appendix B.2 provides the calculations for bandwidth satisfaction of the random search using the model presented in Section 5.4.1.

The full-satisfaction levels and sensing latency performances for three different strategies are shown in Fig. 5.11, where coverage and bandwidth requirements are equally prioritised for the QWBPS case. In particular, coverage and bandwidth requirements are strictly prioritised with the developed weighting mechanism in both Figs. 5.11a and 5.11b, however, the sensing latency is discarded in Fig. 5.11a, whereas in Fig. 5.11b prioritises the sensing latency equally with the other requirements. The results demonstrated in Fig. 5.11a reveal that, when sensing latency was not prioritised along with the coverage and bandwidth requirements, QWBPS was able to enhance the full-satisfaction of WBPS and the random search by 95.7% and 83.8%, respectively. On the other hand, WBPS and QWBPS decreased the sensing latency of the random search by 60.2% and 8.96%, respectively, thereby it is hard to claim that QWBPS was strong in the sensing latency reduction. However, considering that no priority was given to the sensing latency while equally prioritising the coverage and bandwidth requirements, these results also prove that QWBPS works very well according to the priority inputs: it gave quite good results in terms of the full-satisfaction and performed poorly in the sensing latency.



(a) Full-satisfaction and sensing latency when $w_{\mathrm{b}} = w_{\mathrm{c}} = 5$ and $w_{\mathrm{a}} = 0$.

(b) Full-satisfaction success and sensing latency when $w_{\mathrm{a}} = w_{\mathrm{b}} = w_{\mathrm{c}} = 5$.

Figure 5.11: Results for sensing latency and the number of fully satisfied instances. Note that the results are the average of 100 runs. The indexing from (a) to (b) in the legend and the first set of bars are done for identification, and the same order follows for all sets of the bars.

In Fig. 5.11b—where the sensing latency, coverage and bandwidth requirements are all equally prioritised—it is demonstrated that the full-satisfaction level of QWBPS decayed by 16.3%, while the sensing latency performance was improved by 47.8%. Compared to the results in Fig. 5.11a,—when all the parameters (i.e., sensing latency, coverage and bandwidth requirements) are equally prioritised—QWBPS needed to compromise on the full-satisfaction to some extent in order to decrease the sensing latency. In that regards, on one hand, QWBPS boosted the full-satisfaction of WBPS and the random search by 54.1% and 64.1%, respectively, while, on the other hand, it managed to decrease the sensing latency of the random search by 52%. Furthermore, the difference between QWBPS and WBPS in terms of the sensing latency declined from 56.3% to 17.3%. These results affirm the superiority of QWBPS over the other methods, since it is capable of finding a good trade-off between the full-satisfaction and the sensing latency. The full-satisfaction results of the random search is again the outcome of the random process, as occurred in Figs. 5.9 and 5.10. If (5.1) and (5.13)) are used with (B.3) and (B.4), the expected value for the random search to satisfy both the requirements simultaneously becomes very similar to that obtained through the numerical simulations, proving that the analytic model in Section 5.4.1 works properly.

Note that although the results in Figs. 5.8, 5.9, 5.10, and 5.11 would be different for various simulation setups, such as different number of channels, different data rates, different RAT types, etc., they simply demonstrate the proof of concept for WBPS and QWBPS. Towards that end, one of the primary takeaways from all these results is that the QWBPS strategy is more versatile than WBPS, since it is capable of adjusting itself to different user requirements. Put it another way, QWBPS can be similar to WBPS when only the sensing latency is prioritised, however it has more capabilities that is not available in WBPS. In other words, WBPS can be said to be a subset of QWBPS, since it is only one task that QWBPS executes.

However, the power of QWBPS comes at the expense of computational complexity, as it includes an additional Q-learning implementation, and therefore, it is quite important to choose the correct method among WBPS and QWBPS by considering application-specific conditions. For example, if the application is latency-intolerant and does not value other QoS parameters, then there is no sense to use computationally more demanding QWBPS.

5.8 Summary

In this work, a novel and comprehensive virtual predictive WB spectrum sensing approach is proposed with a *QoS-optimisation* phase. In particular, the proposed approach introduces an intelligent interface between WB and NB sensing methods by benefiting from both. Moreover, it does not suffer from the issues of huge memory requirements and inflated energy consumption like the existing predictive sensing methods, because it treats the bandwidth of an RAT as a whole, which in turn reduces the amount of data to be handled.

Based on that, two different decision strategies are proposed in the novel *QoS-optimisation* phase: WBPS focuses only on minimising the sensing latency, while QWBPS considers user satisfaction as well. Both strategies have different purposes and strengths, which they proved during the performance evaluation by fulfilling their tasks successfully. Particularly, if the latency is the only concern for a given application, WBPS should be selected as a strategy, since it merely focuses on reducing the sensing latency. Moreover, due to the absence of a Q-learning implementation in its algorithm, WBPS is computationally less expensive than QWBPS. Nonetheless, if the user has multiple requirements, then the choice should be QWBPS owing to its strong multi-objective optimisation capabilities, which come at the expense of computational complexity.

Chapter 6

Conclusions, Future Trends, and Open Issues

This chapter provides conclusions regarding the cognitive networking concept in cellular networks followed by a summary of each work presented in the thesis, namely energy optimisation, capacity enhancement, and dynamic spectrum access. Then, a top-down approach is adopted while introducing future trends, such that the discussion starts from the cognitive networking in cellular networks and continues for each individual work. Lastly, open issues in the presented key design challenges are identified and discussed in order to provide future research directions.

6.1 Conclusions

The rapid proliferation of wireless devices as well as the advent of IoT have led to unprecedented data rate requirements among mobile users. Furthermore, in the IoT—realising the idea of Internet—a wide variety of devices, such as home appliances, vehicles, sensors, etc., would be assigned to a transmitting terminal. It is proposed that almost every device will be connected to the Internet for monitoring and controlling purposes: in the case of agriculture, for example, sensors would be deployed over large farms in order to monitor changes in the environment, such as temperature and humidity. In this scenario, the deployed sensors are required to connect to the Internet to inform the relevant people so that they can act accordingly. As the wireless connection is the most dominant option enabling devices to connect remote focal points, wireless networks will need to accommodate a massive number of devices without compromising their QoS. In this regard, there are many challenges that need to be addressed properly to be able to serve such a gargantuan number of connections.

In addition to IoT, data rate requirements of users with traditional handsets have also been increasing owing to the advances in technology, which accounts for: i) more advanced mobile handsets that have more computational capabilities; and ii) the emergence of bandwidth-hungry and delay-intolerant applications, including augmented reality, remote surgery, and tactile Internet, to name a few. Therefore, given that the demand from cellular networks has always been on the rise—making the network design more challenging—, stringent requirements in terms of data rate, latency, and energy efficiency have already been included in 5G NR. Even though new concepts including massive MIMO, network densification, and mmWave communications help in meeting the data rate requirements, they come with inevitable side effects: the energy consumption of cellular networks, for example, is expected to boost due to more intense BS deployments. In that regard, energy optimisation should also be taken into consideration while executing the tasks, since energy efficient networking, which is also referred to as green communication, is substantially demanded in future networks.

Moreover, as the number of devices connecting to the Internet is envisioned to be amplified exponentially, spectrum access will be another issue to be tackled. For this reason, the EM spectrum should be utilised more efficiently in order to combat this increase in the number of devices and to provide on-demand services. Besides, additional network capacity enhancement is also needed especially for UDNs and big events/gathering, such as sports competitions, concerts, etc., where the existing capacity would be insufficient. Thus, both network capacity enhancement and the efficient use of EM spectrum are vital to accommodate such kind of huge number of nodes/connections.

Based on these, it is clear that immense data volumes to be generated in cellular communication networks, especially due to i) the vast and rapid spread of IoT devices; ii) the increased number of BSs owing to network densification; and iii) the growing amount of data demand from users [4, 14, 17, 26–29, 31, 36–38]. Furthermore, these also result in more complexity in communication networks, making the network optimisation more challenging with increased number of parameters to consider, thereby it becomes harder and less practical to implement static and cumbersome solutions. Therefore, intelligence (or cognition) in the next generation of cellular networks will be crucial in order to make use of the significant amount of data generated and to combat more challenging network conditions.

Towards that end, as seen in Fig. 1.2, the concept of cognitive networking is adopted in this thesis for cellular communication networks, since it offers great potentials in terms of efficiency, agility, dynamism, and effectiveness by introducing automated actions [3, 52, 69, 71]. AI is employed as a generic methodology given that it is a strong, and often cost-effective, tool that can deal with stringent tasks in complex scenarios. In particular, ML—as a main component of AI—is used in order to bring such cognition to cellular networks by supporting learning from the generated data and gaining experience by interacting with the environment. In this regard, all types of ML techniques are employed including supervised learning (through ANN), unsupervised learning (through k-means), RL (through Q-learning and SARSA with VFA), and Markov chains.

After identifying the cognitive networking concept as the primary methodology of the thesis, energy optimisation, network capacity enhancement, and dynamic spectrum access are diagnosed as key design challenges for future networks. While identifying these design challenges, in addition to the aforementioned strong and convincing motivations, the inter-relation between them was also quite determinant: as presented in Fig. 1.1, the considered design issues may have immediate impacts on each other. For example, when a network is densified with more BS deployments to enhance the capacity, the frequency of HOs as well as the total energy consumption of the communication system proportionally increase, hence HO management and energy optimisation should be in the loop while expanding the network.

In summary, introducing cognition to wireless communication systems through learning from data and gaining experience is the main contribution provided in this thesis, and the proposed concept is implemented in three key and inter-related design challenges, which will play important roles in the next generation of cellular networks. In other words, this thesis is an attempt to make the cellular communication networks more sustainable and efficient with cognitive networking, which is enabled by data-driven learning.

As such, after employing AI techniques, the following gains are achieved:

- **energy optimisation:**
  - in the context-aware IoT connectivity and processing optimisation problem, the proposed method (LBRA) produces 0% joint cost out of 100%, while ScA, ScB, ScC, ScD, ScE, and ScF produce 15.61%, 39.34%, 37.06%, 60.77%, 76.27%, and 100%, respectively.
  - in the traffic-aware cell switching problem around 52% and 17% gains are observed in the total energy consumption of the network for simplistic and realistic scenarios, respectively.
- **network capacity enhancement:**

  - in the predictive mobility management problem, a significant gain, ranging from 8.23% to 25.37% for different randomness levels, are obtained on the HO signalling cost with the proposed 3-D transition matrix solution. Similarly, the prediction accuracy is improved by 8.23% for $\alpha = 0.1$ with the presented threshold-based method.
  - in the UAV positioning problem, the developed algorithm manages to reduce the number of users in outage by 76.33% and 80.47% compared to random and symmetric deployments, respectively.

- **dynamic spectrum access:** compared to the random search technique, the full-satisfaction is enhanced by 95.7% with QWBPS, while WBPS reduces the sensing latency by 85.25%.

6.1.1 Conclusions on Energy Optimisation in Cellular Networks

As mentioned earlier, energy consumption is one of the major problems in cellular networks, while being a crucial aspect to combat for mobile network operators due to two main reasons:

- increasing the energy consumption is environmentally dangerous, since it increases the CO_2 emission, which subsequently harms the environment;
- the energy bills also increase proportionally with the growing amount of energy consumption, which in turn damages the business sustainability and profitability.

However, in addition to the network energy consumption, the device energy should also be minimised—especially for IoT devices—given that they are often battery operated and their life-span is generally limited by their battery-life[1].

As such, in this thesis, both device and network side energy consumption is investigated and corresponding solutions are introduced. First, for IoT networks, an intelligent decision making mechanism is developed to jointly optimise the type of wireless connectivity and data processing unit. More particularly, a scenario—where multiple wireless connectivity (e.g., Wi-Fi and NB-IoT) and processing unit (e.g., device, fog, and cloud processing) options are available—is considered, and the requirements of IoT devices—in terms of security and response time—are also accounted along with their prioritisation weights. Then, a

[1] Recharging or replacing the battery of an IoT devices is needed once its battery depletes. Before this recharging or replacing process is executed, the device becomes unable to operate unless using another energy source and/or energy harvesting technique.

Q-learning algorithm is designed in order to jointly select the connectivity type and data processing unit by considering: i) the requirement of users, ii) total energy consumption, iii) total monetary charge due to the data processing fees, and iv) the remaining battery of the devices, such that if the remaining battery level of a device is under a certain threshold level (i.e., low-battery regime), then the algorithm ignores the monetary cost and focuses on the energy minimisation. The priority between the energy consumption minimisation and meeting the device requirements is determined by the corresponding weights: the higher values of the weights make the requirements more prioritised over the energy consumption, and vice versa. On the other hand, in case the remaining battery of the device is above the threshold (i.e., high-battery regime), then the algorithm loosens its energy consumption policies and focuses on the monetary cost. The priority between the monetary cost minimisation and requirement satisfaction is determined by the associated weights in a similar fashion to the low-battery regime. In addition to the connection type and processing unit, the proposed approach also optimises the percentage of data to be offloaded. The results suggest that the developed Q-learning assisted decision making algorithm performs quite well, as it manages to minimise the energy consumption under the low-battery regime while keeping the IoT devices satisfied with their requirements. Similarly, under the high-battery regime, the algorithm manages to minimise the monetary cost without compromising device requirements.

Second, the energy consumption minimisation is investigated from the network perspective by implementing a smart cell switching technique. More specifically, a CDSA environment is considered, where a CBS (i.e., MC) controls all the DBSs (i.e., SCs) under its coverage. The proposed algorithm is responsible for switching off/on SCs based on their relative data traffic in order to minimise the energy consumption of the network. Furthermore, the data traffic of the SCs that are switched off are offloaded to the MC, and thus the capacity of the MC becomes a constraint due to the limited capacity allocated to it. In this regard, the proposed algorithm is supposed to minimise the energy consumption of the network by switching off/on SCs without exceeding the available capacity of the MC. If the MC capacity is exceeded, on the other hand, the MC penalises the throughput of all the users with a certain amount in order to keep the users in service. This, in turn, results in a degradation in the RAN throughput. Note that the intelligent solution is implemented at the CBS given that it is located at an upper layer, meaning that it is capable of obtaining all the data traffic related information.

An RL based switching algorithm is developed in this work, but provided that the number of switching combinations exponentially grows with the number of SCs, a conventional RL implementation would make the solution unscalable. Therefore, SARSA with VFA is developed, as it is capable of dealing with huge state spaces owing to their function approximation technique rather than keeping a huge action-value table. The obtained results confirm that the proposed methodology minimises the energy consumption of the network without undermining the RAN throughput.

6.1.2 Conclusions on Capacity Enhancement in Cellular Networks

Although new concepts and technologies including mmWave and massive MIMO offer great potentials in terms of data rates, there is still room for improvements. First, the implementation of mmWave and network densification would increase HO frequencies owing to the reduced footprints of SCs, resulting in more HO costs to be incurred while users are commuting. This subsequently lessens the average network and user throughput[2], which is inversely proportional to HO cost. Given that the mentioned HO cost is a function of the number of HOs and the time spent for each HO, reducing them becomes a crucial task.

On the other hand, even after eliminating such inefficiencies from the HO process, further capacity enhancement is still needed, especially for UDNs, where the user intensity is much higher. In addition, the number of users increases extraordinarily in the occasions of big events, where many people gather together, thereby it would be harder to serve all the users with available network capacities—usually designed according to regular data demands.

Therefore, in this thesis, both of these problems are investigated and corresponding solutions are proposed. First, a predictive HO mechanism is proposed for more efficient HO management in the future generations of cellular networks, offering important reductions in HO signalling costs. In particular, two different issues are identified with the conventional Markov chains based HO predictors[3]: i) producing close transition probabilities due to revisits; and ii) making predictions regardless of the confidence level. A structural change in Markov chains is introduced for the former problem, where the traditional 2-D transition matrix is proposed to be changed to a 3-D one, in which the orders of HOs are included as an additional contextual information to make more informed predictions. The fundamental idea behind this concept is that, instead of having a 2-D transition

[2] This is an average throughput over time.

[3] Markov chains are one of the most commonly used methods for HO prediction.

matrix that includes all the transition probabilities for each state, a 3-D transition matrix is stored, in which each 2-D transition matrix is responsible for one particular HO order within a day. The obtained results reveal that the proposed methodology is more immune to the revisit problem, as its prediction performance is subject to less degradations when revisits are included in the user path.

A thresholding mechanism is presented for the former problem. More particularly, in order to prevent the Markov chains based HO predictors from making predictions even when the confidence level is not sufficient, a threshold is introduced by taking into account the HO signalling cost, such that non-predictive HO procedure is followed if the threshold criterion is not met. In other words, while the conventional Markov chains based HO predictors have a single criterion[4] to select the next state, the proposed methodology brings in another criterion, with the aim of protecting the predictors from incorrect predictions. The rationale behind this proposal is that triggering the HO process by relying on incorrect HO prediction incurs more signalling cost than that of the conventional non-predictive HO procedure. As such, minimising the instances of incorrect predictions becomes crucial, and the performance evaluation phase shows that the proposed threshold mechanism improves the prediction accuracy while reducing the HO signalling cost.

Second, given their multiple benefits in terms of flexibility and mobility, UAVs have already been considered as flying BSs by mounting SCs on them. While they can act as main BSs in emergency scenarios, where the existing communication infrastructure is damaged or destroyed, they can also be employed as network capacity enhancers for UDN scenarios and/or irregular big gatherings. However, their positioning is an important design parameters, since they should avoid interference with each other[5] while maximising[6] the capacity of the network. Therefore, a 2-D UAV positioning problem is converted to a clustering problem due to their inherent similarities, and a k-means algorithm is employed to find the latitudes and longitudes of the UAVs. After that, a trigonometric approach is followed to determine the altitudes by considering their antenna angle and the position of the outermost user. A comprehensive simulation campaign is performed, and the received SINR values are investigated thoroughly in order to observe the impacts of the altitude in the network capacity: higher altitudes are prone to result in more interference owing to more probable overlapping regions in the footprints of multiple UAVs.

[4] The only criterion is holding the maximum probability among all the candidate states.

[5] Even with the ground network in case the same carrier frequency is used.

[6] The maximum level may increase with the number of UAVs, but the maximisation mentioned here is in terms of capacity that can be achieved with a certain number of UAVs.

6.1.3 Conclusions on Dynamic Spectrum Access

With growing data rate demands due to advances in technology—resulting in the emergence of more bandwidth-hungry applications as well as development of more computationally capable UEs—the EM spectrum becomes an even more valuable commodity. Moreover, the fast spread of the IoT technology also contributes to the congestion in the spectrum owing to the huge number of devices connected to the Internet. In this regard, in this thesis, CRNs are studied provided that they offer a promising strength in terms of utilising the EM spectrum more efficiently. In particular, SUs, which are unlicensed and can only use the spectrum opportunistically, sense the spectrum in order to identify frequency holes to utilise. Based on the size of the bandwidth to be sensed[7], the sensing methods can be either NB or WB, both of which have advantages and disadvantages against each other, hence predictive sensing methods, where the occupancy states of frequency channels from WB are predicted and then sensed with NB techniques, have already been developed to create a bridge between the two.

Existing predictive sensing methodologies predominantly require historic occupancy data sets for each frequency channel to predict their future states. This, however, becomes impractical from different perspectives:

- it is unlikely to have an ever-present historic data set for each frequency channel;
- even if such data set was possible, ML implementations with such an immense volume of data would be very challenging;
- the flow and storage of such huge data is also an issue.

Therefore, in this thesis, the resolution of the prediction is decreased in a way that rather than predicting each individual frequency channels, the occupancy levels—or relative data traffics—of RATs are predicted. This is not only more practical and more feasible in terms of computational and storage constraints, but also more convenient in acquiring the needed data set, since only the traffic load information is required to make predictions.

Towards that end, if the only concern of an SU is the sensing delay, then the RATs are listed by ranking their percentage traffic loads in an ascending order, which is followed during the sensing phase. However, if the SU has additional requirements in terms of data rate and mobility, a further implementation, where Q-learning is involved as a decision maker, is executed. In other words, unlike the existing works, the requirements and their associated weights are reported by the

[7] Compared to the coherence bandwidth.

SUs, which then become inputs for the developed Q-learning algorithm along with the predicted relative traffic loads of the RATs. Then, based on the requirements and their corresponding prioritisation weights, the developed algorithm selects the optimum RAT to sense, and the results obtained from numerical simulations present that the developed methods are capable of remarkably decreasing the sensing latency while keeping the SUs satisfied.

6.2 Future Trends and Open Issues

In this section, future trends on each individual design challenge (i.e., energy optimisation, capacity enhancement, and dynamic spectrum access) are presented along with the identification of open issues.

6.2.1 Future Trends and Open Issues in Energy Optimisation

This section is divided into two in order to reflect the energy optimisation trends in IoT networks and cell switching independently.

Energy Optimisation in IoT Networks

As mentioned and implemented in this thesis, cloud computing has been gaining a significant amount of attention, since it advances the capabilities of the network by providing additional resources at the cloud, such as computational power, storage, and energy [39, 295–300]. However, although cloud computing relieves the burden on IoT devices and reduces their energy consumption, it can increase the overall energy consumed for a certain task. Therefore, the current focus is to make the cloud computing more energy efficient in order to provide a greener IoT networking. In this regard, energy efficient network function virtualisation using virtual machines becomes an important part of energy saving in cloud computing [298], but this requires the optimisation of physical machine scheduling [301].

Moreover, security and privacy are other challenging issues for cloud computing due to the fact that it is quite likely that mobile IoT devices will be cooperatively connected to the Internet [295]. This, in turn, makes them vulnerable to threats, such as violating the data privacy and security, thereby a considerable amount of research is needed to ensure the security of data transmission [295]. Furthermore, in the case of cloud computing, an optimisation is also needed in terms of communication, storage, and computation, such that the communication link is to be minimised, while utilising the storage and computational resources offered through cloudifiying [300].

Lastly, albeit being a well-developed concept, energy harvesting will play a vital role in keeping the IoT networks alive for longer [302]. Solar power, for example, is a very good energy source and can easily be integrated into IoT networks through photovoltaic panels. Moreover, owing to the new technologies adopted in 5G networks, such as massive MIMO and mmWave communications, wireless charging would be made more efficient, since they benefit from a good level of antenna directivity, which in turn improves transmitter and receiver antenna efficiencies [302].

Energy Efficiency through Cell Switching

As also implemented in this thesis, ML approaches due to their great potentials in optimisation problems have been playing crucial roles in providing energy efficiency for cellular networks [37, 51, 53, 62, 127]. In that regard, the application of deep learning and data analytics in order to make cellular networks more energy efficient seems as a viable solution [51, 303], however, there are multiple issues to be addressed to benefit from these concepts:

- data collection is the most fundamental part of both deep learning and big data analytics, and thus reliable and effective data collection policies are supposed to be determined;

- deep learning and big data analytics implementations often require huge computational power, which might undermine their energy saving. In addition, the processing units and the routing of data should also be optimised based on a given application;

- the storage of data is also challenging due to its volume, and thus the storage units and the type of data (e.g., raw, abstract, etc.) become important issues to be accounted;

- the type of data to be collected is vital to avoid collecting redundant data, which subsequently helps in reducing the computational complexity and storage requirements.

Besides, since UAV deployments have been a popular idea in cellular networks, their energy efficiency is also important as they act as flying BSs [127]. There are two main trade-offs to consider:

- **downlink:** provided that the UAVs are mostly battery operated, their energy consumption is important to maximise their flight-time for longer. As such, it would be preferable to avoid some movements, which result in higher energy consumption, but at the expense of poorer signal quality.

- **uplink:** unlike the downlink scenario, the movement of the UAV not only provides better signal quality, but can also offer energy efficiency for the transmitter due to the fact that the required transmit power is lessened with better channel conditions. On the other hand, UAV still consumes a lot of energy due to the continuous movements.

Therefore, the policies for the UAVs should be determined based on the requirement of the use-case as well as the circumstances experienced. Nonetheless, the UAV positioning is a challenging issue that involve multiple considerations including localisation of the users, optimisation implementations, etc.

6.2.2 Future Trends and Open Issues in Capacity Enhancement

Intelligent reflecting surfaces (IRSs) have recently gained a considerable interest, since they are shown to improve the received SINR significantly, which in turn enhances the user throughput. The definition of IRS is presented in an quite explanatory way in [304] as follows: *"a man-made two-dimensional (2D) surface of EM material, namely metasurface, that is composed of a large array of passive scattering elements with specially designed physical structure"*. The basic idea behind IRS assisted wireless networking is that they are deployed at a location different than a BS and responsible for beamforming the signal travelling from the source to its destination [305,306] via software-defined controlling [304]. This, by its turn, improves the received SINR either by increasing the receiving power or minimising the interference due to the well-tuned beamformed signals [304,305]. Nevertheless, since the concept is still in its infancy, a lot of works are needed to prove its superiority over the existing technologies that are already identified in 5G. In [305], for example, the authors analytically proved that IRS-assisted communication requires immense number of elements to provide as much SNR as massive MIMO. Moreover, the design of a proper propagation modelling is a challenging issue, and there is an enormous need for researches focusing on this [306]. Furthermore, the energy consumption of IRSs due to the channel estimation is another issue to combat owing to the fact that channel sensing and estimation require intense signal processing activities, which subsequently contributes to the overall energy consumption [304]. Thus, more energy efficient approaches are necessitated in order to keep the concept practical.

6.2.3 Future Trends and Open Issues in Dynamic Spectrum Access

Despite the fact that the concept of CR supplies an efficient use of the EM spectrum, many issues immediately arise because a sort of sharing and spectrum leasing are involved in the process. Blockchain technology, on the other hand, has been increasing its popularity in wireless communication networking due to its reliability, security, and resilience [307–311]. As such, the blockchain technology is expected to play an important role in enabling CRNs by providing a secure and reliable transaction system. For example, a PU, which holds the license for a certain spectrum portion, would open up its spectrum fully or partially for leasing during the periods that it either does not utilise it at all or under-utilises. In case there are multiple SUs trying to access the spectrum, they need to compete with each other in order to secure the spectrum access, and an auction system has already been proposed in the literature for this purpose [307]. Thus, the blockchain technology can play its role here by providing more efficient, reliable, and secure leasing mechanism, as it is done in [307, 308]. Alternatively, the blockchain technology can be employed for identity management, as in [309], and admission controlling, as proposed in [311].

Nevertheless, despite multiple benefits in terms of security and reliability, there are still open issues to address. For example,

- albeit being more reliable and secure, the use of public blockchains is a challenging task due to the increased energy consumption for mining [310];
- as mentioned in [312], the blockchain systems can also be subject to attacks, which may make the security of them questionable;
- the privacy is another concern for the blockchain system in general [313], and the privacy of transactions is an important consideration when used in auction based CRNs.

In other words, the energy consumption, security, and privacy should be addressed properly in order to integrate the CR concept with the blockchain technology. Hence, there are many issues to be solved to make these two concepts work together, since the integration would require some substantial changes. In this regard, more research activities are needed in order to demystify the actual potentials and practical challenges.

Appendix A

Normalised RAN Throughput

After penalisation, the provided throughput for $\mathcal{B}_i$ at time t can be expressed as the product of the average user throughput and the number of users it serves, as

$$\mathfrak{T}_{\mathrm{p},i}(t) = \hat{\mathfrak{t}}_{\mathrm{u},i}(t) N_{\mathrm{u},i}. \tag{A.1}$$

After that, using (3.51) into (A.1), it is obtained that

$$\mathfrak{T}_{\mathrm{p},i}(t) = \left(\mathfrak{t}_{\mathrm{u},i}(t) - \Upsilon_i\right) N_{\mathrm{u},i}. \tag{A.2}$$

When (3.52) is used in (A.2), $\mathfrak{T}_{\mathrm{p},i}(t)$ becomes

$$\mathfrak{T}_{\mathrm{p},i}(t) = \ \mathfrak{t}_{\mathrm{u},i}(t) N_{\mathrm{u},i} - \begin{cases} \mathfrak{T}_{\mathrm{r},i}(t) - \mathfrak{T}_{\mathrm{m},i}, & \mathfrak{T}_{\mathrm{r},i}(t) > \mathfrak{T}_{\mathrm{m},i} \\ 0, & \text{otherwise,} \end{cases} \tag{A.3}$$

after simplifying.

Then, using (3.50), (A.4) can be rewritten as

$$\mathfrak{T}_{\mathrm{p},i}(t) = \begin{cases} \mathfrak{T}_{\mathrm{m},i}, & \mathfrak{T}_{\mathrm{r},i}(t) > \mathfrak{T}_{\mathrm{m},i} \\ \mathfrak{T}_{\mathrm{r},i}(t), & \text{otherwise.} \end{cases} \tag{A.4}$$

Next, the throughput can be normalised with respect to the installed capacity, as

$$\Lambda_i(t) = \frac{\mathfrak{T}_{\mathrm{r},i}(t)}{\mathfrak{T}_{\mathrm{m},i}}, \tag{A.5}$$

where $\Lambda_i(t)$ is the te load factor of B_i at time t, but as mentioned earlier, it is also treated as the normalised throughput of B_i at time t. Therefore, dividing (A.4) and using (A.5), the normalised throughput can be obtained as

$$\tilde{\mathfrak{T}}_{\mathrm{p},i}(t) = \begin{cases} 1, & \Lambda_i(t) > 1 \\ \Lambda_i(t), & 0 \le \Lambda_i(t) \le 1, \end{cases} \tag{A.6}$$

Then, using the unit step function, (A.6) can be rewritten as

$$\tilde{\mathfrak{T}}_{\mathrm{p},i}(t) = u(-\Lambda_i(t) + 1)\Lambda_i(t) + u(\Lambda_i(t) - 1). \tag{A.7}$$

Lastly, in order to calculate the total provided network throughput, a summation is performed over all BSs, arriving at (3.53).

Appendix B

Requirement Satisfaction through Random Search

B.1 Random Search in Coverage Satisfaction

Since there are four RAT options considered, the available option set becomes as follows:

$$\mathbb{O} = \{\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4\}. \tag{B.1}$$

In addition, it is assumed that SU's coverage requirement is within the range of Θ_c, such that

$$\Sigma_\mathrm{c} \leqslant \max(\varpi_\mathrm{c}). \tag{B.2}$$

Therefore, $E[p_\mathrm{c}]$ for this scenario becomes 0.625 using (5.9), where

$$p_{\mathrm{c},i} = \begin{cases} 1.00 & i = 1, \\ 0.75 & i = 2, \\ 0.50 & i = 3, \\ 0.25 & i = 4. \end{cases} \tag{B.3}$$

If the SU chooses RAT-I, its coverage requirement will definitely be satisfied, as RAT-I's coverage capability is the greatest of all the options in (B.1) and the assumption in (B.2) ensures that the SU cannot require more than available in ϖ_c. Furthermore, as the process follows the discrete uniform distribution, $p_{\mathrm{s},i} = 0.25,\ i \in \{1, 2, 3, 4\}$.

B.2 Random Search in Bandwidth Satisfaction

Using (B.1), and (B.4), the expected value in (5.12) is calculated as 0.655, which is in line with the result of 0.651 in Fig. 5.10. In the implementations, $\Sigma_{\mathrm{b}} = W_{\mathrm{u}} N_{\mathrm{W}}$, where W_{u} represents 200 kHz bandwidth, and N_{W} is a coefficient determining the number of 200 kHz bandwidth that the user requires. N_{W} follows the discrete uniform distribution between 1 and 20, $U \sim [1, 20]$, and the average probabilities of satisfying Σ_{b} for RAT-I, RAT-II, RAT-III, and RAT-IV are evaluated through (5.10) as follows by calculating the average number of available W_{u} values for each:

$$p_{\mathrm{b},i} = \begin{cases} 0.4 & i = 1, \\ 0.22 & i = 2, \\ 1.00 & i = 3, \\ 1.00 & i = 4. \end{cases} \tag{B.4}$$

Bibliography

[1] T. Rappaport, *Wireless communications: Principles and practice*, 2nd ed., ser. Prentice Hall communications engineering and emerging technologies series. Prentice Hall, 2002.

[2] A. Goldsmith, *Wireless Communications.* Cambridge University Press, Aug. 2005. [Online]. Available: https://doi.org/10.1017%2Fcbo9780511841224

[3] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 336–361, First 2013.

[4] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[5] S. Faruque, *Code Division Multiple Access (CDMA).* Cham: Springer International Publishing, 2019, pp. 45–62. [Online]. Available: https://doi.org/10.1007/978-3-319-91651-4_5

[6] B. G. Evans and K. Baughan, "Visions of 4G," *Electronics Communication Engineering Journal*, vol. 12, no. 6, pp. 293–303, Dec 2000.

[7] P. Stuckmann and R. Zimmermann, "Toward ubiquitous and unlimited-capacity communication networks: European research in framework programme 7," *IEEE Communications Magazine*, vol. 45, no. 5, pp. 148–157, May 2007.

[8] ITU-R, "Requirements related to technical performance for IMT-Advanced radio interface(s)," Internation Telecommunication Union-Radio Communication Sector, Report M.2134, Nov. 2008.

[9] A. Adeel, M. Gogate, S. Farooq, C. Ieracitano, K. Dashtipour, H. Larijani, and A. Hussain, *A Survey on the Role of Wireless Sensor Networks and*

IoT in Disaster Management. Singapore: Springer Singapore, 2019, pp. 57–66. [Online]. Available: https://doi.org/10.1007/978-981-13-0992-2_5

[10] R. W. Heath, G. Laus, T. Q. S. Quek, S. Talwar, and P. Zhou, "Signal processing for the 5G revolution [from the guest editors]," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 12–13, Nov 2014.

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

[12] S. Kumar, G. Gupta, and K. R. Singh, "5G: Revolution of future communication technology," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, Oct 2015, pp. 143–147.

[13] 3GPP, "5G; Study on scenarios and requirements for next generation access technologies," 3rd Generation Partnership Project (3GPP), TS 38.913, Sept. 2018.

[14] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1201–1221, June 2017.

[15] 3GPP, "5G; NR; Base Station (BS) radio transmission and reception," 3rd Generation Partnership Project (3GPP), TS 38.104, July 2018.

[16] T. S. Rappaport, Y. Xing, G. R. MacCartney, A. F. Molisch, E. Mellios, and J. Zhang, "Overview of millimeter wave communications for fifth-generation (5G) wireless networks—with a focus on propagation models," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 12, pp. 6213–6230, 2017.

[17] S. A. Busari, S. Mumtaz, S. Al-Rubaye, and J. Rodriguez, "5G millimeter-wave mobile broadband: Performance and challenges," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 137–143, June 2018.

[18] A. Mesodiakaki, F. Adelantado, L. Alonso, M. Di Renzo, and C. Verikoukis, "Energy- and spectrum-efficient user association in millimeter-wave backhaul small-cell networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1810–1821, Feb 2017.

[19] D. Torrieri, S. Talarico, and M. C. Valenti, "Analysis of a frequency-hopping millimeter-wave cellular uplink," *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 7089–7098, Oct 2016.

[20] F. Jameel, Faisal, M. A. A. Haider, and A. A. Butt, "Massive MIMO: A survey of recent advances, research issues and future directions," in *2017 International Symposium on Recent Advances in Electrical Engineering (RAEE)*, Oct 2017, pp. 1–6.

[21] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive mimo detection techniques: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3109–3132, 2019.

[22] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, February 2014.

[23] F. E. Idachaba, "5G networks: Open network architecture and densification strategies for beyond 1000x network capacity increase," in *2016 Future Technologies Conference (FTC)*, 2016.

[24] B. Romanous, N. Bitar, A. Imran, and H. Refai, "Network densification: Challenges and opportunities in enabling 5G," in *2015 IEEE 20th International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, Sep. 2015, pp. 129–134.

[25] J. Liu, M. Sheng, L. Liu, and J. Li, "Network densification in 5G: From the short-range communications perspective," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 96–102, Dec 2017.

[26] Ericsson, "Ericsson Mobility Report," Tech. Rep., June 2019. [Online]. Available: https://www.ericsson.com/49d1d9/assets/local/mobility-report/documents/2019/ericsson-mobility-report-june-2019.pdf

[27] GSM Association, "The Mobile Economy 2019," Tech. Rep., Feb. 2019. [Online]. Available: https://www.gsma.com/r/mobileeconomy/

[28] S. Lee, M. Bae, and H. Kim, "Future of IoT networks: A survey," *Applied Sciences*, vol. 7, no. 10, p. 1072, 2017.

[29] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[30] ITU-T, “Overview of the Internet of things,” International Telecommunication Union, Recommendation Y.2060, June 2012.

[31] S. Bera, S. Misra, and A. V. Vasilakos, “Software-defined networking for Internet of things: A survey,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, Dec 2017.

[32] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, “Mobile augmented reality survey: From where we are to where we go,” *IEEE Access*, vol. 5, pp. 6917–6950, 2017.

[33] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, “Design considerations for a 5G network architecture,” *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, Nov 2014.

[34] I. F. Akyildiz, J. M. Jornet, and C. Han, “Terahertz band: Next frontier for wireless communications,” *Physical Communication*, vol. 12, pp. 16 – 32, 2014.

[35] C. Liu, M. Li, S. V. Hanly, P. Whiting, and I. B. Collings, “Millimeter-wave small cells: Base station discovery, beam alignment, and system design challenges,” *IEEE Wireless Communications*, vol. 25, no. 4, pp. 40–46, August 2018.

[36] L. D. Xu, W. He, and S. Li, “Internet of things in industries: A survey,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[37] M. E. Morocho Cayamcela and W. Lim, “Artificial intelligence in 5G technology: A survey,” in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct 2018, pp. 860–865.

[38] M. Agiwal, A. Roy, and N. Saxena, “Next generation 5G wireless networks: A comprehensive survey,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1617–1655, thirdquarter 2016.

[39] S. Li, L. D. Xu, and S. Zhao, “5G Internet of things: A survey,” *Journal of Industrial Information Integration*, vol. 10, pp. 1 – 9, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2452414X18300037

[40] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, “What will 5G be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[41] J. Wu, Y. Zhang, M. Zukerman, and E. K. Yung, "Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 803–826, 2015.

[42] T. S. Rappaport, W. Roh, and K. Cheun, "Mobile's millimeter-wave makeover," *IEEE Spectrum*, vol. 51, no. 9, pp. 34–58, Sep. 2014.

[43] M. Ozturk, M. Gogate, O. Onireti, A. Adeel, A. Hussain, and M. A. Imran, "A novel deep learning driven, low-cost mobility prediction approach for 5G cellular networks: The case of the control/data separation architecture (CDSA)," *Neurocomputing*, vol. 358, pp. 479 – 489, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231219300438

[44] M. Mollel, M. Ozturk, M. Kisangiri, S. Kaijage, O. Onireti, M. A. Imran, and Q. H. Abbasi, "Handover management in dense networks with coverage prediction from sparse networks," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, April 2019, pp. 1–6.

[45] R. Arshad, H. ElSawy, S. Sorour, T. Y. Al-Naffouri, and M. Alouini, "Co-operative handover management in dense cellular networks," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[46] Y. Chang, H. Liu, and H. Wei, "Group-based sidelink communication for seamless vehicular handover," *IEEE Access*, vol. 7, pp. 56 431–56 442, 2019.

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

[48] D. Datla, A. M. Wyglinski, and G. J. Minden, "A spectrum surveying framework for dynamic spectrum access networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4158–4168, Oct. 2009.

[49] Y. Liang, K. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: an overview," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 7, pp. 3386–3407, Sep. 2011.

[50] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[51] C. Zhang, P. Patras, and H. Haddadi, “Deep learning in mobile and wireless networking: A survey,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2224–2287, thirdquarter 2019.

[52] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, “A survey of machine learning techniques applied to self-organizing cellular networks,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017.

[53] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, “Application of machine learning in wireless networks: Key techniques and open issues,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3072–3108, Fourthquarter 2019.

[54] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An introduction.* MIT press Cambridge, 1998, vol. 1.

[55] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[56] W. Lin, Y. Hu, and C. Tsai, “Machine learning in financial crisis prediction: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 421–436, July 2012.

[57] A. Kamilaris and F. X. Prenafeta-Boldú, “Deep learning in agriculture: A survey,” *Computers and Electronics in Agriculture*, vol. 147, pp. 70 – 90, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169917308803

[58] J. Archenaa and E. M. Anita, “A survey of big data analytics in healthcare and government,” *Procedia Computer Science*, vol. 50, pp. 408 – 413, 2015, Big Data, Cloud and Computing Challenges. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915005220

[59] H. Zhang and L. Dai, “Mobility prediction: A survey on state-of-the-art schemes and future applications,” *IEEE Access*, vol. 7, pp. 802–822, 2019.

[60] X. Xing, T. Jing, W. Cheng, Y. Huo, and X. Cheng, “Spectrum prediction in cognitive radio networks,” *IEEE Wireless Communications*, vol. 20, no. 2, pp. 90–96, Apr. 2013.

[61] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 4, pp. 627–640, 2015.

[62] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 393–430, Firstquarter 2019.

[63] A. I. Abubakar, M. Ozturk, S. Hussain, and M. A. Imran, "Q-learning assisted energy-aware traffic offloading and cell switching in heterogeneous networks," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Sep. 2019, pp. 1–6.

[64] S. M. Asad, M. Ozturk, R. N. Bin Rais, A. Zoha, S. Hussain, Q. H. Abbasi, and M. A. Imran, "Reinforcement learning driven energy efficient mobile communication and applications," in *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2019, pp. 1–7.

[65] Y. A. Sambo, G. C. Valastro, G. M. M. Patané, M. Ozturk, S. Hussain, M. A. Imran, and D. Panno, "Motion sensor-based small cell sleep scheduling for 5G networks," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Sep. 2019, pp. 1–5.

[66] A. Turkmen, M. S. Mollel, M. Ozturk, S. Yao, L. Zhang, R. Ghannam, and M. A. Imran, "Coverage analysis for indoor-outdoor coexistence for millimetre-wave communication," in *2019 UK/ China Emerging Technologies (UCET)*, 2019, pp. 1–4.

[67] M. Ozturk, A. I. Abubakar, N. U. Hassan, S. Hussain, M. A. Imran, and C. Yuen, "Spectrum cost optimization for cognitive radio transmission over tv white spaces using artificial neural networks," in *2019 UK/ China Emerging Technologies (UCET)*, 2019, pp. 1–4.

[68] M. Ozturk, M. Jaber, and M. A. Imran, *Life-span Extension for Sensor Networks in the Industry.* John Wiley & Sons, Ltd, 2019, ch. 2, pp. 19–45.

[69] R. W. Thomas, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005.*, Nov 2005, pp. 352–360.

[70] J. M. III, “Cognitive radio architecture,” in *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems.* Springer Netherlands, pp. 43–107. [Online]. Available: https://doi.org/10.1007%2F978-1-4020-5542-3_3

[71] C. Fortuna and M. Mohorcic, “Trends in the development of communication networks: Cognitive networks,” *Computer Networks*, vol. 53, no. 9, pp. 1354 – 1376, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128609000085

[72] C.-T. Lee, Y.-H. Liang, P. H. Chou, A. H. Gorji, S. M. Safavi, W.-C. Shih, and W.-T. Chen, “EcoMicro: A miniature self-powered inertial sensor node based on bluetooth low energy,” in *Proceedings of the International Symposium on Low Power Electronics and Design.* ACM, 2018, p. 30.

[73] S. Kim and P. H. Chou, “Energy harvesting by sweeping voltage-escalated charging of a reconfigurable supercapacitor array,” in *Proc. IEEE/ACM Int. Symp. Low Power Electronics and Design*, Aug. 2011, pp. 235–240.

[74] A. González, R. Aquino, W. Mata, A. Ochoa, P. Saldaña, and A. Edwards, “Open-wise: A solar powered wireless sensor network platform,” *Sensors*, vol. 12, no. 6, pp. 8204–8217, 2012.

[75] Y. Wu, B. Li, and F. Zhang, “Predictive power management for wind powered wireless sensor node,” *Future Internet*, vol. 10, no. 9, p. 85, 2018.

[76] E. Vandelle, D. H. N. Bui, T. P. Vuong, G. Ardila, K. Wu, and S. Hemour, “Harvesting ambient RF energy efficiently with optimal angular coverage,” *IEEE Transactions on Antennas and Propagation*, p. 1, 2018.

[77] C. H. Yang, Y. Song, J. Jhun, W. S. Hwang, S. Do Hong, S. B. Woo, T. H. Sung, S. W. Jeong, and H. H. Yoo, “A high efficient piezoelectric windmill using magnetic force for low wind speed in wireless sensor networks,” *Journal of the Korean Physical Society*, vol. 73, no. 12, pp. 1889–1894, 2018.

[78] G. Verma and V. Sharma, “A novel thermoelectric energy harvester for wireless sensor network application,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3530–3538, May 2019.

[79] R. Elhabyan, W. Shi, and M. St-Hilaire, “Evolutionary-based coverage control mechanism for clustered wireless sensor networks,” in *International Conference on Wired/Wireless Internet Communication.* Springer, 2018, pp. 67–80.

[80] N. Dinh and Y. Kim, "An energy efficient integration model for sensor cloud systems," *IEEE Access*, vol. 7, pp. 3018–3030, 2019.

[81] C. Cheng and C. Yu, "Mobile data gathering with bounded relay in wireless sensor networks," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3891–3907, Oct. 2018.

[82] D. Renga, H. Al Haj Hassan, M. Meo, and L. Nuaymi, "Energy management and base station on/off switching in green mobile networks for offering ancillary services," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 3, pp. 868–880, Sep. 2018.

[83] J. Chen, X. Ge, X. Song, and Y. Zhong, "Base-station switch-off with mutual repulsion in fifth-generation massive multi-input–multi-output networks," *IET Communications*, vol. 12, no. 16, pp. 2038–2045, 2018.

[84] T. Beitelmal, S. S. Szyszkowicz, D. G. González, and H. Yanikomeroglu, "Sector and site switch-off regular patterns for energy saving in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 2932–2945, May 2018.

[85] H. Farooq, A. Asghar, and A. Imran, "Mobility prediction-based autonomous proactive energy saving (AURORA) framework for emerging ultra-dense networks," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 4, pp. 958–971, Dec 2018.

[86] E. R. Bastidas-Puga, . G. Andrade, G. Galaviz, and D. H. Covarrubias, "Handover based on a predictive approach of signal-to-interference-plus-noise ratio for heterogeneous cellular networks," *IET Communications*, vol. 13, no. 6, pp. 672–678, 2019.

[87] A. Mohamed, M. A. Imran, P. Xiao, and R. Tafazolli, "Memory-full context-aware predictive mobility management in dual connectivity 5G networks," *IEEE Access*, vol. 6, pp. 9655–9666, 2018.

[88] H. Farooq and A. Imran, "Spatiotemporal mobility prediction in proactive self-organizing cellular networks," *IEEE Communications Letters*, vol. 21, no. 2, pp. 370–373, Feb 2017.

[89] W. Zhang, Y. Liu, T. Liu, and C. Yang, "Trajectory prediction with recurrent neural networks for predictive resource allocation," in *2018 14th IEEE International Conference on Signal Processing (ICSP)*, Aug 2018, pp. 634–639.

[90] Y. Sun, Y. Chang, M. Hu, and T. Zeng, “A universal predictive mobility management scheme for urban ultra-dense networks with control/data plane separation,” *IEEE Access*, vol. 5, pp. 6015–6026, 2017.

[91] M. Gapeyenko, V. Petrov, D. Moltchanov, S. Andreev, N. Himayat, and Y. Koucheryavy, “Flexible and reliable UAV-assisted backhaul operation in 5G mmwave cellular networks,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2486–2496, Nov 2018.

[92] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, and T. Umer, “Energy-efficient industrial Internet of UAVs for power line inspection in smart grid,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2705–2714, June 2018.

[93] R. Amorim, H. Nguyen, P. Mogensen, I. Z. Kovács, J. Wigard, and T. B. Sørensen, “Radio channel modeling for UAV communication over cellular networks,” *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 514–517, Aug 2017.

[94] P. V. Klaine, J. P. B. Nadas, R. D. Souza, and M. A. Imran, “Distributed drone base station positioning for emergency cellular networks using reinforcement learning,” *Cognitive Computation*, vol. 10, no. 5, pp. 790–804, Oct 2018. [Online]. Available: https://doi.org/10.1007/s12559-018-9559-8

[95] X. Chen, J. Yang, and G. Ding, “Minimum bayesian risk based robust spectrum prediction in the presence of sensing errors,” *IEEE Access*, vol. 6, pp. 29 611–29 625, 2018.

[96] Z. Jin, K. Yao, B. Lee, J. Cho, and L. Zhang, “Channel status learning for cooperative spectrum sensing in energy-restricted cognitive radio networks,” *IEEE Access*, vol. 7, pp. 64 946–64 954, 2019.

[97] B. S. Shawel, D. Hailemariam Woledegebre, and S. Pollin, “Deep-learning based cooperative spectrum prediction for cognitive networks,” in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct 2018, pp. 133–137.

[98] V. Nguyen and O. Shin, “Cooperative prediction-and-sensing-based spectrum sharing in cognitive radio networks,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 1, pp. 108–120, March 2018.

[99] P. Zuo, X. Wang, W. Linghu, R. Sun, T. Peng, and W. Wang, “Prediction-based spectrum access optimization in cognitive radio networks,” in *2018*

IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Sep. 2018, pp. 1–7.

[100] P. Supraja, V. M. Gayathri, and R. Pitchai, “Optimized neural network for spectrum prediction using genetic algorithm in cognitive radio networks,” *Cluster Computing*, vol. 22, no. 1, pp. 157–163, Jan 2019. [Online]. Available: https://doi.org/10.1007/s10586-018-1978-5

[101] J. Sun, X. Liu, G. Ren, M. Jia, and Q. Guo, “A spectrum prediction technique based on convolutional neural networks,” in *Wireless and Satellite Systems*, M. Jia, Q. Guo, and W. Meng, Eds. Cham: Springer International Publishing, 2019, pp. 69–77.

[102] X. Zhang, F. Gao, R. Chai, and T. Jiang, “Matched filter based spectrum sensing when primary user has multiple power levels,” *China Communications*, vol. 12, no. 2, pp. 21–31, Feb 2015.

[103] Y. Arjoune, Z. E. Mrabet, H. E. Ghazi, and A. Tamtaoui, “Spectrum sensing: Enhanced energy detection technique based on noise measurement,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2018, pp. 828–834.

[104] P. Semba Yawada and A. J. Wei, “Cyclostationary detection based on non-cooperative spectrum sensing in cognitive radio network,” in *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, June 2016, pp. 184–187.

[105] Y. Chen, X. Jing, J. Mu, and J. Li, “An improved covariance spectrum sensing algorithm establish on AD test,” in *International Conference On Signal And Information Processing, Networking And Computers*. Springer, 2018, pp. 16–22.

[106] A. Kumar, S. Saha, and R. Bhattacharya, “Wavelet transform based novel edge detection algorithms for wideband spectrum sensing in crns,” *AEU - International Journal of Electronics and Communications*, vol. 84, pp. 100 – 110, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1434841117316771

[107] E. Y. Imana, T. Yang, and J. H. Reed, “Suppressing the effects of aliasing and iq imbalance on multiband spectrum sensing,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1074–1086, Feb 2017.

[108] H. Zhang, D. L. Ruyet, D. Roviras, and H. Sun, "Polyphase filter bank based multi-band spectrum sensing in cognitive radio systems," *International Journal of Communication Systems*, vol. 29, no. 12, pp. 1844–1862, apr 2014. [Online]. Available: https://doi.org/10.1002%2Fdac.2798

[109] H. Zhao, S. Wang, and L. Li, "A directly seeking spectrum holes algorithm by compressive sampling," *International Journal of Communication Systems*, vol. 32, no. 15, p. e4107, jul 2019. [Online]. Available: https://doi.org/10.1002%2Fdac.4107

[110] J. Yang, M. Jia, X. Gu, and Q. Guo, "Low complexity sub-Nyquist wideband spectrum sensing for cognitive radio," *IEEE Access*, vol. 6, pp. 45 166–45 176, 2018.

[111] A. Asghar, H. Farooq, and A. Imran, "Self-healing in emerging cellular networks: Review, challenges, and research directions," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 1682–1709, thirdquarter 2018.

[112] E. Alpaydin, *Introduction to machine learning.* MIT press, 2014.

[113] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.

[114] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1994.

[115] M. A. Nielsen, *Neural Networks and Deep Learning.* Determination Press, 2015.

[116] L. Prechelt, *Early Stopping - But When?* Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 55–69. [Online]. Available: https://doi.org/10.1007/3-540-49430-8_3

[117] M. Kubat, *An Introduction to Machine Learning.* Springer International Publishing, 2015. [Online]. Available: https://doi.org/10.1007%2F978-3-319-20010-1

[118] T. Kansal, S. Bahuguna, V. Singh, and T. Choudhury, "Customer segmentation using k-means clustering," in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, Dec 2018, pp. 135–139.

[119] P. Mehta, H. Shah, V. Kori, V. Vikani, S. Shukla, and M. Shenoy, "Survey of unsupervised machine learning algorithms on precision agricultural data," in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, March 2015, pp. 1–8.

[120] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 159–170, Second 2010.

[121] J. Sun and C. Masouros, "Drone positioning for user coverage maximization," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 318–322.

[122] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651 – 666, 2010.

[123] H. Xiong, J. Wu, and J. Chen, "K-means clustering versus validation measures: A data-distribution perspective," *Part B (Cybernetics) IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 2, pp. 318–331, Apr. 2009.

[124] P. H. V. Klaine, "Self-organization for 5G and beyond mobile networks using reinforcement learning," Ph.D. dissertation, University of Glasgow, 2019.

[125] R. Douc, E. Moulines, P. Priouret, and P. Soulier, *Markov Chains*. Springer International Publishing, 2018. [Online]. Available: https://doi.org/10.1007%2F978-3-319-97704-1

[126] A. Mohamed, O. Onireti, M. A. Imran, A. Imran, and R. Tafazolli, "Predictive and core-network efficient RRC signalling for active state handover in RANs with control/data separation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1423–1436, March 2017.

[127] M. A. Imran, A. F. dos Reis, G. Brante, P. V. Klaine, and R. D. Souza, "Machine learning in energy efficiency optimization," pp. 105–117, dec 2019. [Online]. Available: https://doi.org/10.1002%2F9781119562306.ch6

[128] E. Björnson and P. Giselsson, "Two applications of deep learning in the physical layer of communication systems," 2020.

[129] S. Manzoor, S. Mazhar, A. Asghar, A. Noor Mian, A. Imran, and J. Crowcroft, "Leveraging mobility and content caching for proactive load balancing in heterogeneous cellular networks," *Transactions on Emerging Telecommunications Technologies*, p. e3739, 2019.

[130] H. Gebrie, H. Farooq, and A. Imran, "What machine learning predictor performs best for mobility prediction in cellular networks?" in *2019 IEEE International Conference on Communications Workshops (ICC Workshops).* IEEE, 2019, pp. 1–6.

[131] P. Fazio, M. Tropea, C. Sottile, S. Marano, M. Voznak, and F. Strangis, "Mobility prediction in wireless cellular networks for the optimization of call admission control schemes," in *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE).* IEEE, 2014, pp. 1–5.

[132] A. Nadembega, A. Hafid, and T. Taleb, "Mobility-prediction-aware bandwidth reservation scheme for mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2561–2576, 2014.

[133] A. Mohamed, O. Onireti, S. A. Hoseinitabatabaei, M. Imran, A. Imran, and R. Tafazolli, "Mobility prediction for handover management in cellular networks with control/data separation," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 3939–3944.

[134] S. Kumar, K. Kumar, and P. Kumar, "Mobility based call admission control and resource estimation in mobile multimedia networks using artificial neural networks," in *2015 1st International Conference on Next Generation Computing Technologies (NGCT).* IEEE, 2015, pp. 852–857.

[135] J. Liu, B. Krishnamachari, S. Zhou, and Z. Niu, "DeepNap: Data-driven base station sleeping operations through deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4273–4282, 2018.

[136] J. Ye and Y.-J. A. Zhang, "DRAG: Deep reinforcement learning based base station activation in heterogeneous networks," *arXiv preprint arXiv:1809.02159*, 2018.

[137] M. Imran, L. U. Khan, I. Yaqoob, E. Ahmed, M. A. Qureshi, and A. Ahmed, "Energy harvesting in 5G networks: Taxonomy, requirements, challenges, and future directions," 2019.

[138] C. H. Liu, X. Ma, X. Gao, and J. Tang, "Distributed energy-efficient multi-UAV navigation for long-term communication coverage by deep reinforcement learning," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[139] Y. Ye, M. Xiao, and M. Skoglund, "Mobility-aware content preference learning in decentralized caching networks," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2019.

[140] O. Semiari, W. Saad, M. Bennis, and B. Maham, "Caching meets millimeter wave communications for enhanced mobility management in 5G networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 779–793, Feb 2018.

[141] H. Farahat and H. S. Hassanein, "Proactive caching for producer mobility management in named data networks," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, June 2017, pp. 171–176.

[142] H. Eltom, S. Kandeepan, Y. Liang, and R. J. Evans, "Cooperative soft fusion for HMM-based spectrum occupancy prediction," *IEEE Communications Letters*, vol. 22, no. 10, pp. 2144–2147, Oct 2018.

[143] B. Hussain, Q. Du, S. Zhang, A. Imran, and M. A. Imran, "Mobile edge computing-based data-driven deep learning framework for anomaly detection," *IEEE Access*, vol. 7, pp. 137 656–137 667, 2019.

[144] C. V. Murudkar and R. D. Gitlin, "QoE-driven anomaly detection in self-organizing mobile networks using machine learning," in *2019 Wireless Telecommunications Symposium (WTS)*, April 2019, pp. 1–5.

[145] S. Chia, M. Gasparroni, and P. Brick, "The next challenge for cellular networks: backhaul," *IEEE Microwave Magazine*, vol. 10, no. 5, pp. 54–66, August 2009.

[146] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "A distributed SON-based user-centric backhaul provisioning scheme," *IEEE Access*, vol. 4, pp. 2314–2330, 2016.

[147] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, "Machine learning on big data: Opportunities and challenges," *Neurocomputing*, vol. 237, pp. 350–361, may 2017.

[148] A. L'Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, "Machine learning with big data: Challenges and approaches," *IEEE Access*, vol. 5, pp. 7776–7797, 2017.

[149] S. Suthaharan, "Big data classification," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, pp. 70–73, apr 2014.

[150] K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison, and M. A. M. Capretz, "Challenges for mapreduce in big data," in *2014 IEEE World Congress on Services*, 2014, pp. 182–189.

[151] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 13, no. 3, p. e0194889, mar 2018.

[152] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Communications Surveys Tutorials*, vol. 13, no. 3, pp. 443–461, 2011.

[153] G. Tuna and V. Gungor, "Energy harvesting and battery technologies for powering wireless sensor networks," in *Industrial Wireless Sensor Networks*. Elsevier, 2016, pp. 25–38.

[154] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer networks*, vol. 51, no. 4, pp. 921–960, 2007.

[155] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.

[156] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume, and A. Fehske, "How much energy is needed to run a wireless network?" *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40–49, October 2011.

[157] A. K. Trehan, "Energy conservation solutions for mobile networks," in *Intelec 2012*, Sep. 2012, pp. 1–5.

[158] L. M. Correia, D. Zeller, O. Blume, D. Ferling, Y. Jading, I. Gódor, G. Auer, and L. V. Der Perre, "Challenges and enabling technologies for energy aware mobile radio networks," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 66–72, November 2010.

[159] M. Sajedin, I. Elfergani, J. Rodriguez, R. Abd-Alhameed, and M. F. Barciela, “A survey on RF and microwave doherty power amplifier for mobile handset applications,” *Electronics*, vol. 8, no. 6, p. 717, jun 2019. [Online]. Available: https://doi.org/10.3390%2Felectronics8060717

[160] H. Wu, X. Xu, Y. Sun, and A. Li, “Energy efficient base station on/off with user association under C/U split,” in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, mar 2017.

[161] F. Han, S. Zhao, L. Zhang, and J. Wu, “Survey of strategies for switching off base stations in heterogeneous networks for greener 5G systems,” *IEEE Access*, vol. 4, pp. 4959–4973, 2016.

[162] B. Debaillie, C. Desset, and F. Louagie, “A flexible and future-proof power model for cellular base stations,” in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–7.

[163] 3GPP, “Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications,” 3rd Generation Partnership Project (3GPP), TR TR 21.905, June 2003.

[164] 3GPP, “5G; 5G System; Access and Mobility Management Services; Stage 3,” 3rd Generation Partnership Project (3GPP), TS TS 29.518, Oct. 2018.

[165] 3GPP, “5G; NR; Radio Resource Control (RRC); Protocol specification,” 3rd Generation Partnership Project (3GPP), TS TS 38.331, June 2018.

[166] A. Khlass, D. Laselva, and R. Jarvela, “On the flexible and performance-enhanced radio resource control for 5G NR networks,” in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, Sep. 2019, pp. 1–6.

[167] 3GPP, “5G; NR; User Equipment (UE) procedures in idle mode and in RRC Inactive state,” 3rd Generation Partnership Project (3GPP), TS TS 38.304, Oct. 2018.

[168] M. Tayyab, X. Gelabert, and R. Jäntti, “A survey on handover management: From LTE to NR,” *IEEE Access*, vol. 7, pp. 118 907–118 930, 2019.

[169] J. Sen, “Mobility and handoff management in wireless networks,” in *Trends in Telecommunications Technologies*. InTech, mar 2010. [Online]. Available: https://doi.org/10.5772%2F8482

[170] 3GPP, “5G; NR; Requirements for support of radio resource management,” 3rd Generation Partnership Project (3GPP), TS TS 38.133, Oct. 2018.

[171] X. Cheng, L. Fang, L. Yang, and S. Cui, "Mobile big data: The fuel for data-driven wireless," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1489–1516, Oct 2017.

[172] J. Wang, Y. Wu, N. Yen, S. Guo, and Z. Cheng, "Big data analytics for emergency communication networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1758–1778, thirdquarter 2016.

[173] Y. Liu, L. Kong, and G. Chen, "Data-oriented mobile crowdsensing: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2849–2885, thirdquarter 2019.

[174] X. Huang, S. Tang, Q. Zheng, D. Zhang, and Q. Chen, "Dynamic femtocell gNB on/off strategies and seamless dual connectivity in 5G heterogeneous cellular networks," *IEEE Access*, vol. 6, pp. 21 359–21 368, 2018.

[175] Z. Zhang, Z. Junhui, S. Ni, and Y. Gong, "A seamless handover scheme with assisted eNB for 5G C/U plane split heterogeneous network," *IEEE Access*, vol. 7, pp. 164 256–164 264, 2019.

[176] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, thirdquarter 2019.

[177] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.

[178] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: Multi-tier drone-cells," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–55, November 2016.

[179] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone networks: Communications, coordination, and sensing," *Ad Hoc Networks*, vol. 68, pp. 1 – 15, 2018, advances in Wireless Communication and Networking for Cooperating Autonomous Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870517301671

[180] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, Dec 2014.

[181] U. Challita, W. Saad, and C. Bettstetter, "Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.

[182] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, p. 2125–2140, Apr 2019. [Online]. Available: http://dx.doi.org/10.1109/TWC.2019.2900035

[183] ITU-R, "Definitions of Software Defined Radio (SDR) and Cognitive Radio System (CRS)," International Telecommunication Union, Spectrum Management Series SM.2152, Sept. 2009.

[184] I. F. Akyildiz, W. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 40–48, April 2008.

[185] FCC, "Facilitating Opportunities for Flexible, Efficient, and Reliable Spectrum Use Employing Cognitive Radio Technologies," Federal Communications Commission, Notice of Proposed Rulemaking ET Docket No. 03-108, Dec. 2003. [Online]. Available: https://www.fcc.gov/document/facilitating-opportunities-flexible-efficient-and-reliable-spectrum-1

[186] M. T. Masonta, M. Mzyece, and N. Ntlatlapa, "Spectrum decision in cognitive radio networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1088–1107, Third 2013.

[187] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127 – 2159, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128606001009

[188] S. Pandit and G. Singh, *Spectrum Sharing in Cognitive Radio Networks.* Springer International Publishing, 2017. [Online]. Available: https://doi.org/10.1007%2F978-3-319-53147-2

[189] I. Christian, S. Moh, I. Chung, and J. Lee, "Spectrum mobility in cognitive radio networks," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 114–121, June 2012.

[190] Y. Arjoune and N. Kaabouch, “A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions,” *Sensors*, vol. 19, no. 1, p. 126, jan 2019. [Online]. Available: https://doi.org/10.3390%2Fs19010126

[191] A. Ali and W. Hamouda, “Advances on spectrum sensing for cognitive radio networks: Theory and applications,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1277–1304, Secondquarter 2017.

[192] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, “Spectrum sensing for cognitive radio : State-of-the-art and recent advances,” *IEEE Signal Processing Magazine.*

[193] K. S. Kumar, R. Saravanan, and R. Muthaiah, “Cognitive radio spectrum sensing algorithms based on eigenvalue and covariance methods,” *Int. J. Eng. Technol*, vol. 5, no. 2, pp. 385–395, 2013.

[194] P. Dibal, E. Onwuka, J. Agajo, and C. Alenoghena, “Application of wavelet transform in spectrum sensing for cognitive radio: A survey,” *Physical Communication*, vol. 28, pp. 45 – 57, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874490717303300

[195] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, “Optimal multiband joint detection for spectrum sensing in cognitive radio networks,” *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1128–1140, March 2009.

[196] H. Sun, A. Nallanathan, C. Wang, and Y. Chen, “Wideband spectrum sensing for cognitive radio networks: a survey,” *IEEE Wireless Communications*, vol. 20, no. 2, pp. 74–81, April 2013.

[197] B. Farhang-Boroujeny, “Filter bank spectrum sensing for cognitive radios,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1801–1811, May 2008.

[198] F. Salahdine, N. Kaabouch, and H. E. Ghazi, “A survey on compressive sensing techniques for cognitive radio networks,” *Physical Communication*, vol. 20, pp. 61 – 73, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874490716300386

[199] E. J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb 2006.

[200] M. Rani, S. B. Dhok, and R. B. Deshmukh, "A systematic review of compressive sensing: Concepts, implementations and applications," *IEEE Access*, vol. 6, pp. 4875–4894, 2018.

[201] C. Yen, Y. Tsai, and X. Wang, "Wideband spectrum sensing based on subnyquist sampling," *IEEE Transactions on Signal Processing*, vol. 61, no. 12, pp. 3028–3040, June 2013.

[202] D. D. Ariananda, G. Leus, and Z. Tian, "Multi-coset sampling for power spectrum blind sensing," in *2011 17th International Conference on Digital Signal Processing (DSP)*, July 2011, pp. 1–8.

[203] A. M. Mikaeil, "Bayesian online learning-based spectrum occupancy prediction in cognitive radio networks," *ITU Journal: ICT Discoveries*, 2017.

[204] E. Chatziantoniou, "Spectrum sensing and occupancy prediction for cognitive machine-to-machine wireless networks," Ph.D. dissertation, University of Bedfordshire, 2014.

[205] S. Wang, F. Tosato, and J. P. Coon, "Reliable energy-efficient spectrum management and optimization in cognitive radio networks: How often should we switch?" *IEEE Wireless Communications*, vol. 20, no. 6, pp. 14–20, 2013.

[206] Z. Chen, N. Guo, Z. Hu, and R. C. Qiu, "Channel state prediction in cognitive radio, part II: Single-user prediction," in *2011 Proceedings of IEEE Southeastcon*, March 2011, pp. 50–54.

[207] M. Joneidi, I. Alkhouri, and N. Rahnavard, "Large-scale spectrum occupancy learning via tensor decomposition and lstm networks," *arXiv preprint arXiv:1905.04392*, 2019.

[208] M. Ozturk, M. Jaber, and M. A. Imran, "Energy-aware smart connectivity for IoT networks: Enabling smart ports," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11, jun 2018. [Online]. Available: https://doi.org/10.1155%2F2018%2F5379326

[209] M. Ozturk, A. I. Abubakar, R. N. B. Rais, M. Jaber, S. Hussain, and M. A. Imran, "Context-Aware Wireless Connectivity and Processing Unit Optimization for IoT Networks," *arXiv e-prints*, p. arXiv:2005.00407, Apr. 2020.

[210] S. Andreev, O. Galinina, A. Pyattaev, M. Gerasimenko, T. Tirronen, J. Torsner, J. Sachs, M. Dohler, and Y. Koucheryavy, "Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 32–40, 2015.

[211] S. Tayade, P. Rost, A. Maeder, and H. D. Schotten, "Device-centric energy optimization for edge cloud offloading," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[212] L. Atzori, A. Iera, and G. Morabito, "The Internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[213] N. Kouzayha, M. Jaber, and Z. Dawy, "Measurement-based signaling management strategies for cellular IoT," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1434–1444, 2017.

[214] Y. Yang, M. Zhong, H. Yao, F. Yu, X. Fu, and O. Postolache, "Internet of things for smart ports: Technologies and challenges," *IEEE Instrumentation & Measurement Magazine*, vol. 21, no. 1, pp. 34–43, 2018.

[215] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); LTE coverage enhancements," 3GPP, Thechnical Report 36.824 V11.0.0, Jun 2012.

[216] Keysight Technologies, "The menu at the IoT café: A guide to IoT wireless technologies," Application Note, Sep. 2017.

[217] L. Farhan, S. T. Shukur, A. E. Alissa, M. Alrweg, U. Raza, and R. Kharel, "A survey on the challenges and opportunities of the Internet of Things (IoT)," in *2017 Eleventh International Conference on Sensing Technology (ICST)*, Dec 2017, pp. 1–5.

[218] F. Renna, J. Doyle, V. Giotsas, and Y. Andreopoulos, "Media query processing for the Internet-of-things: Coupling of device energy consumption and cloud infrastructure billing," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2537–2552, 2016.

[219] S. Persia, C. Carciofi, and M. Faccioli, "NB-IoT and LoRA connectivity analysis for M2M/IoT smart grids applications," in *AEIT International Annual Conference, 2017*. IEEE, 2017, pp. 1–6.

[220] A. Mihovska and M. Sarkar, "Smart connectivity for Internet of Things (IoT) applications," in *New Advances in the Internet of Things*. Springer, 2018, pp. 105–118.

[221] N. Kouzayha, M. Jaber, and Z. Dawy, "M2M data aggregation over cellular networks: signaling-delay trade-offs," in *Globecom Workshops (GC Wkshps), 2014*. IEEE, 2014, pp. 1155–1160.

[222] J. Xu, L. Chen, and P. Zhou, "Joint Service Caching and Task Offloading for Mobile Edge Computing in Dense Networks," *arXiv e-prints*, p. arXiv:1801.05868, Jan. 2018.

[223] O. Y. Bursalioglu, Z. Li, C. Wang, and H. Papadopoulos, "Efficient C-RAN Random Access for IoT Devices: Learning Links via Recommendation Systems," *arXiv e-prints*, p. arXiv:1801.04001, Jan. 2018.

[224] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[225] E. Oyekanlu, "Predictive edge computing for time series of industrial IoT and large scale critical infrastructure based on open-source software analytic of big data," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1663–1669.

[226] S. Barbarossa, S. Sardellitti, E. Ceci, and M. Merluzzi, "The edge cloud: A holistic view of communication, computation and caching," *arXiv e-prints*, p. arXiv:1802.00700, Feb. 2018.

[227] T. X. Vu, S. Chatzinotas, and B. Ottersten, "Edge-caching wireless networks: Performance analysis and optimization," *IEEE Transactions on Wireless Communications*, 2018.

[228] E. M. Russek, I. Momennejad, M. M. Botvinick, S. J. Gershman, and N. D. Daw, "Predictive representations can link model-based reinforcement learning to model-free mechanisms," *PLOS Computational Biology*, vol. 13, no. 9, pp. 1–35, Sep. 2017.

[229] E. Even-Dar and Y. Mansour, "Convergence of optimistic and incremental Q-learning," in *Advances in neural information processing systems*, 2002, pp. 1499–1506.

[230] F. S. Melo, "Convergence of q-learning: A simple proof," *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

[231] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. [Online]. Available: https://doi.org/10.1007/BF0099269

[232] C. Luo and J. Liu, “Load based dynamic small cell on/off strategy in ultra-dense networks,” in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, oct 2018.

[233] Y. Sun and Y. Chang, “Energy-efficient user association and cell sleeping strategy for multi-tier ultra-dense small cell networks,” *Electronics Letters*, vol. 54, no. 12, pp. 787–789, jun 2018.

[234] J. Wu, J. Liu, and H. Zhao, “Dynamic small cell on/off control for green ultra-dense networks,” in *2016 8th international conference on wireless communications & signal processing (WCSP)*. IEEE, 2016, pp. 1–5.

[235] S. Buzzi, C. I, T. E. Klein, H. V. Poor, C. Yang, and A. Zappone, “A survey of energy-efficient techniques for 5G networks and challenges ahead,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 697–709, April 2016.

[236] A. Mohamed, O. Onireti, M. A. Imran, A. Imran, and R. Tafazolli, “Control-data separation architecture for cellular radio access networks: A survey and outlook,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 446–465, 2016.

[237] H. A. U. Mustafa, M. A. Imran, M. Z. Shakir, A. Imran, and R. Tafazolli, “Separation framework: An enabler for cooperative and D2D communication for future 5G networks,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 419–445, 2016.

[238] O. Onireti, A. Imran, M. A. Imran, and R. Tafazolli, “Energy efficient inter-frequency small cell discovery in heterogeneous networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7122–7135, 2015.

[239] A. Capone, I. Filippini, B. Gloss, and U. Barth, “Rethinking cellular system architecture for breaking current energy efficiency limits,” in *2012 Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE, 2012, pp. 1–5.

[240] H. Pervaiz, O. Onireti, A. Mohamed, M. A. Imran, R. Tafazolli, and Q. Ni, “Energy-efficient and load-proportional eNodeB for 5G user-centric networks: A multilevel sleep strategy mechanism,” *IEEE Vehicular Technology Magazine*, vol. 13, no. 4, pp. 51–59, 2018.

[241] H. Tabassum, M. Salehi, and E. Hossain, "Mobility-Aware Analysis of 5G and B5G Cellular Networks: A Tutorial," *arXiv e-prints*, p. arXiv:1805.02719, May 2018.

[242] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1790–1821, 2017.

[243] P.-Y. Kong and D. Panaitopol, "Reinforcement learning approach to dynamic activation of base station resources in wireless networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC).* IEEE, 2013, pp. 3264–3268.

[244] A. El Amine, M. Iturralde, H. A. H. Hassan, and L. Nuaymi, "A distributed Q-learning approach for adaptive sleep modes in 5G networks," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)(IEEE WCNC 2019*, 2019.

[245] F. E. Salem, Z. Altman, A. Gati, T. Chahed, and E. Altman, "Reinforcement learning approach for advanced sleep modes management in 5G networks," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall).* IEEE, 2018, pp. 1–5.

[246] A. El Amine, H. A. H. Hassan, M. Iturralde, and L. Nuaymi, "Location-aware sleep strategy for energy-delay tradeoffs in 5G with reinforcement learning," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC).* IEEE, Sep 2019.

[247] L. Busoniu, R. Babuska, and B. D. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, mar 2008.

[248] N. Saxena, B. J. R. Sahu, and Y. S. Han, "Traffic-aware energy optimization in green LTE cellular systems," *IEEE Communications Letters*, vol. 18, no. 1, pp. 38–41, January 2014.

[249] N. Saxena, A. Roy, and H. Kim, "Traffic-aware cloud RAN: A key for green 5G networks," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 1010–1021, April 2016.

[250] M. Ozturk, P. V. Klaine, and M. A. Imran, “3D transition matrix solution for a path dependency problem of Markov chains-based prediction in cellular networks,” in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1–5.

[251] M. Ozturk and P. V. Klaine and M. A. Imran, “Improvement on the performance of predictive handover management by setting a threshold,” in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Sep. 2017, pp. 1–5.

[252] M. Ozturk, P. V. Klaine, and M. A. Imran, “Introducing a novel minimum accuracy concept for predictive mobility management schemes,” in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[253] Li-Liann Lu, J. . C. Wu, and Wei-Yeh Chen, “The study of handoff prediction schemes for resource reservation in mobile multimedia wireless networks,” in *18th International Conference on Advanced Information Networking and Applications, 2004. AINA 2004.*, vol. 1, March 2004, pp. 379–384 Vol.1.

[254] Ming-Hsing Chiu and M. A. Bassiouni, “Predictive schemes for handoff prioritization in cellular networks based on mobile positioning,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 510–522, March 2000.

[255] Li-Liann Lu and J. . C. Wu, “Handoff prediction by mobility characteristics in wireless broadband networks,” in *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, June 2005, pp. 469–471.

[256] Q. Huang, S. Chan, and M. Zukerman, “Improving handoff QoS with or without mobility prediction,” *Electronics Letters*, vol. 43, no. 9, pp. 534–535, April 2007.

[257] P. Fazio and S. Marano, “Mobility prediction and resource reservation in cellular networks with distributed markov chains,” in *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2012, pp. 878–882.

[258] N. A. Amirrudin, S. H. S. Ariffin, N. N. N. A. Malik, and N. E. Ghazali, “User’s mobility history-based mobility prediction in lte femtocells net-

work," in *2013 IEEE International RF and Microwave Conference (RFM)*, Dec 2013, pp. 105–110.

[259] Y. Qiao, Z. Si, Y. Zhang, F. B. Abdesslem, X. Zhang, and J. Yang, "A hybrid Markov-based model for human mobility prediction," *Neurocomputing*, vol. 278, pp. 99–109, feb 2018. [Online]. Available: https://doi.org/10.1016%2Fj.neucom.2017.05.101

[260] H. Abu-Ghazaleh and A. S. Alfa, "Application of mobility prediction in wireless networks using Markov renewal theory," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 788–802, Feb 2010.

[261] P. Fazio and S. Marano, "A new Markov-based mobility prediction scheme for wireless networks with mobile hosts," in *2012 International Symposium on Performance Evaluation of Computer Telecommunication Systems (SPECTS)*, July 2012, pp. 1–5.

[262] A. Markov and N. Nagorny, "Algorithm theory," *Trudy Mat. Inst. Akad. Nauk SSSR*, vol. 42, pp. 1–376, 1954.

[263] M. Ruan, X. Chen, and H. Zhou, "Centrality prediction based on k-order Markov chain in mobile social networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 6, pp. 1662–1672, Nov 2019. [Online]. Available: https://doi.org/10.1007/s12083-019-00746-y

[264] X. Wang, X. Jiang, L. Chen, and Y. Wu, "KVLMM: A trajectory prediction method based on a variable-order Markov model with kernel smoothing," *IEEE Access*, vol. 6, pp. 25 200–25 208, 2018.

[265] Y. Sambo, P. V. Klaine, J. P. B. Nadas, and M. A. Imran, "Energy minimization UAV trajectory design for delay-tolerant emergency communication," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2019, pp. 1–6.

[266] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[267] F. Lagum, I. Bor-Yaliniz, and H. Yanikomeroglu, "Strategic densification with UAV-BSs in cellular networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 384–387, 2018.

[268] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Mobile internet of things: Can uavs provide an energy-efficient mobile architecture?” in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.

[269] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, “Beyond 5G with UAVs: Foundations of a 3D wireless cellular network,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 357–372, 2019.

[270] L. Wang, B. Hu, and S. Chen, “Energy efficient placement of a drone base station for minimum required transmit power,” *IEEE Wireless Commun. Lett.*, pp. 1–1, 2018.

[271] R. Ghanavi, E. Kalantari, M. Sabbaghian, H. Yanikomeroglu, and A. Yongacoglu, “Efficient 3D aerial base station placement considering users mobility by reinforcement learning,” in *Wireless Communications and Networking Conference (WCNC), 2018 IEEE.* IEEE, 2018, pp. 1–6.

[272] Q. Zhang, M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Machine learning for predictive on-demand deployment of UAVs for wireless communications,” in *2018 IEEE Global Communications Conference (GLOBECOM).* IEEE, 2018, pp. 1–6.

[273] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, “Modeling air-to-ground path loss for low altitude platforms in urban environments,” in *2014 IEEE Global Communications Conference.* IEEE, dec 2014.

[274] A. Shokry, M. Torki, and M. Youssef, “DeepLoc,” in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL '18.* ACM Press, 2018.

[275] M. Ozturk, M. Akram, S. Hussain, and M. A. Imran, “Novel QoS-aware proactive spectrum access techniques for cognitive radio using machine learning,” *IEEE Access*, vol. 7, pp. 70 811–70 827, 2019.

[276] G. Yang, J. Wang, J. Luo, O. Y. Wen, H. Li, Q. Li, and S. Li, “Cooperative spectrum sensing in heterogeneous cognitive radio networks based on normalized energy detection,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1452–1463, Mar. 2016.

[277] M. Hejazi and B. Abolhassani, “Cyclostationarity-based multi-antenna cooperative spectrum sensing in cognitive radio networks over correlated fading channels,” in *Proc. Iranian Conf. Electrical Engineering (ICEE)*, May 2018, pp. 627–632.

[278] S. Dannana, B. P. Chapa, and G. S. Rao, "Spectrum sensing for fading wireless channel using matched filter," in *Soft Computing for Problem Solving*. Springer, Jan. 2019.

[279] Y. Ma, Y. Gao, A. Cavallaro, C. G. Parini, W. Zhang, and Y. Liang, "Sparsity independent sub-Nyquist rate wideband spectrum sensing on real-time tv white space," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8784–8794, Oct. 2017.

[280] V. K. Tumuluru, P. Wang, and D. Niyato, "A neural network based spectrum prediction scheme for cognitive radio," in *Proc. IEEE Int. Conf. Communications*, May 2010, pp. 1–5.

[281] J. Yang and H. Zhao, "Enhanced throughput of cognitive radio networks by imperfect spectrum prediction," *IEEE Communications Letters*, vol. 19, no. 10, pp. 1738–1741, Oct. 2015.

[282] C. Park, S. Kim, S. Lim, and M. Song, "HMM based channel status predictor for cognitive radio," in *Proc. Asia-Pacific Microwave Conf*, Dec. 2007, pp. 1–4.

[283] A. Eltholth, "Forward backward autoregressive spectrum prediction scheme in cognitive radio systems," in *Proc. 9th Int. Conf. Signal Processing and Communication Systems (ICSPCS)*, Dec. 2015, pp. 1–5.

[284] S. Jain, A. Goel, and P. Arora, "Spectrum prediction using time delay neural network in cognitive radio network," in *Smart Innovations in Communication and Computational Sciences*. Springer, Jan. 2019.

[285] J. Yang and H. Zhao, "Genetic algorithm optimized training for neural network spectrum prediction," in *Proc. 2nd IEEE Int. Conf. Computer and Communications (ICCC)*, Oct. 2016, pp. 2949–2954.

[286] Z. Wen, T. Luo, W. Xiang, S. Majhi, and Y. Ma, "Autoregressive spectrum hole prediction model for cognitive radio systems," in *Proc. ICC Workshops - 2008 IEEE Int. Conf. Communications Workshops*, May 2008, pp. 154–157.

[287] Y. Zhao, Z. Hong, Y. Luo, G. Wang, and L. Pu, "Prediction-based spectrum management in cognitive radio networks," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3303–3314, Dec. 2018.

[288] X. Sun, G. Gui, Y. Li, R. P. Liu, and Y. An, "ResInNet: A novel deep neural network with feature reuse for internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 679–691, Feb. 2019.

[289] K. Mirylenka, V. Christophides, T. Palpanas, I. Pefkianakis, and M. May, "Characterizing home device usage from wireless traffic time series," in *Proc. of the 19th Int. Conf. on Extending Database Technology (EDBT)*, Mar. 2016, pp. 539–550.

[290] D. J. Ketchen and C. L. Shook, "The application of cluster analysis in strategic management research: An analysis and critique," *Strategic Management Journal*, vol. 17, no. 6, pp. 441–458, 6 1996.

[291] D. J. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, no. 3, pp. 415–447, 1992.

[292] F. Dan Foresee and M. T. Hagan, "Gauss-Newton approximation to Bayesian learning," in *Proc. Int. Conf. Neural Networks (ICNN'97)*, vol. 3, Jun. 1997, pp. 1930–1935 vol.3.

[293] J. Heaton, *Introduction to neural networks with Java.* Heaton Research, Inc., 2008.

[294] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. Upper Saddle River, NJ: Pearson Education, 2009.

[295] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based IoT: Challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, January 2017.

[296] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green IoT: An investigation on energy saving practices for 2020 and beyond," *IEEE Access*, vol. 5, pp. 15 667–15 681, 2017.

[297] W. K. A. Hasan, Y. Ran, J. Agbinya, and G. Tian, "A survey of energy efficient IoT network in cloud environment," in *2019 Cybersecurity and Cyberforensics Conference (CCC)*, May 2019, pp. 13–21.

[298] A. Alnoman, G. H. S. Carvalho, A. Anpalagan, and I. Woungang, "Energy efficiency on fully cloudified mobile networks: Survey, challenges, and open issues," *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1271–1291, Secondquarter 2018.

[299] Z. T. Al-Azez, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Virtualization framework for energy efficient IoT networks," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, Oct 2015, pp. 74–77.

[300] L. Chettri and R. Bera, "A comprehensive survey on Internet of things (IoT) toward 5G wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, Jan 2020.

[301] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, and H. Tenhunen, "Using ant colony system to consolidate VMs for green cloud computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187–198, March 2015.

[302] X. Liu and N. Ansari, "Toward green IoT: Energy solutions and key challenges," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 104–110, March 2019.

[303] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE network*, vol. 28, no. 6, pp. 27–33, 2014.

[304] S. Gong, X. Lu, D. T. Hoang, D. Niyato, L. Shu, D. In Kim, and Y.-C. Liang, "Towards Smart Radio Environment for Wireless Communications via Intelligent Reflecting Surfaces: A Comprehensive Survey," *arXiv e-prints*, p. arXiv:1912.07794, Dec. 2019.

[305] E. Björnson and L. Sanguinetti, "Demystifying the Power Scaling Law of Intelligent Reflecting Surfaces and Metasurfaces," *arXiv e-prints*, p. arXiv:1908.03133, Aug. 2019.

[306] O. Özdogan, E. Björnson, and E. G. Larsson, "Intelligent reflecting surfaces: Physics, propagation, and pathloss modeling," *IEEE Wireless Communications Letters*, pp. 1–1, 2019.

[307] K. Kotobi and S. G. Bilen, "Secure blockchains for dynamic spectrum access: A decentralized database in moving cognitive radio networks enhances security and user access," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 32–39, March 2018.

[308] K. Kotobi and S. G. Bilén, "Blockchain-enabled spectrum access in cognitive radio networks," in *2017 Wireless Telecommunications Symposium (WTS)*, April 2017, pp. 1–6.

[309] S. Raju, S. Boddepalli, S. Gampa, Q. Yan, and J. S. Deogun, "Identity management using blockchain for cognitive cellular networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[310] M. B. H. Weiss, K. Werbach, D. C. Sicker, and C. E. C. Bastidas, "On the application of blockchains to spectrum management," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 193–205, June 2019.

[311] W. Ni, Y. Zhang, and W. Li, "Optimal admission control for secondary users using blockchain technology in cognitive radio networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, July 2019, pp. 1518–1526.

[312] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395 – 411, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17315765

[313] A. P. Joshi, "A survey on security and privacy issues of blockchain technology," p. 121, 2018. [Online]. Available: http://aimsciences.org//article/id/d27803a2-7ce7-46e8-900d-30001fd4785a