Tan, Kang (2023) *Adaptive vehicular networking with Deep Learning.* PhD thesis

http://theses.gla.ac.uk/83558/

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses
https://theses.gla.ac.uk/
research-enlighten@glasgow.ac.uk

Adaptive Vehicular Networking with Deep Learning

Kang Tan

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

School of Engineering
College of Science and Engineering
University of Glasgow

February 2023

Abstract

Vehicular networks have been identified as a key enabler for future smart traffic applications aiming to improve on-road safety, increase road traffic efficiency, or provide advanced infotainment services to improve on-board comfort. However, the requirements of smart traffic applications also place demands on vehicular networks' quality in terms of high data rates, low latency, and reliability, while simultaneously meeting the challenges of sustainability, green network development goals and energy efficiency. The advances in vehicular communication technologies combined with the peculiar characteristics of vehicular networks have brought challenges to traditional networking solutions designed around fixed parameters using complex mathematical optimisation. These challenges necessitate greater intelligence to be embedded in vehicular networks to realise adaptive network optimisation. As such, one promising solution is the use of Machine Learning (ML) algorithms to extract hidden patterns from collected data thus formulating adaptive network optimisation solutions with strong generalisation capabilities

In this thesis, an overview of the underlying technologies, applications, and characteristics of vehicular networks is presented, followed by the motivation of using ML and a general introduction of ML background. Additionally, a literature review of ML applications in vehicular networks is also presented drawing on the state-of-the-art of ML technology adoption. Three key challenging research topics have been identified centred around network optimisation and ML deployment aspects.

The first research question and contribution focus on mobile Handover (HO) optimisation as vehicles pass between base stations; a Deep Reinforcement Learning (DRL) handover algorithm is proposed and evaluated against the currently deployed method. Simulation results suggest that the proposed algorithm can guarantee optimal HO decision in a realistic simulation setup.

The second contribution explores distributed radio resource management optimisation. Two versions of a Federated Learning (FL) enhanced DRL algorithm are proposed and evaluated against other state-of-the-art ML solutions. Simulation results suggest that

the proposed solution outperformed other benchmarks in overall resource utilisation efficiency, especially in generalisation scenarios.

The third contribution looks at energy efficiency optimisation on the network side considering a backdrop of sustainability and green networking. A cell switching algorithm was developed based on a Graph Neural Network (GNN) model and the proposed energy efficiency scheme is able to achieve almost 95% of the metric normalised energy efficiency compared against the "ideal" optimal energy efficiency benchmark and is capable of being applied in many more general network configurations compared with the state-of-the-art ML benchmark.

University of Glasgow

College of Science & Engineering

Statement of Originality

Name: Kang Tan

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 the review of the thesis, the examination may be postponed pending the outcome of any investigation of the issues.

Signature: .....

Date: .....

Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

3GPP	3rd Generation Partnership Project.
4G	Fourth Generation.
5G	Fifth Generation.
A2C	Advantage Actor-critic.
AC	Alternating Current.
AdamW	Adam with Decoupled Weight Decay.
AI	Artificial Intelligence.
AMF	Access and Mobility Management Function.
ANN	Artificial Neural Network.
API	Application Programming Interface.
BS	Base Station.
CDR	Call Detail Records.
CDSA	Control Data Separated Architecture.
CNN	Convolutional Neural Network.
CSI	Channel State Information.
C-V2X	Cellular Vehicle-to-everything.
CO_2	Carbon Dioxide.
DC	Direct Current.
DDPG	Deep Deterministic Policy Gradient.
DDQN	Double Deep Q-network.
DL	Deep Learning.
DQN	Deep Q-network.
DRL	Deep Reinforcement Learning.
DSRC	Dedicated Short-range Communication.
ES	Exhaustive Search.

E-UTRAN	Evolved UMTS Terrestrial Radio Access Network.
EARTH	Energy Aware Radio and neTwork tecHnologies.
FL	Federated Learning.
GBCSS	GNN-based Cell-switching Solution.
GCN	Graph Convolutional Network.
GDP	Gross Domestic Product.
GELU	Gaussian Error Linear Units.
GNN	Graph Neural Network.
HetNet	Heterogeneous Network.
HMM	Hidden Markov Model.
HO	Handover.
ICT	Information and Communications Technology.
ITS	Intelligent Transportation Systems.
KNN	K-nearest Neighbours.
KPI	Key Performance Indicator.
LSTM	Long Short-time Memory.
LTE	Long-term Evolution.
LTE-A	Long-term Evolution Advanced.
MARL	Multi-agent Reinforcement Learning.
MC	Macro Cell.
MDP	Markov Decision Process.
MEC	Mobile Edge Computing.
ML	Machine Learning.
mmWave	Millimetre Wave.
MNIST	Modified National Institute of Standards and Technology.
MSE	Mean Squared Error.
NFV	Network Function Virtualisation.
OpEx	Operational Expenses.

PDCP	Packet Data Convergence Protocol.
PTDQ	Post-training Dynamic Quantisation.
QoS	Quality of Service.
RAN	Radio Access Networks.
RAT	Radio Access Technology.
ReLU	Rectified Linear Unit.
RL	Reinforcement Learning.
RNN	Recurrent Neural Network.
RRC	Radio Resource Control.
RSRP	Reference Signal Received Power.
RSRQ	Reference Signal Received Quality.
RSSI	Received Signal Strength Indication.
RSU	Roadside Unit.
SC	Small Cell.
SINR	Signal-to-Interference-plus-Noise Ratio.
SMS	Short Message Service.
SNR	Signal-to-Noise Ratio.
SVM	Support-vector Machine.
TTT	Time-To-Trigger.
UAV	Unmanned Aerial Vehicle.
UDP	User Datagram Protocol.
UE	User Equipment.
UMTS	Universal Mobile Telecommunications System.
UTRAN	UMTS Terrestrial Radio Access Network.
V2I	Vehicle-to-infrastructure.
V2N	Vehicle-to-network.
V2P	Vehicle-to-pedestrian.
V2V	Vehicle-to-vehicle.
V2X	Vehicle-to-everything.
VUE	Vehicular User Equipment.

List of Symbols

a	An action in reinforcement learning.
s	A state observation in reinforcement learning.
r	The reward in reinforcement learning.
$\pi(a\|s)$	The policy in reinforcement learning.
G_t	The accumulated return at time step t in reinforcement learning.
γ	The discount factor in reinforcement learning.
$V_\pi(s)$	The state value function in reinforcement learning.
$Q_\pi(s, a)$	The state-action value (Q-value) function in reinforcement learning.
α	The step-size parameter of Q-learning in reinforcement learning.
l	A layer of an artificial neural network.
$\bar{x}$	The input of a neural network in deep learning.
$\bar{y}$	The output of a neural network in deep learning.
$\phi(\cdot)$	An activation function of a neural network's layer in deep learning.
W_l	The weights of layer l's neurons of a neural network in deep learning.
b_l	The bias of layer l of a neural network in deep learning.
$L(\theta)$	The loss function for neural network θ.
ρ	The the behaviour distribution in reinforcement learning.
θ_a	The actor-network used in advantage actor-critic algorithm.
θ_c	The critic-network used in advantage actor-critic algorithm.

$A_(s, a)$	The advantage function in reinforcement learning.
g	A graph (data structure).
$\mathcal{V}_g$	The set of vertexes/nodes of graph g.
$\mathcal{E}_g$	The set of edges/arcs of graph g.
$\mathcal{X}_g$	The set of node features/attributes of graph g.
$\mathcal{A}_g$	The set of edge features/attributes of graph g.
x_v	The feature/attribute of node v of graph g.
e_{uv}	An edge connecting node pair (u, v) of graph g.
a_{uv}	The feature/attribute of edge e_{uv} of graph g.
N_{rb}	The number of resource blocks over the carrier RSSI measurement bandwidth.
N_{sb}	The number of sub-bands for the spectrum selection problem.
P_m	The transmit power of the m-th transmitter.
$g_{m,n}$	The fading channel gain from the transmit channel m to n, m may equal to n.
σ^2	The noise power.
P_{max}	The maximum transmit power (of a transmitter or a base station).
N_{TRX}	The number of transceiver chains of a cellular base station.
P_{in}	The total power consumption of a cellular base station.
η_{PA}	The power amplifier's power efficiency of a cellular base station.
σ_{feed}	The coupling losses from the feeders of a cellular base station.
σ_{DC}	The coupling losses from the DC-DC converter of a cellular base station.
σ_{MS}	The coupling losses from the mains supply of a cellular base station.
σ_{cool}	The coupling losses from the cooling systems of a cellular base station.
P_{out}	The average transmit power for each transceiver link of a cellular base station.
P_{RF}	The power consumptions of the radio frequency unit of a cellular base station.
P_{BB}	The power consumption of the baseband unit of a cellular base station.

P_0	The static operational power consumption of a cellular base station.
P_s	The sleeping power consumption of a cellular base station.
Δ_p	The slope of the load-dependent power consumption for a cellular base station.
t_0	The optimal point for handover.
Δ	The handover parameter: hysteresis.
TTT	The handover parameter: time-to-trigger.
δ_t	The delayed time to trigger a handover with respect to t_0.
t_d	The handover interruption time.
N_{ho}	The number of handovers.
D_{HO}	The handover cost.
β_{HO}	The normalised handover delay cost for a unit time.
F_n	The RSRP value after layer-3 filtering.
M_n	The raw measurement of RSRP value from the physical layer.
S	The state space in reinforcement learning.
A	The action space in reinforcement learning.
$\mathbb{B}$	The replay buffer for deep Q-network algorithm.
N_{rp}	The replay buffer capacity.
N_{mb}	The mini-batch size.
N_f	Target network update frequency for the double deep Q-network algorithm.
ϵ	The probability to take a random action for ϵ-greedy.
$N_{episode}$	The number of episode for RL training.
$\mathbb{G}_{SINR}$	The defined metric "SINR gain" used in Chapter 4.
$\mathbb{G}_{packet}$	The defined metric "throughput gain" used in Chapter 4.
$\mathbb{G}_{\delta_t}$	The defined metric "gain in delayed time for handover" used in Chapter 4.
$\mathcal{J}$	The set of V2N links for the scenario of Chapter 5.
$\mathcal{K}$	The set of V2V links for the scenario of Chapter 5.

$\mathcal{M}$	The set of orthogonal spectrum sub-bands for the scenario of Chapter 5.
$\rho_{j,m}, \rho_{k,m}$	The sub-band selection decisions of the j-th Vehicle-to-network (V2N) link and k-th Vehicle-to-vehicle (V2V) link for sub-band m for the scenario in Chapter 5.
$h_{j,\bar{m}_j}$	The channel gains for the j-th V2N link over sub-band $\bar{m}_j$ for the scenario in Chapter 5.
$g_{k,m}$	The channel gains for the k-th V2V link over sub-band m for the scenario in Chapter 5.
$I_{j,\bar{m}_j}$	The interference to the j-th V2N link over sub-band $\bar{m}_j$ for the scenario in Chapter 5.
$I_{k,m}$	The interference to the k-th V2V link over sub-band m for the scenario in Chapter 5.
p^{v2n}	The transmit power of V2N links in Chapter 5.
p^{v2v}	The transmit power of V2V links in Chapter 5.
C_j^{v2n}	The data rate of V2N link j in Chapter 5.
C_k^{v2v}	The data rate of V2V link k in Chapter 5.
$\zeta_{k,u}$	The binary flag representing the successful transmission of V2V payload u for the k-th V2V link is represented in Chapter 5.
B_u	The size of packet u in Chapter 5.
t_{max}	The is the maximum delay tolerance for V2V payload transmission in Chapter 5.
η_k	The successful transmission probability of V2V link k in Chapter 5.
$N_{payload}$	The number of V2V payloads in Chapter 5.
E	The metric "system-level V2V payload delivery ratio" in Chapter 5.
Λ	The metric "system-level V2N data rate" in Chapter 5.
N_{pow}	The number of power levels for V2V transmit power in Chapter 5.
ΔB	The average transmission rate for a given V2V payload in Chapter 5.
N_{agent}	The number of participants in federated learning.

$\mathbb{M}$	Memory for n-step temporal difference calculation in reinforcement learning.
θ_g	The global model of federated learning.
N_{agg}	The step size for global model aggregation in federated learning.
$\mathbb{G}$	The metric "aggregated gain" in Chapter 5.
P_{tot}	The power consumption of a HetNet unit in Chapter 6.
N_{SC}	The number of small cell base stations in a HetNet unit in Chapter 6.
P_{BS}^{i}	The power consumption base stations i in a HetNet unit in Chapter 6.
P_{MC}	The power consumption of the macro cell in a HetNet unit in Chapter 6.
P_{SC}^{i}	The power consumption of the i-th small cell in a HetNet unit in Chapter 6.
P_{out}^{i}	The transmit power of the i-th base station in a HetNet unit in Chapter 6.
P_{max}^{i}	The maximum transmit power of the i-th base station in a HetNet unit in Chapter 6.
λ_t^i	The factorised traffic load of the i-th base station of a HetNet unit at time t in Chapter 6.
d_t^i	The radio resources of base station i utilised by its served users at time t.
C^i	The radio resource capacity of base station i in Chapter 6.
Γ_t	The cell switching strategy set for all considered base stations at time t in Chapter 6.
γ_t^i	The cell switching strategy of base station i at time t in Chapter 6.
ϕ_i	The ratio of base station i capacity to that of the macro cell within a HetNet unit in Chapter 6.
Λ_t	The factorised sum traffic load of a HetNet unit at time t in Chapter 6.
$\hat{\lambda}_t^i$	The factorised traffic load of the i-th base station of a HetNet unit at time t after cell switching in Chapter 6.

$\hat{P}^i_{BS}$	The power consumption base stations i in a HetNet unit after cell switching in Chapter 6.
$\hat{\Lambda}_t$	The factorised sum traffic load of a HetNet unit at time t after cell switching in Chapter 6.
$\hat{P}_{tot}$	The power consumption of a HetNet unit after cell switching in Chapter 6.
$\mu_l\langle\cdot\rangle$	A parametric combination function for graph neural networks.
$\psi_l\langle\cdot\rangle$	The activation function of layer l.
Θ^l_i	The i-th learnable parameter of the LEConv operator for layer l.
$\Psi\langle\cdot\rangle$	The activation function of the final layer of the graph neural network that maps the graph embedding values to the continuous values ranging between [0, 1] in Chapter 6.
$I_{[0.5,1]}\langle\cdot\rangle$	The indicator function of event "between 0.5 and 1".
$\zeta_{i,g}(\cdot)$	The objective function for the j-th sample of graph g in a given data batch.
$\sigma\langle\cdot\rangle$	The sigmoid activation function.
E_{saving}	The metric "total energy saved" used in Chapter 6.
$\Lambda_{\%}$	The metric "normalised network traffic load" used in Chapter 6.
$\eta_{\%}$	The metric "normalised energy efficiency" used in Chapter 6.

Acknowledgements

"I always tried to take the fastest shortcut in this journey, but the shortest route was a detour. It was the detour that was the shortest path. It has been true the whole time and because of you, I was able to take this route." There are many others on this journey, without whom this thesis would never be accomplished. I would like to express my gratitude to all of them for their support, help, understanding and accompany.

First and foremost, I would like to convey my sincere appreciation to my supervision team: Dr. Duncan Bremner, Dr. Julien Le Kernec, Prof. Muhammad Ali Imran, and the later joined Dr. Yusuf Sambo. Thank you all for your support and never-ending guidance throughout this project from project planning, and management to writing and presentation. Specifically, I would like to thank Dr. Duncan Brenmner and Dr. Julien Le Kernec for the "suffering" they cast upon me to sharpen my mind and reach for a higher level of thinking, and Dr. Yusuf Sambo and Prof. Muhammad Ali Imran for the valuable insights and heartwarming encouragement. All of these have helped me towards a better researcher.

Secondly, I would like to express my gratitude to Dr. Lei Zhang, Dr. Paulo Henrique Valente Klaine, Dr. Shuja Ansari, and Dr. Metin Öztürk. Although not part of my supervision team, they have provided their kind support and valuable discussions in project planning, practical skills, and writing, which greatly helped me in my research.

I would also like to thank my dear friends: Rongxiao Fu, Huihua Cheng, Shunxin Wan, and many others, for their accompany throughout this journey. And my girlfriend Yutong Wu, I thank her for her love, understanding, and companionship since the very beginning of this journey. I believe that the darkest hour is just before dawn for all of you currently in a difficult time.

Last but not least, I am extremely grateful to my dear family, especially my parents: Jun Tan and Xiangrong Zhao. All of this would never happen without your endless support and trust in me and I am always proud to be your son.

Chapter 1

Introduction

Road transportation is part of the indispensable foundation of modern human activities. It accounts for up to 12% of the Gross Domestic Product (GDP) and 15% of average household expenditures in many developed countries, with people spending about 8% or approximately two hours of their workday, commuting to and from work [1]. However, with the rapid growth in the number of vehicles over the last half a century and 60 million car sales annually (2005 to 2020) [2]. This has led to the saturation of road infrastructures especially in urban areas, causing traffic congestion, delays, and road accidents that now raise serious socioeconomic issues. Traffic congestion and delays cause work efficiency reduction and an increase in air pollution, while road accidents lead to injuries and even death [3, 4].

The World Health Organisation statistics show that road traffic crashes cause 1.35 million lives lost every year, with another 20 to 50 million people suffering non-fatal injuries and disabilities because of these accidents [5]. Road traffic accidents are now the 8^{th} largest cause of death for people of all ages and the number one cause of death for children and young adults [6]. Moreover, the overall economic impact of road crashes was estimated to be $518 billion globally, accounting for up to 5% of the GDP in some countries [7].

In addition to traffic accidents, the congestion aspect of road traffic is also giving rise to concerns as the median annual time loss in over 400 cities worldwide was found to be 48 hours, with the most congested cities such as Istanbul losing up to 142 working hours annually due to road congestion[8]. This congestion not only affects the mental health of the road traffic commuters but also brings an unquantified economical loss due to its impact on the overall work efficiency of the society [9, 10].

From a global climatic standpoint, road transport is a major contributor to climate

change accounting for up to two thirds (65%) of the Carbon Dioxide (CO_2) emissions for the whole transportation sector in the first decade of 21^{th} century [11]. Such statistics are worsened when the average travelling speed is reduced by traffic congestion, further exacerbated by the stop-and-go pattern of road vehicles thus increasing vehicles' energy consumption and hence the emission of CO_2 and contributing to climate change [12].

All these challenges have motivated governments and motor manufacturers to explore smart transportation systems to increase road traffic safety, improve efficiency, and reduce the impact of the above statistics on their economies using intelligent traffic management applications such as adaptive traffic-signal control, traffic route planning, and driving automation [1]. Furthermore, the development of road traffic systems also introduces the opportunity for new service requirements to boost onboard comfort and infotainment experiences leading to new applications such as real-time navigation updates, high-resolution video streaming, online gaming, etc [13]. To support these applications, it becomes essential to utilise use case-dependent information distributed in the transportation system combined with transport-centric cloud servers, necessitating the sharing of information from various data sources such as vehicles, road traffic control infrastructure, and street-side furniture [14].

1.1 Context

As onboard sensors and sensing equipment become more comprehensive, vehicles now have improved awareness of the surrounding environment and thus become an important information source. As a result, vehicular networks are now critical to future smart transportation by integrating road vehicles with other communication entities such as traffic lights and mobile phones for useful information exchange [7]. Note that vehicular networks refer to networks among vehicles and between vehicles and external devices in this thesis, excluding intra-vehicle networks. Vehicular networks consist of various communication types such as V2V, Vehicle-to-infrastructure (V2I), V2N, or Vehicle-to-pedestrian (V2P) communications, based on the target entity to communicate. Together, these communication types are referred to as Vehicle-to-everything (V2X) communications exemplified in Figure 1.1.

The figure shows exemplar applications enabled by different V2X communication links, such as the safety-related applications of collision avoidance messaging among vehicles via V2V and safety alerts from vehicles to pedestrians through V2P. Non-safety applications can be classified as traffic utility or infotainment applications, consisting of real-time road traffic monitoring and smart traffic light control enabled by V2I between vehicles and road infrastructure such as traffic lights, and on-board access to cloud-based services (e.g. video streaming) from a BS or access point via V2N, respectively [15, 16].

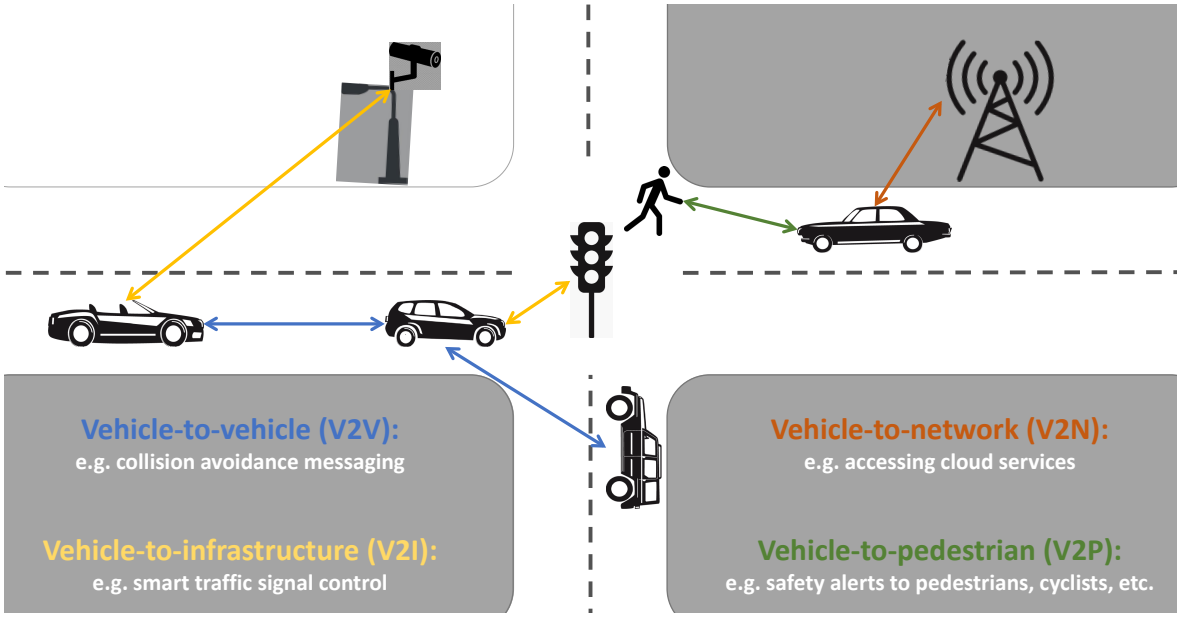


Figure 1.1: A demonstrative V2X scenario with different communication types. Other communication types that do not contain vehicles are omitted, e.g., infrastructure-to-pedestrian and pedestrian-to-network.

These connections predominantly rely on wireless technology which is managed by the underlying physical connection method termed Radio Access Technologys (RATs). For vehicular networks, the Dedicated Short-range Communication (DSRC) protocol has been considered the predominant RAT protocol since 2010 providing highly reliable and secure communication links focusing on safety applications [17]. However, DSRC has some major drawbacks such as its short transmission distance of 300 m, the requirement for dedicated roadside units, and limited radio bandwidth of 75 MHz [18]. With the evolution of cellular network standardisation, Cellular Vehicle-to-everything (C-V2X) was introduced in the cellular Long-term Evolution (LTE) standards with major advantages over DSRC such as providing wider coverage and higher capacity, and utilisation of existing cellular infrastructure [18, 19]. Despite these advantages, C-V2X also has some drawbacks compared with DSRC, such as higher costs for consumers and the already heavy traffic loads that may affect V2X service delivery. As the Fifth Generation (5G) of cellular networks roll out globally, the better designed 5G C-V2X has gained increased attention from both academia and industry, 5G will probably be the mainstay technology to implement V2X communication networks [20–23]. An example of improvement by 5G can be shown by a recent research's evaluation that 5G C-V2X can improve the reliability by 20% for resource allocation over its predecessor with other improvements [24]. As a result, C-V2X is selected as the underlying protocol for the research work of this thesis.

In contrast to existing mobile phone connections, a defining characteristic of vehicular networks is the vehicles' fast-moving speed and thus continuous position changes. This leads to short connection time between the vehicle node and its paired connection link (such as a static-positioned BS and another vehicle node), requiring frequent yet reliable

establishment and termination of these communication links. Additionally, vehicular network environments also contain multiple environmental scenarios such as motorways and urban street blocks that contain specific radio environment characteristics (e.g. losses, reflections, and fading) and vehicular mobility patterns. These environments may also encounter radio path loss uncertainty due to temporary obstacles appearing and blocking possible routing paths, e.g., a dense metal truck passing between two civilian cars who are communicating via V2V, further complicating the communication link management in vehicular networks.

V2X communication networks, with their corresponding applications, have stringent and differentiated Quality of Service (QoS) requirements. For instance, the essential safety messaging utilising V2V demands ultra-reliability and very low latency, while infotainment services that access the Internet via V2N require high data rates and large bandwidths [25]. With the development of vehicular networks, the number of vehicles with built-in connectivity in operation reached 237 million in 2021 and it is estimated that connected vehicles will take up 96% of the newly shipped vehicles in 2030 [26]. This huge increase in the number of connected vehicles and the heterogeneous application requirements will bring new challenges to the management of vehicular networks to efficiently utilise the limited radio spectrum resources and network infrastructure.

In addition to network performance, it is also important for vehicular networks to be developed in an environmentally and ecologically friendly manner while still being cost-effective from an economic standpoint. Presently, the Information and Communications Technology (ICT) sector is a major contributor to climate change and is estimated to share 2.1% to 3.9% of global greenhouse gas emissions due to its energy consumption [27]. As a result, the sector must reduce its CO_2 emissions by 42% by 2030 and 72% by 2040 in line with other sectors to stay within the 1.5°C global goal [27]. Meanwhile, the economic burden of energy cost constitutes between 10%-15% and 50% of the total cellular network Operational Expenses (OpEx) in mature and developing markets respectively [28]; developing markets have a much higher energy cost due to the proliferation of off-grid sites [29]. As part of the "5G and beyond" cellular network strategy, it is important for C-V2X networks to be energy efficient to meet the sustainability requirement as 5G is designed with the goal of being 10 times more energy efficient than Fourth Generation (4G), its predecessor, while even more stringent requirement is envisioned for beyond 5G [30].

1.2 Motivation and Objectives

To achieve the goal of road safety and road traffic efficiency, vehicular networks will be pivotal in supporting and deploying essential safety applications as described in Figure 1.1 and utility-oriented non-safety applications such as autonomous driving, remote

driving, and intelligent traffic management [31]. Moreover, the goal of improved onboard entertainment for comfort requires vehicular networks to also support futuristic onboard infotainment services such as 4K video streaming and 3D augmented reality. Together, these applications require very high throughput, scalability, and reliability with ultra-low latency [32]. Combining these emerging applications onto the requirements for C-V2X use cases, they can be summarised as stringent Key Performance Indicators (KPIs) include the reliability of 99.999%, a maximum end-to-end latency of 3 ms, and a peak data rate of 1000 Mb/s [33], with an estimated tenfold performance increase envisioned for technologies beyond 5G [30], all demanding seamless connectivity and high spectrum efficiency.

To boost network performance, several new technologies have been proposed in 5G and C-V2X systems to meet the QoS requirements and address near-future challenges. For example, as the current cellular network capacity reaches its bottleneck, network densification (i.e. more base stations per km^2) becomes a solution to enhance cellular network capacity for V2N that also subsequently improves the spectral utilisation [34]. The principle of network densification is to deploy compact, low-power BSs, known as Small Cell (SC), within the coverage area of a conventional Macro Cell (MC) thus bringing the BSs closer to the users and hence able to provide higher data rate. The coverage areas (footprints) of SCs are much smaller when compared with a conventional MC because of their low transmit power, which permits frequency reuse for a group of SCs and hence improving spectrum utilisation efficiency [35]. The concept of network densification is shown in Figure 1.2 with a demonstrative frequency reuse scheme.

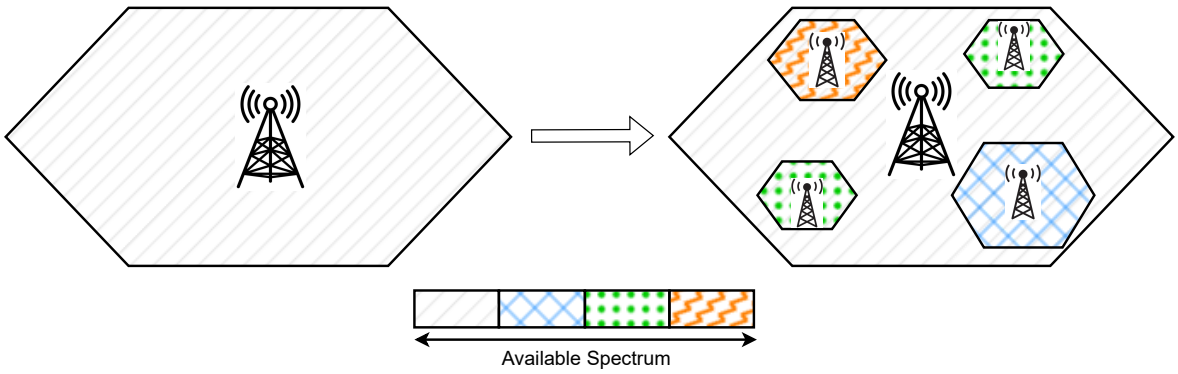


Figure 1.2: Concept of network densification with an exemplar frequency reuse scheme.

Due to vehicular networks' highly mobile and dynamic nature, complex environments, and the heterogeneous QoS requirements for different V2X applications have introduced new challenges to achieve successful vehicular networking. These demands must be combined with the significant drive towards OpEx and greenhouse gas emission reduction against a global backdrop of energy efficiency [36]. This thesis has identified the following key challenging research topics for the development of effective and efficient cellular vehicular networks:

- Handover decision optimisation
- Radio resource management optimisation
- Energy efficiency optimisation at the network side

Besides these chosen topics, there are also other essential and challenging research topics identified by the research community that require much attention for V2X networks, such as channel modelling, routing, data dissemination, security issues [19, 36–38], to name a few. Nevertheless, the three research topics were chosen as they are considered to be directly linked to the aforementioned seamless connectivity, high performance and reliability, and energy efficiency requirements for future V2X development from the networking perspective, which is the main standpoint of this thesis.

1.2.1 Motivation of Research Topics

The first research question explores handover decision optimisation. Vehicular networks' characteristics due to high mobility lead to much more frequent HO (when a vehicle changes from its serving BS (BS_1) to a new BS (BS_2) during an active connection) for V2N links compared to conventional cellular networks. When these events are combined with the complex environmental scenarios found around vehicular communications it adds to the complexity of vehicular networks' HO management. This becomes even more obvious with increased network densification despite its capacity enhancement benefits. As SCs are low power, are deployed in bulk, and have limited range coverage, network densification further exacerbates the frequency of V2N HO while simultaneously increasing the possible target BSs for HO. This now complicates the search space for HO decision-making. Therefore, efficient HO management mitigating all these is crucial to ensure smooth connection transitions while supporting uninterrupted high data rates for various applications [17].

The second research topic is radio resource management which is the system-level management of interference, radio frequency bands, and other radio transmission properties of wireless communication systems [39]. For vehicular networks, the stringent QoS requirement of V2X applications necessitates meticulous spectrum and transmit power management for improved spectrum efficiency and interference management. The significant increase in connected vehicles further contributes to this challenge by introducing more users to the already congested radio frequency bands. Moreover, the highly dynamic nature of vehicular networks also adds to this challenge of radio resource management by introducing rapid temporal variation in vehicular wireless channels [40], making tasks such as radio channel estimation and signal detection for resource allocation in vehicular networks much more challenging than traditional cellular networks [41]. Therefore, it is essential for vehicular networks to have adaptive and

optimal spectrum and power allocation to improve resource efficiency and guarantee QoS delivery for various V2X applications.

The third research topic is energy efficiency optimisation. C-V2X has the goal of efficient use of energy in line with that of “5G and beyond" strategy [17, 32]. From a network perspective, energy saving for BSs is an essential aspect of energy efficiency improvement as BSs are the major consumers of electrical power in both conventional cellular networks and the emerging 5GV2N communications, accounting for 60% to 80% of the network's total power consumption [42]. With the deployment of 5G, the total energy consumption of 5G BSs will increase by almost 70% compared to the present generation [43, 44], further raising the energy cost and overall OpEx for network operators. In addition, while individual SCs have much smaller power consumption due to their reduced transmission power and numbers of antenna sectors compared to a traditional MC [45], the power consumption accumulation by the ultra-dense deployment of SCs leads to significant energy usage. As a result, network densification also leads to an increase in energy consumption. Due to these issues in conjunction with the CO_2 emission goal of the ICT sector, greater attention should be paid to the energy efficiency of both vehicular networks and general cellular networks to reduce OpEx cost-effectively and become environmentally friendly.

These three research topics are selected for this thesis to carry out the research work. The choice of these topics follows the rationale that these challenges will become more important to deliver the underlying requirements and emerging demands with the development of vehicular networks. In addition, the 3 themes have strong interconnections by the overarching goal of network optimisation, while focusing on distinct aspects of network performance and sustainable networks.

1.2.2 Motivation of Machine Learning for Vehicular Networks

A defining property of the challenges of vehicular networks is the complex and dynamic network environment, making traditional solutions with a rigorously parameterised general-purpose design highly complicated and underperforming in vehicular networks [18], motivating the investigation of contemporary solutions using Machine Learning (ML). ML is the study of methodologies that automatically improve computer algorithms through experience [46] and is considered as part of Artificial Intelligence (AI); a broad research field that focuses on the synthesis and analysis of intelligent computational agents [47].

ML methods exploit values in data to improve performance on a defined set of tasks. These methods identify / extract patterns and learning from data samples to make predictions or decisions without being explicitly programmed [37] and have been successfully applied to other research fields such as computer vision and automatic

control in the last thirty years [48]. The major advantages of ML over traditionally implemented analytical methods for vehicular networks are:

- ML is capable of efficiently analysing and processing a large amount of data and learning from it, making it suitable to the everchanging vehicular networks' environmental uncertainties, exploiting values from the datasets generated in vehicular networks and enabling adaptive network optimisation [49, 50].

- ML algorithms can be utilised when *a priori* knowledge of the network and parameters is unavailable [51], which is true for most cases in vehicular networks. This makes ML algorithm design more flexible compared to conventional mathematical optimisations built upon fixed parameterised models requiring such knowledge. ML techniques initialise generalised models whose parameters can be updated corresponding to the specific data collected in the environment, making ML able to generalise to vehicular networks' dynamic environment and ideal to deal with complex tasks [52].

- With sufficient data collected locally, it is possible to train localised ML algorithms, making ML more suitable to develop decentralised solutions compared to conventional methods [52]. This is a key feature to enable vehicular networks since sending too much data to a central controller for processing and decision-making is more time-consuming and costly in bandwidth for a centralised control scheme in some use cases.

- With the development of techniques, ML and specifically Deep Learning (DL) techniques (a subset of ML techniques utilising artificial neural networks), these can reach near-optimal performance in many use cases while also being more computationally efficient than conventional algorithms which are usually computationally expensive to get optimal results [37].

As a result, ML applications to wireless and vehicular networking have grown as a research topic with increased research attention at the start of this project, with various learning-based algorithms proposed to address complicated networking problems [49]. However, despite the significant progress in algorithmic design for performance optimisation, the application of ML to vehicular networking is still in its infancy with many research gaps; the real-world deployment aspects such as scalability and impact on implementation cost are largely untouched. This thesis tries to address this shortcoming in the case of V2X networks.

1.2.3 Objectives

This thesis will provide adaptive, intelligent and scalable solutions to vehicular networking, with ML being identified as one of the key methodologies and toolsets to tackle the network optimisation challenges. The main focus of this research's ML application is to improve vehicular networking performance which can be expanded in a hierarchical manner. Firstly, the new research has focused on the novel algorithmic design in vehicular networking utilising the latest ML techniques and architectures to provide high-performing solutions to the identified research topics. Secondly, by considering the deployment aspects such as input parameters, scalability, communication costs, etc. to increase the practicality of the proposed algorithms for potential real-world deployment and reduce the implementation impact on the vehicular network systems. In summary, the research work objective of this thesis is structured as follows:

1. Explore the realm of cellular vehicular networks to identify key challenges for future vehicular network development and provide a literature review on the state-of-the-art ML applications to vehicular networking, centred around the identified research topics.

2. Develop a ML-based HO decision-making solution for V2N in the current cellular architecture to avoid unnecessary HOs and select the best performing target BS.

3. Develop a decentralised resource allocation solution to improve spectrum efficiency and interference management for V2V and V2N and investigate the effect of the enabling decentralised ML architecture on deployment costs.

4. Utilise the latest ML technique to adaptively switch BSs into sleep mode to improve energy efficiency without compromising on QoS requirements or network throughput that can be applied to both V2N and general cellular networks.

5. Provide discussions on the future trends on each identified research topic and ML applications to vehicular networks in general.

Since vehicular networks include a broad domain of research, the first objective of this project is to explore this area of research and identify key research topics of challenge, and thus provide a literature review of ML algorithms applied in this field to identify the current state of the art. This is followed by the objectives of each identified research topic. In addition, it was recognised that existing research predominately utilised custom-built simulations or datasets that were not publicly available for numerical evaluations of proposed solutions. While it is essential for ML methods to be evaluated using standardised datasets and simulation environments to help researchers focus on learning algorithm design and to simplify performance comparison, an important auxiliary

objective to this work on performance evaluation is the utilisation of standardised real-world datasets or realistic simulation environments.

1.3 Contributions

From the literature review undertaken, three specific research questions are developed for objectives 2 to 4, inspired by and extending the work of Yajnanarayana *et al.* [53], Liang *et al.* [25], Ozturk *et al.* [54], etc. This research addresses these research questions forming three distinct areas of contribution:

1. If deployed within the current cellular architecture, does an ML-based HO decision-making scheme improve HO performances for seamless V2N communications?

2. Can the latest decentralised learning architecture be utilised to improve the decentralised radio resource allocation and reduce communication overheads for V2N and general cellular communication networks?

3. Can the latest ML techniques be leveraged to improve cell switching decision-making performance for energy efficiency optimisation for V2N and general cellular communication networks?

An overview of the contributions produced by this project is presented in Figure 1.3. Three main contributions have been made to the selected research topics in parallel with the innovation of utilising the latest ML techniques for adaptive network optimisation schemes focusing on the deployment perspectives. Specifically, the contributions include DL algorithms for current network architectures utilising established techniques for performance improvement, the exploration of distributed learning architecture and new ML techniques for future vehicular network development, and the consideration of deployment aspects assessed using realistic simulations and a real-world generated dataset.

The research work follows the evolving research interest of ML application to vehicular networking and focuses on three specifically identified key use cases. Detailed information regarding the scope is presented in Chapter 2 and 3 for the background and literature review, which correspond to the publication of [36, 55, 56]. Based on the literature, this research answers the three research questions in this rapidly developing field of research as introduced above. In each case, the contributions made encompass mainly software development including algorithmic design and simulation-based experiments for performance evaluation. Specifically, the undertaken research work for each research question is summarised as follows:

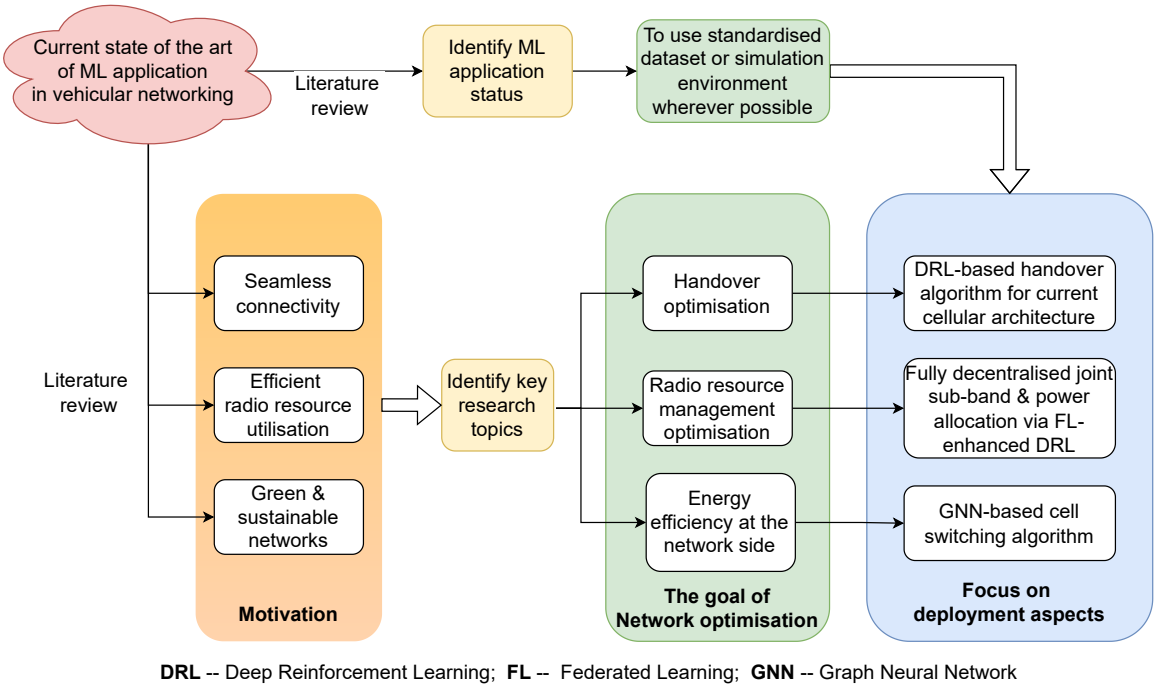


Figure 1.3: Overview of research contributions provided in this thesis.

1. **If deployed within the current cellular architecture, does an ML-based HO decision-making scheme improve HO performances for seamless V2N communications?**

   A HO algorithm utilising Deep Reinforcement Learning (DRL) has been proposed for V2N communications to show the effectiveness of ML. This algorithm produces optimal HO decisions using the same input parameters as the conventional solutions currently deployed in the cellular system, and can be potentially deployed with only a software patch. A realistic bespoke simulation has also been implemented using an established full-stack simulator that provides a standardised evaluation platform. This corresponds to the publication of [57].

2. **Can the latest decentralised learning architecture be utilised to improve the decentralised radio resource allocation and reduce communication overheads for V2V and V2N communications?**

   A fully decentralised DRL algorithm enhanced by the Federated Learning (FL) architecture has been designed for joint spectrum sub-band allocation power control for a spectrum sharing scenario of V2V and V2N communications and is validated for the optimality of allocation policies. The proposed solution had close performance compared to the best-performing state-of-the-art benchmark, while the design of the proposed solution resulted in much lower communication overheads and enabled much simpler online algorithm updates and redeployment. A full-length journal article on this topic has been submitted by the submission

of this thesis.

3. **Can the latest ML techniques be leveraged to improve cell switching decision-making performance for energy efficiency optimisation for V2N and general cellular communication networks**?

   A Graph Neural Network (GNN) based cell switching algorithm has been developed for BSs' energy efficiency optimisation in V2N and generic cellular networks. A well-established telecommunication dataset has been utilised to train and evaluate the algorithm. The proposed GNN solution outperformed the state-of-the-art benchmark with respect to energy efficiency and has a much stronger generalisation capability to different scenarios. This corresponds to the publication of [58].

1.3.1 Publications

Articles:

[36] K. Tan *et al.*, "Machine learning in vehicular networking: An overview," *Digital Communications and Networks*, vol. 8, no. 1, pp. 18–24, 2022, ISSN: 2352-8648. DOI: https://doi.org/10.1016/j.dcan.2021.10.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864821000870

[57] K. Tan *et al.*, "Intelligent handover algorithm for vehicle-to-network communications with double-deep q-learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7848–7862, 2022. DOI: 10.1109/TVT.2022.3169804

[58] K. Tan *et al.*, "Graph neural network-based cell switching for energy optimization in ultra-dense heterogeneous networks," *Scientific Reports*, vol. 12, no. 1, p. 21 581, Dec. 2022, ISSN: 2045-2322. DOI: 10.1038/s41598-022-25800-3. [Online]. Available: https://doi.org/10.1038/s41598-022-25800-3

Conference proceedings:

[55] K. Tan *et al.*, "Clustering algorithm in vehicular ad-hoc networks: A brief summary," in *2019 UK/ China Emerging Technologies (UCET)*, 2019, pp. 1–5. DOI: 10.1109/UCET.2019.8881833

[56] K. Tan *et al.*, "Federated machine learning in vehicular networks: A summary of recent applications," in *2020 International Conference on UK-China Emerging Technologies (UCET)*, 2020, pp. 1–4. DOI: 10.1109/UCET51115.2020.9205482

1.4 Thesis Outline

The remainder of this thesis is organised as follows:

Chapter 2 introduces the essential background of current vehicular networks and the

fundamental ML methodologies and architectures used in this field of research.

Chapter 3 presents an overview of the state-of-the-art ML applications in vehicular networking, consisting of the general introduction and a literature review of the state-of-the-art research utilising ML for each research question.

Chapter 4 focuses on handover optimisation for V2N communications. After the introduction of handover optimisation research and problem formulation, this chapter covers the proposed DRL-based handover algorithm utilising the same input parameters available in the current cellular system, with performance evaluated against the operational handover solution based on measurement event A3. Furthermore, this chapter also provides a qualitative comparison with other state-of-the-art handover solutions with respect to deployment aspects before concluding with a brief summary.

Chapter 5 elaborates the radio resource management for spectrum sharing V2V and V2N communications. It presents the system model after a brief background introduction and a literature review. After that, a fully decentralised DRL algorithm accomplished by the FL architecture is proposed. The performance of the proposed solution is thoroughly evaluated against other state-of-the-art algorithms with an analysis of communication overheads.

Chapter 6 presents the energy consumption optimisation framework for network operators, where a GNN-based cell switching solution is proposed after a succinct background introduction and literature review. With an introduction of the system model, the performance of the proposed solution is evaluated against the theoretical performance upper bound and a state-of-the-art Reinforcement Learning (RL) solution, and the Chapter concludes with a brief summary after a result analysis.

Chapter 7 summarises the conclusions from each of the previous chapters and places these in context with existing challenges, open issues, and future trends for learning-based adaptive vehicular networking.

Chapter 2

Vehicular Networks and Machine Learning Background

Vehicular networks are specific wireless mobile networks with distinctive features in addition to the complex radio environment, while machine learning is a wider set of methodologies that consist of various algorithmic techniques and architectural designs for implementation. Therefore, in order to apply ML to the challenges in vehicular networks, it is crucial to gain fundamental knowledge of vehicular networks and ML before moving on to the detailed design and implementation of the proposed solutions to each of the identified research directions, i.e., Handover optimisation, resource allocation, and energy efficiency. An overview of vehicular networks' background is presented, followed by an introduction to the fundamentals of machine learning relevant to vehicular and general mobile network research.

2.1 Vehicular Network Backgrounds

Vehicular communication networks have attracted interest from different sectors, including academia, industry, and governments in the last two decades [59]. The main goal is to enable information exchange among vehicles (V2V), between vehicles and road infrastructure (V2I), remote networks (V2N), and other road traffic users such as pedestrians (V2P) to support a variety of applications. These applications aim to improve road safety and traffic efficiency and to deliver advanced utility services [36]. With the development of vehicular networks, new technologies have emerged to support new applications and fulfil the futuristic V2X service requirements. As already discussed in Chapter 1, this thesis will focus on vehicular networks with C-V2X technology.

2.1.1 Characteristics and Applications of Vehicular Networks

2.1.1.1 Characteristics of Vehicular Networks

Vehicular networks have unique characteristics because of the Vehicular User Equipments (VUEs) in the network and various V2X applications. These characteristics are summarised as follows and can either be beneficial or challenging to the network designs:

- **Distinctive mobility features**:
  - **High mobility**: Road vehicles move at much higher speeds compared with cyclists and pedestrians. In urban areas, a typical vehicle velocity of 60 km/h means a vehicle can pass a 100 m road segment covered by a static Roadside Unit (RSU) or a small-cell BS in 6 seconds. In a highway scenario, two vehicles driving in opposite directions on the same highway segment can easily have a relative speed of over 300 km/h, covering a 100 m distance in only 1.2 seconds. As a result, vehicular networks have short connection times for all V2X communication types, causing frequent switching and reconnections among V2X terminals (mobile phones, connected traffic lights, base stations, etc.).
  - **More predictive driving routes**: Vehicles are constrained to move on roads and motorways in normal circumstances for urban and highway scenarios. This feature makes it simpler to predict a vehicle's driving route compared with that of a pedestrian, especially if information, including the local map and/or the velocity of the vehicle is available.
- **Complex communication environment** [18]: The communication environment of vehicular networks contains many scenarios such as motorways, rural, and urban street blocks. The motorway scenario is usually considered a one-dimensional environment as vehicles on motorways only travel in two opposing directions with a classic restriction that vehicles' velocities are kept within a predefined range. In comparison, streets in most urban areas are often divided into many segments because of intersections with vehicles moving in different road segments, forming a two-dimensional scenario. Sometimes a direct connection in urban scenarios does not exist due to obstacles around or on the streets such as buildings, trees, or a large metal truck between two cars. Furthermore, urban scenarios have higher vehicle density than motorways in a city centre or congested area, leading to more communication links within the area hence a more significant spectrum resource occupation. The radio environment in urban areas (more specifically city centres with dense building blocks) is more complex as building

reflection and absorption of radio signals become more significant. Additionally, it is also possible for vehicular networks to have three-dimensional scenarios such as viaducts where communication links can be on different layers and even cross layers, further contributing to the complexity of vehicular networks' communication environments.

- **Stringent application requirements**: Some vehicular network applications have very strict requirements in, e.g. maximum end-to-end delay, reliability, and/or data rates. This means that data transmissions must be completed within a certain time and/or achieve minimal required throughput to avoid traffic accidents or to ensure service quality. Safety applications are an obvious example that requires millisecond-level delay tolerance and ultra-reliability. Another example is the remote driving application specified in 3rd Generation Partnership Project (3GPP)'s enhanced V2X scenarios [33], which has a maximum end-to-end delay tolerance of 5 ms and an uplink data rate requirement of 25 Mb/s.

- **VUEs' sufficient energy for communication**: Unlike energy-constrained cellular User Equipments (UEs) such as mobile phones, road vehicles have large batteries and sufficient energy to supply their communication components and thus VUEs have a weak energy constraint.

2.1.1.2 Safety Applications of Vehicular Network

The basic applications of vehicular networks are safety applications aiming to reduce traffic accidents to save lives and property. A straightforward safety application example is event-driven safety warnings which will occur when a vehicle senses or becomes involved in a dangerous situation [18]. The event-driven safety warning includes collision avoidance messaging among vehicles via V2V, hazard warning via V2I links (e.g., a car crash in the coming road section) and early alerts from vehicles to pedestrians via V2P links (e.g., when a pedestrian crossing an intersection without noticing an approaching vehicle) [15, 32]. Another type of safety application is for driving assistance in scenarios such as lane changing and proactive collision avoidance via periodic sharing (e.g. every 20 ms) of position, velocity, direction, and related vehicle-status data among vehicles [18, 60]. This type of information is also essential to more advanced applications such as automated driving. Safety applications are time critical and rely on highly reliable V2X communications, and failures of safety applications can result in serious consequences.

2.1.1.3 Non-safety Applications of Vehicular Network

In addition to safety applications, value-added services for drivers and passengers can be enabled via information sharing among vehicles and between vehicles and other

communication entities, forming the umbrella term of non-safety applications. Based on their target use cases, non-safety applications can be classified into the following application types:

- **Smart traffic management**: The goal of these applications is to improve road traffic efficiency and save time for commuters by reducing traffic congestion and improving traffic flow. An example of this type of application is adaptive traffic light scheduling, which could dynamically adjust the traffic signal timing at intersections according to real-time road traffic conditions [61]. Other examples include pre-trip route planning and live maps based on real-time traffic information gathering via information shared in vehicular networks [15, 62].

- **Infotainment services**: These applications are mainly focusing on providing traveller entertainment and boost onboard comfort. For instance, accessing the Internet in moving vehicles from traditional web browsing to high-resolution video streaming [62]. Infotainment services also provide utility information to vehicles, such as the locations of the nearby petrol stations, parking lots, restaurants, or other places of interest upon request by drivers or passengers [15, 18]. With the advancement of driving automation level towards self-driving cars, there will be greater demands for this kind of services as human drivers getting relieved from the driving duty and become passengers.

- **Advanced driving**: This is an umbrella type summarising the information provided in 3GPP's Release 15 specifications [33] for advanced scenarios. The applications for advanced driving include:

  - **Extended sensors**: Meaning that vehicles can share their sensor data with other vehicular network entities such as another vehicle or a RSU for environmental awareness enhancement, which is a cornerstone for other advanced driving applications.

  - **Automated driving**: As its name suggests, this type of application aims to provide a vehicle with partial-automated driving ability to reduce a human driver's driving workload or fully self-driving capability that does not require a human driver.

  - **Remote driving**: This allows a distant human driver or a cloud server to control a vehicle remotely. Two examples of use cases of this application are public transportation (when passengers cannot drive the vehicle themselves) or driving a vehicle located in hazardous environments such as explosive gas or dust environments [62].

– **Vehicle Platooning**: It enables vehicles to dynamically form a group to travel together, with group management such as adding or removing a platoon member. It essentially means to drive a group of vehicles together by having a leading vehicle that periodically sends commands to member vehicles. This design allows many vehicles to operate (e.g., accelerate or brake) at the same time, thus shortening or removing the required reaction distance and improving the traffic flow.

With both safety and non-safety applications/use cases introduced, Table 2.1 summarises the performance requirement of related KPIs for the above enhanced V2X scenarios, according to the corresponding 3GPP's specifications.

Table 2.1: Performance requirements of a non-exhaustive list of enhanced V2X scenarios [33]

Scenario	**Data rate**[1]	**End-to-end latency**	**Reliability**	**Communication range**
Platooning, V2V (cooperative driving)	65 Mb/s	<10-25 ms	>90-99.99%	>80-150 m
Platooning, V2I (information sharing)	50 Mb/s	<20 ms	-	>180-350 m
Automated driving, V2V (information sharing)	53 Mb/s	<100 ms	-	>700 m
Automated driving, V2I (information sharing)	50 Mb/s	<100 ms	-	>360-700 m
Emergency trajectory alignment, V2V	30 Mb/s	<3 ms	-	-
Intersection safety information, V2I	UL: 25 Mb/s DL: 50 Mb/s	-	-	-
Video sharing, V2N	UL: 10 Mb/s	-	-	-
Video sharing, V2V	10-700 Mb/s	<10-50ms	>90-99.999%	100-400m
Remote driving, V2N (information sharing)	UL: 25 Mb/s DL: 1 Mb/s	<5 ms	>99.999%	-
Extended sensor (information sharing)	10-1000 Mb/s	<3-100ms	>95%-99.999%	50-1000m

[1] UL: Uplink; DL: Downlink.

2.1.2 5G and Beyond for C-V2X

C-V2X has seen enhancement with the development of 5G standardisation. New enabling technology has been defined in the 5G building block to better serve the emerging sophisticated applications and use cases that have more stringent requirements [36]. As technologies advance, legacy network designs are increasingly challenged to meet the new service requirements, leading to a paradigm shift for future vehicular networks. This includes the inclusion of new enabling technologies, the shift from relatively sparse to ultra-dense deployment, the inclusion of higher frequency bands, and the shift from reactive networking solutions to proactive ones.

Cellular V2X applications specify strict requirements in low latency enabling Mobile

Edge Computing (MEC) to become a fundamental building block for "5G and beyond" vehicular networks [21]. MEC is an enabling technology that brings core network services closer to users by moving the related computation tasks from traditional cloud servers towards the network edge, i.e., BSs and VUEs [63]. By increasing the localised computation capability, this reduces the need of a remote server and hence reduce the overall round-trip communication delay. On the other hand, this leads to additional challenges as computing resources, including computing power and memory, must be properly managed at the network edge.

Another important enabling technology for cellular networks and C-V2X is the concept of Network Function Virtualisation (NFV), which decouples network functions from specific hardware components. This is achieved by performing these functions on cloud servers then sending the results back to the edge hardware [50]. By decoupling functions from specific hardware, more adaptive solutions can be developed that are less dependent on specialist hardware. This also enables centralised solutions or solutions demanding computational power and data storage to be implemented more flexibly.

In addition to the enabling technologies, there is also a paradigm shift in the network design to satisfy future network requirements. Network densification is a critical component of cellular networks and V2N communications to cope with the anticipated increase in traffic and capacity. However, network densification leads to a more complex radio environment in the cellular network and will increase the burden on operators due to the increased network complexity. Nevertheless, more data can be generated by configuring densely deployed BSs and future network solutions that can process and exploit value in such data can be highly advantageous. The utilisation of higher frequency spectrum bands is another approach to boost throughput and network capacity for cellular systems such as the inclusion of the Millimetre Wave (mmWave) band from 30 GHz to 300 GHz [21, 64]. This allows network operators to move away from the already saturated microwave bands while also being able to allocate larger bandwidth for applications demanding high network throughput. However, mmWave also brings additional challenges to network solution due to its physical characteristics, specifically high propagation path loss and low signal penetration (e.g. into buildings) [64], which reduces coverage and increases its sensitivity to signal blockage [65].

The current networking solutions in the cellular system are reactive [36, 50], which means that the system produces a response after receiving and processing specific sensory inputs (similar to the event-driven warning of vehicular networks' safety applications). Such approaches are deemed mostly suboptimal in networking research since they may cause extensive waiting periods and thus a reduced QoS. Consequently, it is expected that future vehicular networks will employ proactive solutions that operate in a more adaptive and informed manner for optimised network performance.

2.2 Machine Learning: The Fundamentals

Traditional network optimisations employ analytical heuristics or mathematical modelling that have high computational complexity, lack generalisation capability, and need re-execution upon network condition changes. This motivates the exploration of applying ML techniques to vehicular networks.

ML is a subset of AI methodologies that automatically improve computer algorithms through experiences [46]. The key properties of ML methodologies are that ML programs utilise collected data instead of explicit human instructions and can adapt to new data automatically. Therefore, instead of constructing a complete and complex analytic model of a system by a human programmer as in conventional algorithmic development, ML aims to find intrinsic patterns and produce a model that relates the input to the output of the system through analysing collected datasets [50]. As a result, ML is applied to scenarios that are too complex for analytic algorithms, with specific tasks such as identifying handwritten characters (e.g., Chinese characters), predicting stock prices, and playing video games [66].

ML has also been identified as a promising technique for optimisation problems [67] which is ideally suited to vehicular networks and network optimisation. Note that vehicular networks are well suited to the ML approaches as the network is naturally rich in data collected by VUEs, BSs, and other communication entities. Additionally, the advancement in computing equipment facilitates ML solutions for network analysis and management to complete with high accuracy and in a timely manner, overcoming the run-time limitations of traditional mathematical techniques which usually faced high computational complexity due to the complicated optimisation formulation [37].

Traditionally, ML approaches are classified into three broad categories or learning paradigms, i.e. supervised, unsupervised, and reinforcement learning, based on how the available information is utilised by the learning system. It is important to appreciate the differences between these three ML subclasses so the approach most suited to network optimisation can be chosen.

2.2.1 Supervised Learning

Supervised learning algorithms is a subclass ML algorithms learning on a set of data samples termed “labelled data” that consist of example inputs and their desired outputs [68]. The dataset is known as a training dataset and is provided externally to the learning system through human intelligence and labour. Therefore, supervised learning algorithms can be regarded as learning from a supervisor or a teacher as indicated by the name.

The goal of supervised learning algorithms is to develop a mathematical model representing the relationship between the inputs and outputs in the training dataset, which the algorithms can utilise to produce correct outputs for additional inputs that were not part of the training data. An objective function is used as the optimisation goal for the labelled data when training the algorithm to produce a correct model [69]. Typical supervised learning methods include regression models such as Linear Regression [70], Decision Tree learning [71], Support-vector Machine (SVM) [72], and K-nearest Neighbours (KNN) [73]. The quality of labelled data can influence the learning outcome of supervised learning algorithms significantly due to this learning process and hence it is vitally important that the training data set is accurately pre-processed.

Supervised learning tasks can be split into two main categories: regression and classification tasks. Regression algorithms are used when the outputs are expected to have any numerical value within a range, while classification algorithms should be chosen when the outputs are restricted to a limited set of discrete values. An example regression task is stock price prediction, and a typical classification example is handwritten character identification (e.g. handwritten alphabet recognition) as mentioned above. When applied to vehicular networks, supervised learning is often utilised for prediction such as mobility and traffic load (or congestion) prediction to assist proactive networking decision-making [50].

2.2.2 Unsupervised Learning

Unsupervised Learning is the second category of ML. In contrast to supervised learning, unsupervised learning algorithms do not have a supervisor providing labelled data and must learn directly from datasets containing only inputs. An example of such datasets may be a photo collection of pets without specific tagging on what kinds of animals individual pets are (cats, dogs, etc.).

The aim of unsupervised learning algorithms is to find underlying patterns in the input data [70]. As the production of fully labelled datasets often demand significant effort and thus is very expensive, unsupervised learning algorithms become more useful for unlabelled datasets to extract such patterns. Clustering analysis is a typical unsupervised learning method where input data samples are categorised into different groups according to their key features, and these groups are distinguished with certain boundaries. Typical unsupervised learning algorithms include K-Means clustering [74], hierarchical clustering [75], and self organising feature maps [76], etc.

An example application of unsupervised learning is customer segmentation for business analysis [77]. When applied to wireless networking applications including vehicular networks and wireless sensor networks, unsupervised learning is often utilised for signal overhead reduction [78], network traffic classification [79], and abnormality detection

[79, 80].

2.2.3 Semi-supervised Learning

Supervised and unsupervised learning are classified based on their supervision level, i.e., whether have access to fully labelled data or not. Supervised learning requires a full set of labelled data for training while unsupervised learning works with completely unlabelled data. Semi-supervised learning sits between these two extremes, working with training datasets that contain both labelled and unlabelled data samples. Note that semi-supervised learning is an extension to the above two main ML categories, and thus not treated as a major ML type.

Using structural assumptions to automatically exploits values in unlabelled data, semi-supervised learning is useful to improve supervised learning tasks when the labelled data are scarce or expensive. Additionally, semi-supervised learning also has the potential to be used as a quantitative tool for reasoning where most of the input is self-evidently unlabelled for human beings [81]. Although relatively little attention has been paid to semi-supervised learning in vehicular network research, it is a promising tool to exploit vehicular networks' rich unlabelled datasets while also leveraging the existing domain expertise.

2.2.4 Reinforcement Learning

RL is the third and a distinctive subclass of ML with the inherent property of *learning by interaction*. This idea may be the most natural way to think about "learning" with a lot of examples such as learning to ride a bicycle or to drive a car, and RL is the computation approach to learning from interaction [82] and focuses on sequential decision-making problems.

This property distinguishes RL from supervised and unsupervised learning techniques learning on labelled/unlabelled data, and the learning process in RL is for a goal-directed software agent to interact with an uncertain environment and achieve its goal [82].

RL problems are often formulated as a Markov Decision Process (MDP). As shown in Figure 2.1, an agent takes actions based on its observation of the current environmental state, receives the observation of the following state, and receives a numeric reward generated by the environment. After many rounds of training through trial and error, the agent develops an optimal policy it adopts in taking action within the given environment that maximises the long-term accumulated reward. According to [82], the key elements of a MDP are defined as follows:

- **Agent**: The learner and decision maker, corresponding to the term "controller"

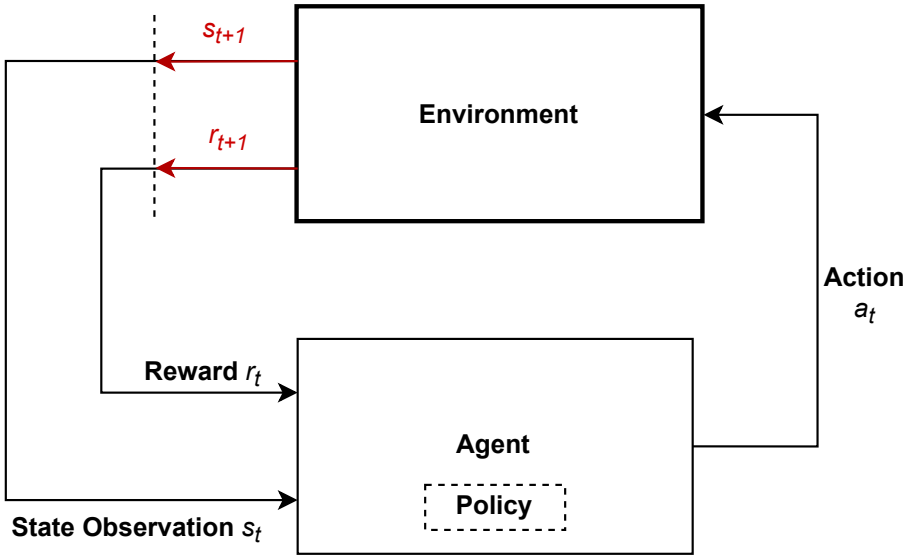


Figure 2.1: The Markov decision process. Figure 3.1, [82].

in the engineering realm.

- **Environment**: covers everything outside the agent that responds to the agent's actions, corresponding to the term "controlled system".

- **Action** a: A possible output selected by the agent to interact with the environment, which can be regarded as a "control signal".

- **State and observation** s: A state contains information that describes the condition of the environment. The agent takes actions based on the observation of a state that can be the complete state information or a subset of it for a partially observed MDP.

- **Reward** r: A special numerical value the environment sends to the agent after each interaction. It shows how good an action is in an immediate sense, i.e., r_t for a_t in s_t at time step t.

- **Policy** $\pi(a|s)$: A mapping from the agent's state observations to its actions, which defines the agent's overall behaviour in the environment. The goal of RL is for the agent to develop an optimal policy acting in the environment, e.g., learning to play chess at a competitive level from scratch.

The agent aims to maximise its expected accumulation of reward values over time by adjusting its actions. This reward accumulation is termed return G_t, whose definition is:

$$G_t \equiv r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... = \sum_{\tau=0}^{\infty} \gamma^k r_{t+\tau+1} \tag{2.1}$$

where $\gamma \in [0, 1]$ is a parameter termed the discount factor, and $\gamma^{t+\tau-1}$ specifies a future reward received τ steps ahead of the current time step t is worth only $\gamma^{\tau-1}$ times what it would be if it were received immediately at t. The agent focuses on immediate reward only when $\gamma = 0$, while the future rewards are taken into consideration more strongly by the agent as γ approaches 1. When related to playing chess, an immediate move to take a pawn results in a positive reward but it may cause the player to lose a higher-valued piece, e.g., a knight, in the next turn, which has a negative reward. In comparison, losing a bishop this turn results in a negative reward, but it may help the player win the next few turns, which leads to a very high positive reward. γ in training an RL chess player hence decides how strong the future situations should be taken into consideration when making an immediate move, i.e., how the agent treats an immediate move in the long term.

To maximise the accumulated reward, RL algorithms evaluate the value functions of a state (or a state-action pair) to estimate the expected return when following a policy. There are two types of value functions: State-value and Action-value (Q-value) functions. The state-value function $V_\pi(s)$ is the expected return when starting in s and following policy π, while the action-value function $Q_\pi(s, a)$ is defined as the expected return starting from s, taking the action a, and thereafter following policy π [82]. The mathematical definition of these two types of value functions, assuming the state and action, at time step t, is $s_t = \mathtt{s}$ and $a_t = \mathtt{a}$, defined in (2.2) and (2.3), respectively.

$$V_\pi(\mathtt{s}) \equiv \mathbb{E}_\pi \left[G_t | s_t = \mathtt{s}\right] = \mathbb{E}_\pi \left[\sum_{\tau=0}^{\infty} \gamma^k r_{t+\tau+1} | s_t = \mathtt{s}\right] \tag{2.2}$$

$$Q_\pi(\mathtt{s}, \mathtt{a}) \equiv \mathbb{E}_\pi \left[G_t | s_t = \mathtt{s}, s_t = \mathtt{a}\right] = \mathbb{E}_\pi \left[\sum_{\tau=0}^{\infty} \gamma^k r_{t+\tau+1} | s_t = \mathtt{s}, a_t = \mathtt{a}\right] \tag{2.3}$$

where $\mathbb{E}_\pi$ denotes the expected value of a random variable when following π, $r_{t+\tau+1}$ stands for the reward at time step $t + \tau + 1$, and S denotes the state space for all possible states. The optimal policy is achieved when the value functions of all states (or state-action pairs) following it reach their upper bound, and thus the optimal value functions are defined as $V_*(s) \equiv \max_\pi V_\pi(s)$, $Q_*(s, a) \equiv \max \pi Q_\pi(s, a)$, for all $s \in S$, $a \in A$.

RL techniques can be broadly classified as tabular solutions and approximation-based solutions [82]. Tabular solutions focus on environments where the state spaces can essentially be represented by tables, while approximation-based solutions extend the tabular ones for applications to arbitrarily large state spaces that would otherwise be too large to be described by tables. Modern RL algorithms are difficult to accurately classify in an all-encompassing manner [83]. However, two important branching criteria are "whether the agent has access to the model of the environment" and "how the learning is performed", which are also relevant when applying RL to vehicular networking research [37, 82, 83].

- **Model-based** or **Model-free**: An RL algorithm is model-based if it is developed with access to the environment model, i.e. a mathematical function that represents state transitions and rewards; otherwise, it is model-free.

- **On-policy** or **Off-policy**: An off-policy algorithm indirectly updates the policy being optimised following another behaviour rule. In contrast, on-policy algorithms aim to directly optimise the policy that guides the agent's behaviour. As a result, off-policy algorithms may use data collected at any point during the training regardless of the agent's behaviour when exploring the environment and obtaining experiences, which is more data-efficient; however, this may potentially lead to unstable solutions in some situations [82]. On the other hand, on-policy algorithms directly optimise the target policy and tend to have more consistent and stable training while only data samples generated with the latest version of the policy can be used in training.

One of the most popular tabular RL algorithms is Q-learning [84], a model-free, off-policy RL algorithm. Q-learning updates the state-action value function (Q value) and evaluates the optimal value for action-taking. The definition of Q-learning's value update is expressed as:

$$Q(s_t, a_t) \leftarrow +\alpha[r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{2.4}$$

where $0 < \alpha \leq 1$ is a step-size parameter (or learning rate).

So far, the fundamental components and essential concepts of RL have been introduced. These components (environment, agent, reward etc. in the MDP model) play a vital role in the algorithmic design of Chapters 4 and 5. The concepts such as the value functions can help in understanding the core of learning in RL. Additionally, a taxonomy of RL techniques has been introduced to help the reader gain high-level knowledge of RL algorithms.

2.2.4.1 Multi-agent Reinforcement Learning

Similar to single-agent RL, Multi-agent Reinforcement Learning (MARL) also aims to solve sequential decision-making problems. However, multi-agent RL extends the decision-making scenario to two or more decision-making agents. This is useful for vehicular network applications, especially for decentralised resource allocation scenarios without a BS.

A typical model for MARL is the Markov Game which is a repeated game played by one or more players with state transitions and is a direct generalisation of the MDP model [85]. Figure 2.2 shows a Markov Game with K agents forming a group that acts in the environment. All agents take actions at the same time after receiving their state observations of the environment to form a combined action in the environment and receive their corresponding rewards.

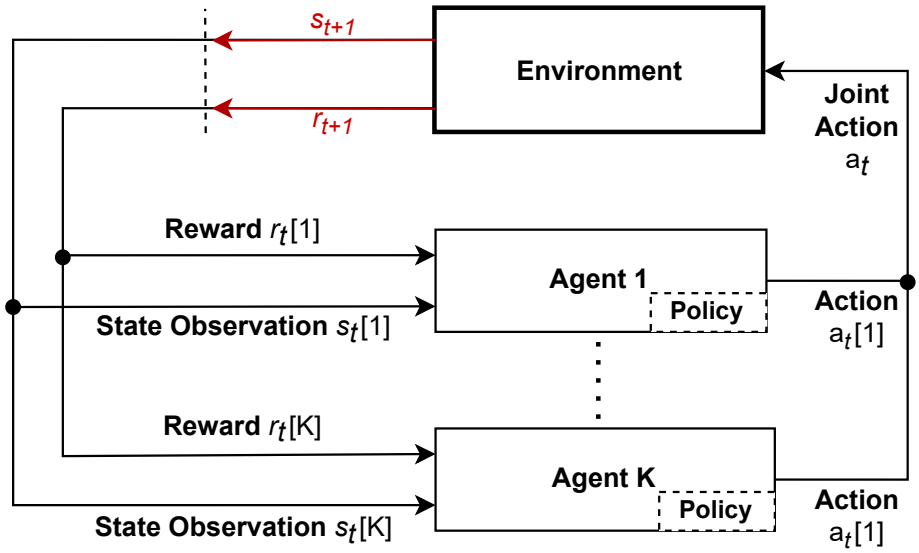


Figure 2.2: The Markov Game model for MARL.

Based on the reward design, MARL can be classified in a three-fold manner summarised below [86]:

- **Fully Cooperative**: All agents share a common reward function in a fully cooperative setting, i.e., a single reward `r` for all K agents such that $r_1 = r_2 = ...r_K =$ `r`. In this case, the Markov Game model can be regarded as a direct multi-agent version of the MDP model. A more generalised variant of this setting is to allow agents to have different reward functions, while the ultimate goal is to maximise the average reward among all agents, which is a fundamental assumption of this setting.

- **Fully Competitive**: Agents in this setting compete with each other to maximise their own rewards, and the improvement in one agent's reward will reduce the reward of another agent. A typical model for this setting is the zero-sum game

[87] in Game Theory, with a real-world example being the game of Go or a classic tic-tac-toe where two players play against each other; if one agent wins then the other agent loses unless the game ends with a draw.

- **Mixed setup**: This is the most generalised setting that does not restrict the optimisation goal and the relationship among agents. All agents are defined to be self-interested, and the reward functions of an agent may conflict or harmonise with other agents' rewards. This is also known as the general-sum game [88] in Game Theory.

MARL can also be categorised based on the information structure that defines the availability of external information (beyond local state observation) to each agent and can thus greatly influence the overall analysis and optimisation design. As shown in Figure 2.3, there are three representative information structures with their features summarised as follows [86]:

- **Centralised**: A centralised controller exists in this setup where agents can communicate and share local information. The controller can aggregate shared information from all agents and thus has the knowledge of such information, including joint actions (actions from all agents), rewards, and joint state observations (aggregated information of the state observation), etc. This is presented in Figure 2.3a.

- **Decentralised with communicating agents**: The centralised controller does not exist for information aggregation in this setup. However, agents can communicate with each other in a communication network such that local information exchange is still possible. Note that an agent may only be able to communicate with its close neighbours due to potential communication channel limitations. This is presented in Figure 2.3b

- **Fully distributed**: In addition to the absence of a centralised controller, information exchange among the agents is also unavailable in this setting. Agents can only act according to their observations although it is still possible for agents to gain some global information through these observations. The extreme case of this scenario is *independent learning* [89] which assumes each agent's observations contain only local information of the local environment state observation, the agent's own actions and received rewards. This is presented in Figure 2.3c and note that the main difference between "fully distributed" and "decentralised with communicating agents" above is whether the agents can share information with each other.

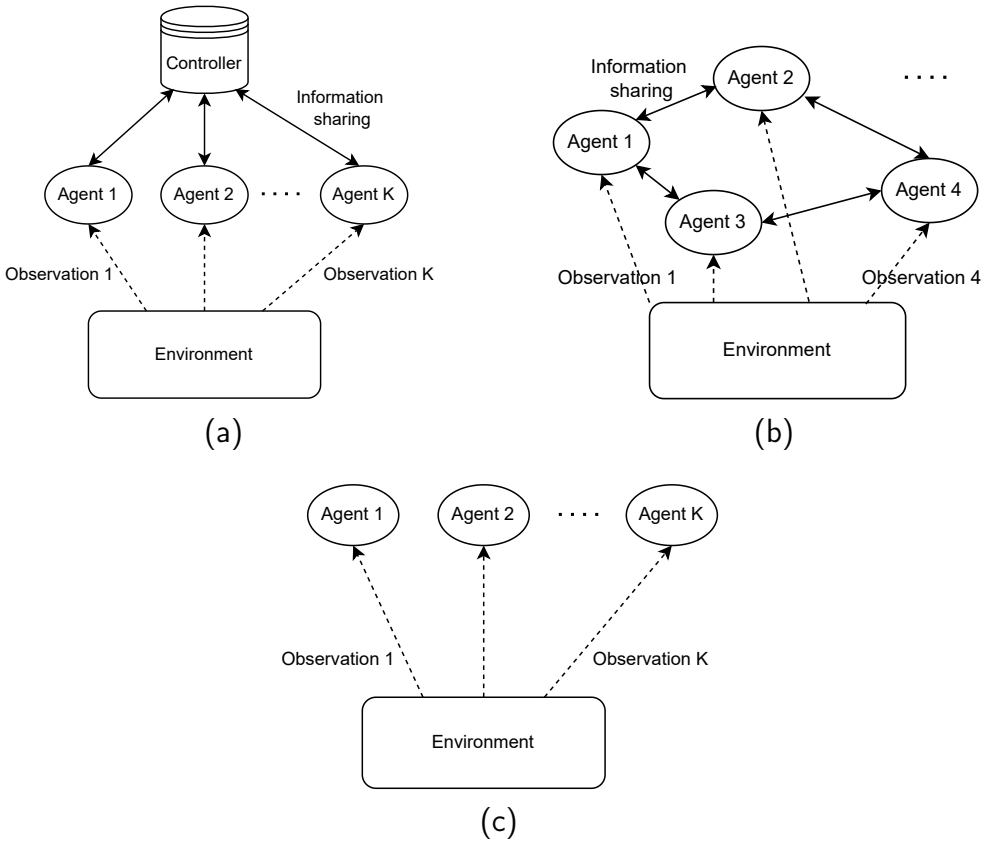


Figure 2.3: Three distinctive MARL information structures. (a) Centralised setting. (b) Decentralised setting with communicating agents. (c) Fully distributed setting.

An example to introduce these representative information structures may be a set of scenarios where a group of robots are exploring the terrain of the planet Mars within an area:

- **Centralised**: There is a transmission tower connecting all robots and collecting information from the robots' sensors and sending it to a remote human operator. The operator can also update each robot's navigation routes through the transmission tower.

- **Decentralised with communicating agents**: The above transmission tower does not exist, but all robots are connected to their geographical neighbours through a wireless ad-hoc network to share their sensory inputs and collectively decide each robot's navigation route and tasks to complete automatically.

- **Fully distributed**: The transmission tower does not exist and the robots are not connected through any communication network. Each robot must decide its following navigation route according to its own sensory inputs only.

2.3 Deep Learning

With the development of ML techniques and the improvement of computing devices, many ML approaches have been developed that do not neatly fit into the above three-fold categorisation. In some cases, an ML system implements more than one type of technique.

DL is such a subset of ML techniques utilising artificial neural networks as the underlying algorithmic learning model for all classes of ML tasks. The strong expressiveness of Artificial Neural Network (ANN)-based DL algorithms have in research directions such as image recognition, natural language processing, and playing complex games such as Go [90–92]. As a result, DL is currently the dominant approach for a lot of ongoing work in the field of ML research and applications [70, 93] and gains much attention from wireless network research [37].

2.3.1 Artificial Neural Network

Artificial Neural Networks (ANNs) are originally inspired by biological neurons, forming the foundation of DL. An ANN is composed of one or more layers of interconnected artificial neurons (processing elements). Each artificial neuron has inputs and produces an output that is either sent to other neurons or used to form the final output [94]. A defining advantage of ANN models is their ease of use and higher accuracy for complex natural systems with large-scale inputs compared to other ML models [95].

2.3.1.1 Feed-forward Neural Networks

The fundamental ANN architecture for DL is the feed-forward ANN [93]. The connections between artificial neurons in a feed-forward ANN are directed from one layer to its next layer and do not form any loops, such that information flows only in one direction. Figure 2.4 demonstrates a typical fully connected, feed-forward ANN consisting of an input layer, a hidden layer, and an output layer. The input layer is the layer that receives external data, while the output layer produces the final result. Between these two layers are the hidden layers that process the intermediate data.

Note that DL algorithms usually utilise ANNs with multiple hidden layers, and the number of hidden layers can vary depending on the scale of the dataset and the complexity of the target problem.

For a linear feed-forward layer l, the relationship between its vector input $\bar{x}$ and output $\bar{y}$ is defined as follows:

$$\bar{y} = \phi(W_l^T \bar{x} + b_l) \tag{2.5}$$

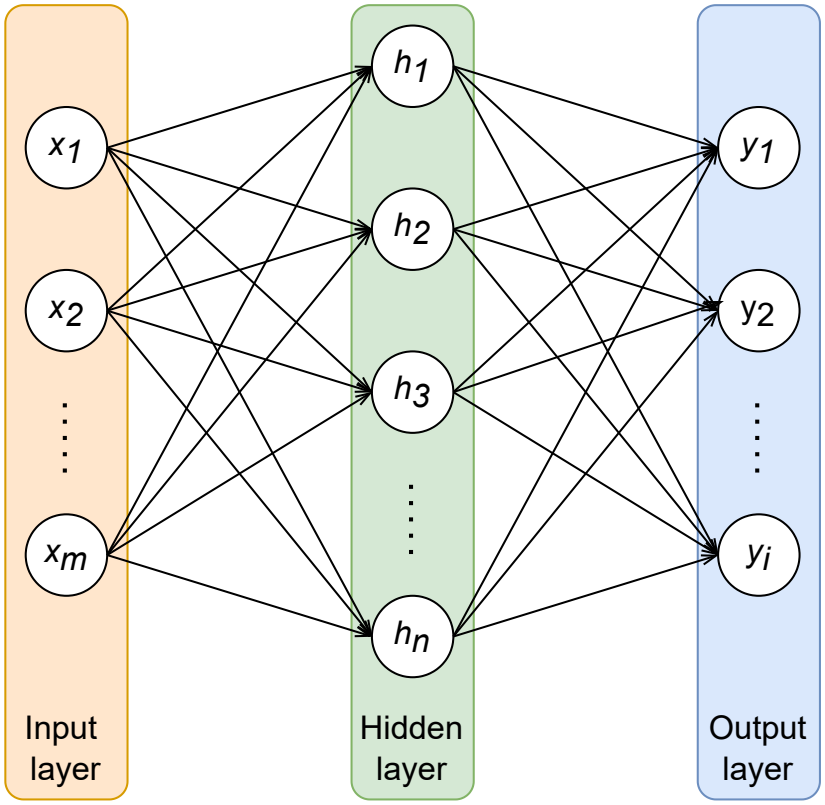


Figure 2.4: A typical fully connected, three-layer feed-forward ANN with one hidden layer. Each circle represents an artificial neuron, and each arrow stands for the connection from the output of a neuron to the input of another neuron.

where $\phi(\cdot)$ is an activation function that provides non-linearity. W_l stands for the weights of all neurons in l, b_l denotes an optional bias of layer l, and T stands for the matrix transpose operation. As a result, (2.5) essentially stands for a two-step operation that contains a linear transformation with a subsequent non-linear transformation.

Other ANN architectures also exist with the development of DL techniques, such as a Recurrent Neural Network (RNN), which allows connections between neurons in the same or previous layers [96], or a Convolutional Neural Network (CNN) that uses a mathematical operation called convolution in place of general matrix multiplication expressed in (2.5) in at least one of their layers [97].

2.3.1.2 Learning and Hyperparameters

The learning process of a DL algorithm is updating and adjusting the underlying ANN's learnable parameters, such as the above W_l and b_l, by adapting to the input data samples to better handle a predefined task, e.g., to gain higher accuracy for a prediction task. The learning process is accomplished by minimising observed errors practically guided by a defined cost function such as the Mean Squared Error (MSE). The learning is completed when the error rate reduction becomes stagnant, even with additional data samples examined. The updates of the ANN's learnable parameters for error reduction are typically achieved by the process of backpropagation [98]. Backpropagation calculates

the gradient of the cost function with respect to a given state of ANN parameters. The calculated gradient is further utilised by an optimisation algorithm (or an optimiser) such as stochastic gradient descent [99] to update the ANN's learnable parameters and reduce the error calculated by the cost function. Note that other optimisation methods also exist besides backpropagation-based optimisers [100–102].

In addition to the learnable parameters such as neurons' weights, another category of parameters of DL exists, termed hyperparameters. Typical hyperparameters for DL algorithms include:

- **The number of hidden layers and neurons in every hidden layer**: Also known as the depth and width of an ANN model. Deeper ANN and wider layers stand for more transformation operations with more parameters, e.g., (2.5) and W_l for a linear layer.

- **The activation function**: An activation function can provide the non-linearity to an ANN's feed-forward process as demonstrated in 2.5. It changes how the output of a previous layer is perceived by the following layer.

- **The cost function (loss)**: For a defined task, the loss is used to measure the performance of an ANN model.

- **The optimiser**: An algorithm used to minimise the loss and update the ANN's learnable parameters.

- **The learning rate**: The corrective step size that the optimiser takes to adjust ANN parameters for error reduction.

Hyperparameters are a set of constants that control the behaviour of a DL algorithm. Defined before the learning process, hyperparameters can greatly influence learning outcomes [97]. The research work in this thesis contains different configurations of the above hyperparameter alongside algorithmic designs. Therefore, it is important to understand the concepts of these hyperparameters for DL implementation.

2.3.2 Deep Reinforcement Learning

In Section 2.2.4, traditional RL algorithms were shown to be successful in a tabular environment, but such approaches become inefficient when faced with a highly complex environment with very large state space or high-dimensional state inputs such as an outdoor radio environment. The problem with large state spaces is that it necessitates very large memory to store large tables, while the time and data required to fill them accurately also being a major issue [82].

As a result, approximation-based RL methods were proposed for such scenarios. This type of RL methods aims to replace the exact RL components such as the value functions, the policy, or the model of the environment to their corresponding approximations [103]. Typical approximation-based RL algorithms include linear value function approximation or policy gradient methods, etc. [104]. For instance, linear value function approximation for Q-learning aims to replace the exact Q-table with a linear function approximation $Q(s, a; \bar{W}_q)$ using a vector of linear weights $\bar{W}_q$, such that $Q(s, a) \approx Q(s, a; \bar{W}_q)$. The goal of learning is to minimise the error between the approximated and the true Q values, defined as $\min_{\bar{W}_q}[Q(s, a) - Q(s, a; \bar{W}_q)]^2$.

With the development of DL techniques using ANN, DRL algorithms utilising the strong approximation abilities of an ANN were developed as a promising candidate for approximation-based RL methods. Straightforwardly, DRL algorithms use ANNs to approximate the corresponding RL components as mentioned above and are trained in a DL manner. The representing DRL algorithm now known as Deep Q-network (DQN) was first proposed by Mnih *et al.* [105] which also substantially launched the field of DRL [83]. For two of the three research questions, DRL algorithms are chosen as the main technique for the design of the proposed solutions. This is because of the following advantages of DRL, inheriting the advantages of general RL:

- RL is highly suitable to model/solve a sequential decision-making optimisation problem as a whole without the need of dividing it into sub-problems to solve analytically.

- RL algorithms learn via interactions between the environment and agent(s) and thus do not rely on collected datasets or knowledge input from an external supervisor.

- The learning process is guided by the reward function which is highly flexible to design and adjust according to different end goals of learning.

- The model-free RL algorithms utilised in this thesis do not require a precise model or *a priori* knowledge of the considered environment, which is highly beneficial for V2X with highly complicated radio environments.

- DRL algorithms utilise ANNs as the underlying approximation model which has very strong expressiveness. This makes DRL suitable for complex problems while resolving the aforementioned limitations of traditional tabular RL solutions.

The remainder of this section introduces the core algorithmic design of two (an off-policy and an on-policy) typical DRL algorithms that are used to develop the proposed

solutions in Chapter 4 and 5.

2.3.2.1 Deep Q-learning

DQN combines a feed-forward ANN and Q-learning [106], a model-free, off-policy RL algorithm, to cope with complex environments where traditional RL approaches encounter difficulty.

Derived from the original Q-learning's temporal difference calculation [106], the update of the Q-valve in DQN is shown in (2.6). In this equation, s and a represent the current state and action taken, respectively, while s' and a' stand for the resultant state and the action taken in that state. $argmax_{a'}Q(s', a'; \theta_q)$ is used to approximate the optimal Q-value $Q^*(s', a')$ where θ_q denotes the ANN approximating the Q table, termed as Q-network. Additionally, r and $\gamma \in [0, 1]$ represent the received reward and the discount factor, respectively.

$$Q(s, a; \theta_q) = r + \gamma Q(s', argmax_{a'}Q(s', a'; \theta_q); \theta_q) \tag{2.6}$$

To update the Q-network θ_q, the DQN algorithm is trained to minimise the loss function $L(\theta)$ as defined by (2.7) for each iteration i [105]:

$$L_i(\theta_{q,i}) = \mathbb{E}_{s,a\ \rho(\cdot)}\left[(y_i - Q(s, a; \theta_{q,i}))^2\right] \tag{2.7}$$

where $y_i = \mathbb{E}_{s',\ \varepsilon}[r + \gamma max_{a'}Q(s', a'; \theta_{q,i-1})|s, a]$ and $\theta_{q,i}$ are the target Q-value and Q-network for iteration i respectively, and $\rho\ (s, a)$ is the behaviour distribution which is a probability distribution over the state s and action a. The gradient of this loss function can be obtained after the differentiation operation [105]:

$$\nabla_{\theta_i}L_i(\theta_i) = \mathbb{E}_{s,a\ \rho(\cdot);s'\ \varepsilon}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{q,i-1}) - Q(s, a; \theta_i)\nabla_{\theta_{q,i}}Q(s, a; \theta_{q,i})\right] \tag{2.8}$$

For DQN, the prediction Q value $Q(s', a')$ and updated Q value $Q(s, a)$ are calculated via the same ANN θ_q. While updating the current $Q(s, a; \theta_{q,i})$ will change the value of future states (as parameters of $\theta_{q,i}$ are updated), it leads to potential instability during training that may result in non-convergence. To overcome this problem, a new DRL algorithm DDQN has been developed [107]. In DDQN, another ANN θ'_q is introduced as the target network for the calculation of the predicted value $Q(s', a')$, alongside the

training network θ_q that calculates the current value $Q(s, a)$. This changes (2.6) to (2.9) and hence the corresponding $Q(s', a'; \theta_{q,i-1})$ to $Q(s', a'; \theta'_{q,i-1})$ in (2.7) and (2.8). Throughout the training process of DDQN, only θ_q will be updated with each training iteration, while θ'_q only synchronises periodically with θ_q by copying all parameters from θ_q to keep the information updated. This design can greatly stabilise the DRL training and improves the chances of convergence.

$$Q(s, a; \theta_q) = r + \gamma Q(s', argmax_{a'} Q(s', a'; \theta_q); \theta'_q) \tag{2.9}$$

During Q-learning (hence DQN and DDQN) training, an experience consisting of the current state, action taken, reward, and the resultant state observations (s, a, r, s') is used once to update the value function parameters and then discarded. This is not only inefficient but may also cause instability since the algorithm only considers the latest experience sample for each parameter update. Consequently, DQN adopts the Experience Replay [108] technique to improve the sample efficiency and reduce the potential instability by re-utilising all experience samples. This is achieved by having a replay buffer $\mathbb{B}$ to store all previous experience samples formulated as the tuple (s, a, r, s') until the maximum capacity of $\mathbb{B}$ is reached, and then the oldest experience will be deleted. To update ANN parameters, experiences in $\mathbb{B}$ are uniformly sampled so that both current and previous experiences are considered during the algorithm's training process. Using experience replay can significantly improve the sample utilisation efficiency and the overall performance of the algorithm [105].

2.3.2.2 Advantage Actor-critic

DQN and DDQN are the representative off-policy DRL algorithms inherited from the original Q-learning. Instead of achieving an optimal policy directly, these algorithms aim to optimise the Q-value such that an underlying optimal policy can be inferred. Although being highly data efficient as previously introduced, these algorithms may have issues in performance stability. The reason is that the optimised target of DQN and DDQN is a self-consistent value function instead of a well-formulated policy [109].

In contrast, on-policy RL algorithms known as policy optimisation learn to directly optimise a policy that guides an agent's behaviours. Being principled with the optimisation target also being the desired entity, policy optimisation algorithms tend to have better stability and consistency compared with Q-learning algorithms [83].

The Advantage Actor-critic (A2C) algorithm is a policy optimisation approach in the DRL algorithm subset known as policy gradient [110]. It encompasses two ANNs: The Actor-network θ_a and the Critic-network θ_c. θ_a represents the policy $\pi(a, s; \theta_a)$ deciding

which action within the action space should be taken, while θ_c works as an estimate of the value function $V(s_t;\theta_c)$ and evaluates how good the action was and how it the adjustment should be. One of the defining properties is its usage of the Advantage function $A_(s,a)$, defined as follows [111]:

$$A(s,a) = Q(s,a) - V(s) \tag{2.10}$$

where $Q(s,a)$ and $V(s)$ are the action-value and state-value functions, respectively, as defined earlier in Section 2.2.4. The advantage function $A(s,a)$ shows the improvement when taking action a in state s compared to the average value of that state given by $V(s)$. An action `a` in state `s` leading to positive A means that taking `a` will result in a higher reward, and the policy should be pushed in favour of it, while the opposite is true for actions that lead to negative A. A2C operates in the forward view and uses the n-step accumulated reward to update both the policy π and the value function V. An update happens after every predefined k action execution or when a terminal state is reached. The advantage function update for the A2C algorithm is defined as follows:

$$A(s,a;\theta_a,\theta_c) = \sum_{i=0}^{k-1} \gamma^i r_{t+i} + \gamma^k V(s_{t+k};\theta_c) - V(s_t;\theta_c) \tag{2.11}$$

where $k \geq 1$ may vary for different states and is upper bounded by the maximum step length defined by the environment, with other notations previously introduced.

The major disadvantage of the above definition of A is that both value functions are required. However, since $Q(s,a)$ can be estimated by $V(s)$ according to the Bellman optimality equation [112] such that $Q(s,a) = r + \gamma V(s')$, Eq. (2.10) can be modified to:

$$A(s,a) = r + \gamma V(s') - V(s) \tag{2.12}$$

where $r + \gamma V(s')$ is the temporal difference error. The actor-network θ_a and critic-network θ_c are updated by minimising their corresponding loss functions via the gradient calculations. The loss for θ_a is the policy gradient [111] combined with the advantage A, while that for θ_c is calculated in the same manner as (2.7) by calculating the MSE with the Bellman equation. Their corresponding gradient calculation is expressed as (2.13) and (2.14), respectively.

$$\nabla_{\theta_a} J(\theta_a) = \nabla_{\theta_a} log\pi(a|s;\theta_a) A(s,a;\theta_a,\theta_c) \tag{2.13}$$

$$\nabla_{\theta_c} J(\theta_c) = \frac{\partial (A(s,a;\theta_a,\theta_c))^2}{\partial \theta_c} \tag{2.14}$$

The fundamentals of two different types of DRL algorithms have been introduced in this section, comprising important concepts, underlying learning mechanisms, and loss calculations for ANN parameter updates. These are the technical backbones for the designed algorithms in Chapters 4 and 5 of this thesis for interested readers, and more detailed information regarding these algorithms can be found in the original publications [105, 107, 110].

2.3.3 Deep Learning on Graphs

One of the key reasons for the success of deep neural networks is their ability to leverage the statistical properties of the input data. However, DL research has mostly focused on data within the 3-dimensional Euclidean space, such as acoustic signals, images, videos, etc [113]. Consequently, it motivated the extension of DL techniques into the non-Euclidean space and deep learning on graphs is one of the broad directions that receives increasing attention [114] and is a promising tool for wireless network research as such a network can be naturally modelled as a graph.

The model termed GNN was first introduced by [115] for pioneering research that extends the ANN's application to data in Euclidean space to the graph domain. With the message-passing mechanism introduced, GNN updates each node's state by exchanging information with other nodes until a stable state is achieved. With the development of ML techniques on a graph, variants of the GNN model were introduced, with Graph Convolutional Network (GCN) being a representative of their variants which extends the convolution operation of the CNN [116] from traditional data (e.g., an image) to graph data.

2.3.3.1 The Graph Data Structure

A graph is a powerful data structure to model a set of objects and their relationships. Wireless networks can be naturally modelled with graphs such as the V2V pair shown in Figure 1.1 where the two VUEs are modelled as “nodes" (the objects with relationships), with the wireless link between them modelled as an “edge" (the relationship between nodes). The graph representation of this V2V pair encompasses relationship information between the two VUEs with underlying network topology that is otherwise difficult to express using a Euclidean data structure such as a vector.

Formally, a graph $g = (\mathcal{V}_g, \mathcal{E}_g, \mathcal{X}_g, \mathcal{A}_g)$ is composed of a set of vertexes/nodes $\mathcal{V}_g$ and a set of edges/arcs $\mathcal{E}_g$ connecting pairs of nodes [117]. When the node pairs in g are unordered, g is referred to as an undirected graph, while ordered node pairs in g make it a directed graph. Figure 2.5 demonstrates the graph data structure with examples for both directed and undirected graphs.

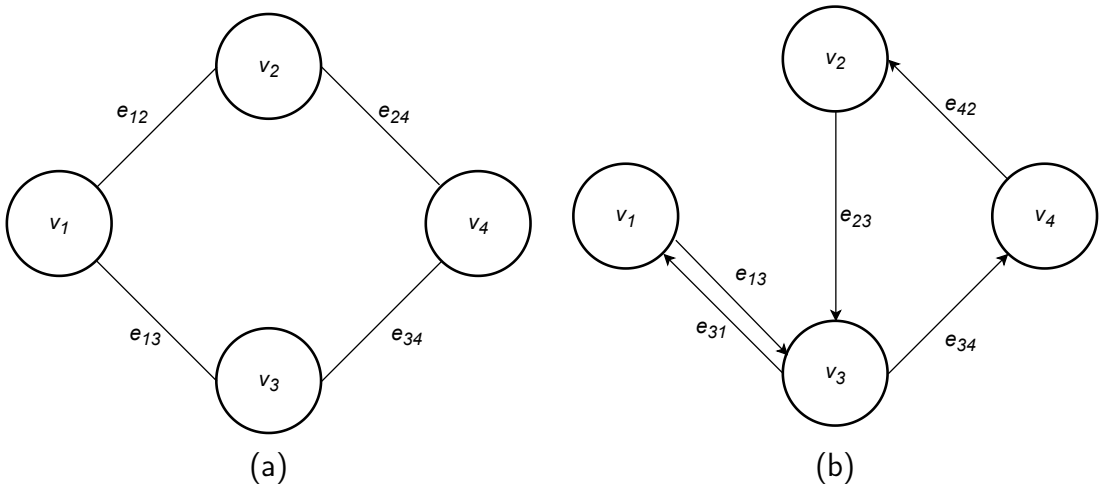


Figure 2.5: The graph data structure. (a) an undirected graph with 4 nodes and 4 edges, (b) a directed graph with 4 nodes and 5 edges.

To enrich the graph g, additional node and edge information can be included in $\mathcal{X}_g$ and $\mathcal{A}_g$ respectively. Each node $v \in \mathcal{V}_g$ is associated with a node feature $x_v \in \mathcal{X}_g$, while an edge $(\overline{u, v})$ (or e_{uv}) connecting a pair of nodes u, v $(u \neq v)$ holds an edge attribute a_{uv} $\in \mathcal{A}_g$.

2.4 Federated Learning

Originally, ML algorithms were designed for centralised setups (such as a cloud server) where datasets are well managed, distributed among local machines, and high-speed communication networks are available for fast data transmission. However, for vehicular networks, data is collected and stored in different network nodes, such as vehicles and pedestrians' mobile devices, leading to incomplete local datasets for these nodes. Since data-driven learning methods require a rich dataset to fully extract the underlying patterns, the partially observed data stored locally will cause individual learners to underperform in vehicular networks. Another scenario is when a learner joins a new environment where the learner has no knowledge of the environment at all and is destined to underperform. RL algorithms for vehicular networks also face a similar challenge in that vehicular decision-making agents can only interact with a part of the whole environment, leading to suboptimal policies due to partial observation. With the development of DL, deep ANN models become more complex with datasets of growing size, which give rise to challenges to the conventional learning architecture.

To solve such problems, distributed learning methods integrating different learners in the same environment for an enhanced dataset while also utilising the computing power of learners thus become a promising approach. The FL [118] methodology is well-suited to this and was first developed to exploit the values of distributed data among discrete learners while protecting learners' data privacy [119].

FL consists of two major components: The FL server and the participants. The participants are the actual learners that train their own ML models using locally available information, while the FL server is responsible to collect local models from each participant and aggregate them to form a global model. This information structure of FL is similar to the centralised MARL setting in Figure 2.3.

The general FL framework is shown in Figure. 2.6 which contains 3 main steps, Initialisation, Local Training, and Global Aggregation:

- **Initialisation**: the FL server determines the training setup, data requirements, and the participants for FL training.

- **Local training**: Each participant follows the configuration and trains a local model using its stored data.

- **Global aggregation**: All participants upload the parameters of their local models to the FL server. The FL server then aggregates the models to form a global model, e.g., averaging the local ANN models' weights to form a new ANN. The global model is then distributed back to all participants for a new round of local training.

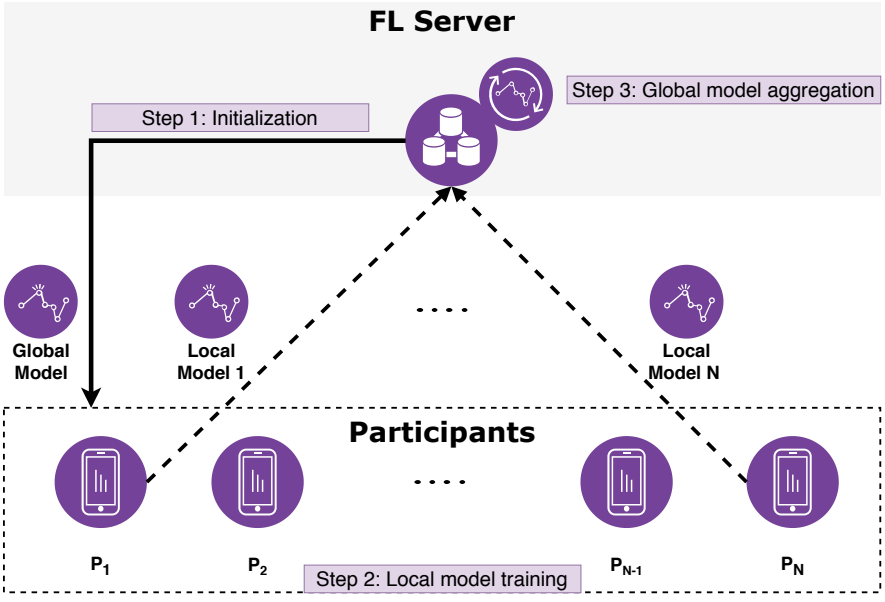


Figure 2.6: General FL training process with N participants

In addition to utilising distributed computing power and exploiting values from dis-

tributed datasets, the FL training flow also preserves the data privacy of participants as mentioned above. This is because only the local models are sent from the participants for global aggregations, while the data used to train the models are kept by the participants, which helps protect sensitive data from being shared or leaked.

2.5 Summary

This chapter presented an overview of vehicular networks and a tutorial on fundamental concepts and techniques of machine learning. To be specific, the characteristics and applications of C-V2X are first introduced, with an overview of use cases and related service requirements and KPIs. This background introduction set the scene for vehicular networks for the considered research topics which are further elaborated for each research question in the following chapters. For the ML tutorial, the core concepts and methodologies are covered for the three learning paradigms, i.e. supervised, unsupervised, and reinforcement learning, followed by the introduction of modern ML techniques including deep learning (and thus DRL and deep learning on graphs) and federated learning. These fundamentals are crucial to understanding the modelling of problems and the proposed algorithmic designs in Chapters 4, 5, and 6.

Chapter 3

Machine Learning Applications in Vehicular Networks

In this chapter, a literature review is presented covering the state-of-the-art solutions that apply different ML techniques in each of the three identified research topics, with a summary of the reviewed papers presented in Table 3.1. More specifically, a detailed background introduction is first presented for each research topic, followed by a review of the corresponding literature that applied ML techniques. Finally, an overview of adopted machine-learning techniques is presented for each identified research topic. Note that the analysis of research gaps is further introduced in Chapters 4, 5, and 6 for each detailed research question, respectively.

ML techniques have strong capabilities in optimisation problems and can adapt to dynamic environments, leading to a wide range of ML applications in vehicular networks [36, 120–123]. Three research topics were chosen (handover management, radio resource allocation, and energy efficiency on the network side) following the identified challenges when considering the underlying characteristics and emerging demands with the development of vehicular networks. These research topics also share a common theme of network optimisation, which is suitable for ML solutions combined with the dynamic vehicular network environment.

Since cellular communication networking is a broad research area even when restricted to vehicular use cases, an all-encompassing literature review covering every related topic is beyond the scope of this thesis. For a more detailed and comprehensive review of ML applications in vehicular and general cellular networking, there are multiple survey papers available in the field such as [37, 50, 51, 124].

Table 3.1: Summary of the literature review

Application Type	**Tasks**	**Covered Work**	**Type of utilised learning models/algorithms**
Handover Management	Handover parameter optimisation	[125, 126]	Q-learning [84]
	Predictive handover triggering	[127, 128]	Long Short-term Memory (LSTM) [129], Feed-forward ANNs, Q-learning, Hidden Markov Model (HMM) [128]
	Handover decision-making	[130–132]	Multi-arm bandits, Dynamic programming, Deep Q-learning, Multi-agent deep Q-learning, HMM
Resource Management	Spectrum, power, and joint radio resource allocation	[25, 133–138]	Deep Q-learning, Deep deterministic policy gradient [139], Feed-forward ANNs, GNNs, Meta-learning [140]
Energy efficiency via cell switching	RL-based direct decision-making	[54, 141–145]	Q-learning, Multi-agent Q-learning, Deep Q-learning, Multi-agent deep Q-learning
	Multi-tier solutions	[146–148]	K-means clustering [74], Feed-forward ANN, LSTM, CNN [116]

3.1 Learning in V2N Handover Management

In cellular V2X, the V2N links are very similar to those in conventional cellular networks, although a road vehicle has a much higher mobility compared with conventional cellular UEs such as a mobile phone used by a pedestrian. When moving across the coverage of different BSs, the fundamental requirement is that the connection between a VUE and the network does not break thus ensuring connectivity. This is accomplished by HO management in C-V2X systems that handle the moving VUE's network association from one BS to another BS.

3.1.1 HO Management in Cellular Networks

HO management is part of mobility management in cellular networks. In cellular networks, the term mobility is defined as "the ability for a user to communicate whilst moving independent of location", and the definition of mobility management is "a relation between the mobile station and the UMTS Terrestrial Radio Access Network (UTRAN)[1] that is used to set-up, maintain, and release the various physical channels." These definitions were defined by 3GPP for the vocabulary of cellular networks and are still valid for 5G [149]. Mobility management is a concept that emphasises the importance of maintaining connections for moving users and thus plays a crucial role in cellular networks.

HO management corresponds to the connected state mobility management. In a connected state, the connection between a UE and the network is established and

[1]Universal Mobile Telecommunications System (UMTS)

active with an ongoing data transmission, which is meant to be maintained without interruption or noticeable degradation while the UE travels within the network. As the UE's geographic position changes, it may lead to signal quality decrease from its serving BS and thus service quality, necessitating the UE to change the connection (be handed over) to another BS to maintain its QoS. The goal of HO management is hence to decide the target BS for the UE to switch connection to for the best service quality. Note that other objectives can also be fulfilled by HO such as re-associating UEs to balance BS traffic loads and improve spectrum and energy efficiency [150, 151]. However, this thesis will only focus on HO mobility management use case.

HO management is network controlled with UE assistance. The UE continuously measures the signal quality from its serving BS and neighbouring BS to form measurement reports that the network controller uses to make HO decisions. A straightforward metric to measure signal quality is the Received Signal Strength Indication (RSSI) which characterises the attenuation of radio signals during propagation [152]. This is a relative number measuring how strong a signal is when it is received by a device. In cellular networks, the UE measures the signal quality of BSs utilising other metrics related to RSSI, namely: RSRP, Reference Signal Received Quality (RSRQ), and SINR. These three metrics are the fundamental metrics in 5G cellular networks to infer signal strength and quality from a BS to UE for HO decision-making [153]. As defined by 3GPP [154], RSRP is the average power of resource elements that carry BS-specific reference signals within the measurement frequency bandwidth, while the RSRQ also includes channel interference and thermal noise. The relationship among RSSI, RSRP and RSRQ is shown as (3.1).

$$RSRQ = N_{rb} \times \frac{RSRP}{RSSI} \tag{3.1}$$

where N_{rb} is the number of resource blocks over the carrier RSSI measurement bandwidth. Finally, the mathematical definition of SINR is expressed as follows, reflected by its name:

$$SINR = \frac{S}{I + N} \tag{3.2}$$

where S, I, and N are the received signal, interference, and noise power at the receiver side, respectively. Note that these metrics are all used as signal quality identifiers from the networking standpoint although they show signal quality from different perspectives, and this thesis will not further expand on the detailed physical definitions for RSRP and RSRQ.

The measurement reports are sent to the serving BS for HO decision-making to ensure the UE connects to the best cell [120, 155, 156]. There are two types of measurement reporting: periodic and event-based [120]. As these names suggest, the UE performs and reports its measurement at a defined frequency in a periodic scheme. In contrast, an event-based measurement report will be sent to the serving BS upon some event-triggering criteria being met. These criteria defined in the 3GPP specifications are summarised as follows and note that the opposite of the conditions indicates the leaving condition of these events [157–159]:

- **Event A1**: Serving cell becomes better than an absolute threshold.
- **Event A2**: Serving cell becomes worse than an absolute threshold.
- **Event A3**: Neighbour cell becomes better than a defined offset relative to the serving cell. For example, the RSRP of a neighbour cell becomes 3 dB higher than that of the serving cell for a UE at a given position.
- **Event A4**: Neighbour cell becomes better than an absolute threshold.
- **Event A5**: Serving cell becomes worse than one absolute threshold, and the neighbour cell becomes better than another absolute threshold.

Two standardised HO algorithms are the A3-based or A2A4-based algorithms [160] as these three measurement events illustrate the situation when a potential HO is required.

A simple HO scenario consisting of two BSs and one VUE is shown in Figure 3.1. The network-controlled HO process starts after receiving the measurement report that triggers it. First, the serving BS check the availability of the target BS. After the target cell confirms that it has enough resources to serve the UE, the HO is executed and multiple signalling occurs among the UE, the serving BS, the target BS, and the core network. Finally, after completion, the target BS becomes the new serving BS of the UE and allocates resources to the UE, and the allocated resources for this UE by the previous serving BS is released.

The HO procedure among BSs is normally performed with the X2 interface in LTE [161] and Xn interface in 5G [161]. These two interfaces are similar with respect to HO workflow, and Figure 3.2 demonstrates the HO procedure using the Xn interface that the proposed HO algorithm in Chapter 4 is directly based on.

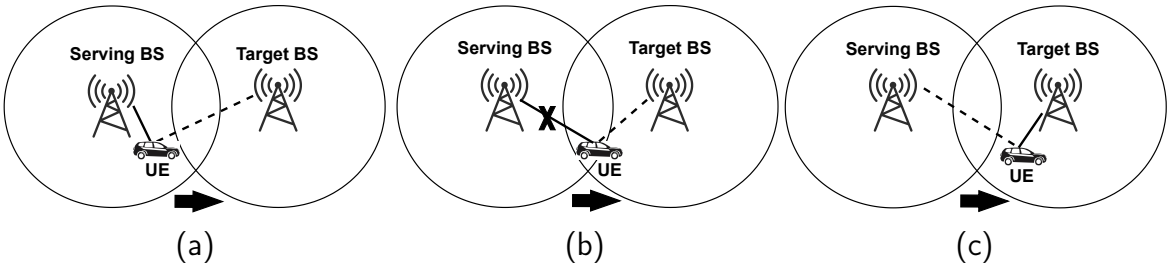


Figure 3.1: A demonstrative scenario of handover. The UE travels between two BSs with its direction indicated by the black arrow and measures the signal quality of both BSs. (a) Before the handover, UE connects to its serving BS. (b) During handover, the connection between the serving BS and UE is terminated while that between the UE and target BS is established. (c) After the handover, UE connects to the target BS.

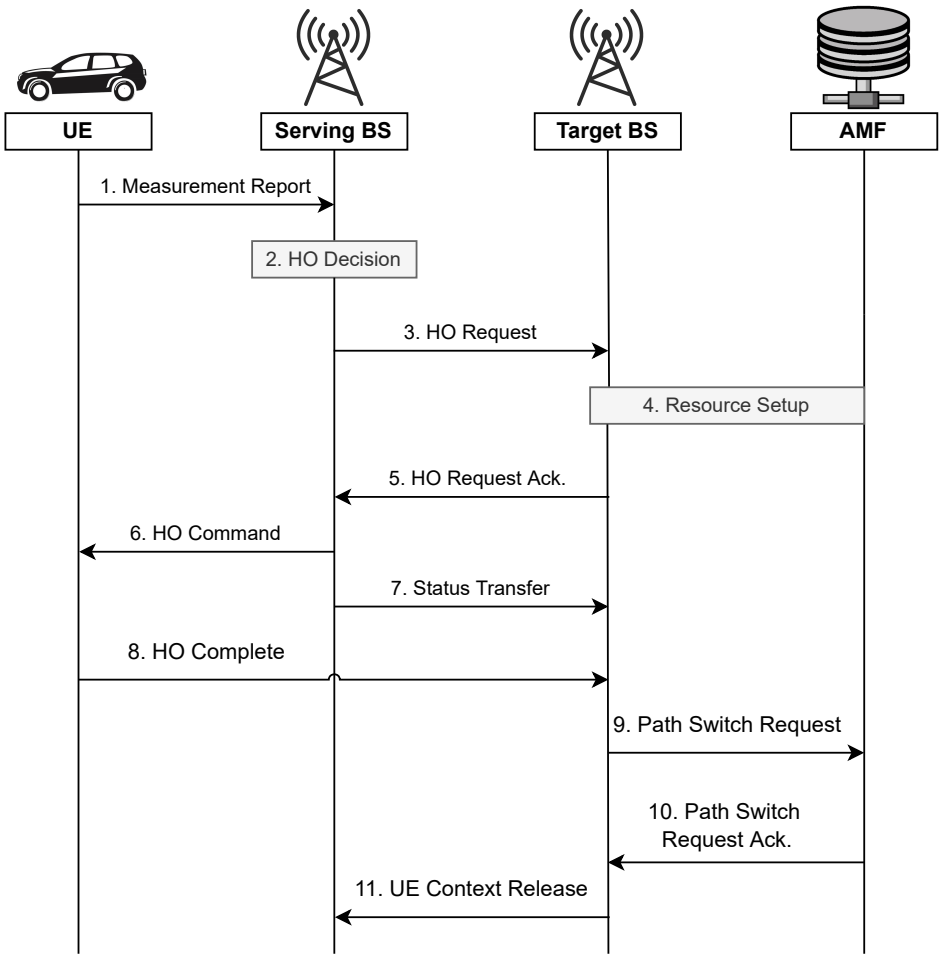


Figure 3.2: The Xn-based handover procedure in a cellular network. Adapted from Figure 9.2.3.2.1-1 of [162]. AMF: Access and Mobility Management Function.

Another important concept in HO is the HO parameters configured in conventional cellular HO algorithms to ensure a UE is handed over to a BS that provides high communication quality. A direct example of HO parameters is the offset and threshold in events A1 - A5 that define the borders of HO triggering. The hysteresis and Time-To-Trigger (TTT) are the other HO parameters that ensure a HO decision is made for a better target BS [163, 164] such that: the target BS is forced to have hysteresis' value of higher signal quality than the current serving cell after entering a measurement event (e.g., A3); the actual HO execution is delayed until the HO triggering condition is held true by TTT. HO parameters can also reduce ping-pong effects for cellular HO, meaning subsequent HOs happen between the serving and target BSs and vice versa, which significantly downgrades network performance [165].

There are various KPIs to evaluate HO's influence on the overall network performance, and some outstanding KPIs are presented as follows [166]:

- **HO interruption time**: The period of time during HO signalling when the UE cannot transmit the user's application data with the BS.

- **HO cost**: The average HO interruption time multiplied by the number of HOs for a particular UE's trajectory. This metric is directly related to the network throughput, with more detail introduced in Chapter 4.

- **Signalling overheads**: The various data generated during the HO process to facilitate the operation. Signalling overhead combined with HO interruption reduces the experienced throughput of a UE.

- **HO failure rate**: The number of unsuccessful HOs divided by the total number of HOs throughout a given UE trajectory or for unit time.

- **Overall service quality**: This can be evaluated using KPIs such as the experienced data rate or signal quality of a UE.

It can be seen from these KPIs that HO management is more challenging in vehicular networks with much higher mobility compared to UEs in conventional cellular networks, as the higher mobility naturally leads to a larger number of HOs per unit time. With ultra-dense deployments, the number of potential targets BSs to handover also significantly increases leading to an increased need for HO decision-making. Moreover, the inclusion of short-range mmWave technology to boost capacity complicates the radio environment for HO decision-making even more as it deals with the resultant narrower coverage and sensitivity to blockage.

3.1.2 Learning-based HO Management: The State of the Art

A variety of research has been conducted for HO performance improvement for different network setups [17, 166]. ML-based solutions for HO management can be classified into three main types: ML-based HO parameter optimisation, ML-based prediction assisted HO, and ML-based HO decision-making.

HO parameter optimisation aims to adaptively update the configured values of HO parameters as introduced above to achieve HO triggering and decision-making optimisation. RL is the major ML methodology used to optimise HO parameters. A Q-learning-based algorithm was proposed by [125]; by setting the reward function to consider the number of HOs, HO delays and throughput system-wise, the proposed algorithm optimised the values of TTT and hysteresis. Similarly, the work of Goyal and Kaushal [126] also utilised Q-learning to optimise two HO parameters: hysteresis and TTT. This scheme utilised several metrics of signal quality, the UE's location and moving direction, and the load on each BS in combination to rank neighbouring BSs for the Q-learning algorithm to make effective HO decisions.

In the area of ML-based prediction assisted HO, prediction algorithms that accurately predict metrics for HOs or the future geographic location of a vehicle are utilised to proactively trigger a HO and reduce the complexity of HO decision making. The algorithm proposed in [127] could proactively trigger optimal HOs by matching the vehicle's predicted future location with known BS locations using a HMM [128]. More recent research also implemented a DL-based method using an RNN structure termed Long Short-time Memory (LSTM) [129] and fully connected feed-forward ANNs to jointly predict a vehicle's future trajectory according to personalised driving styles [131]. Besides mobility prediction, there are also methods for HO metrics prediction. An FL training setup was utilised in [132] to predict future signal quality in a mmWave vehicular network by utilising both the BS and local UEs. The predicted values were then fed to a conventional HO algorithm to proactively trigger HOs.

As for ML-based HO decision-making, an ML algorithm is directly utilised for HO decision-making upon a HO trigger that replaces a conventional HO algorithm. RL is the dominant ML methodology to develop solutions in this category for its advantages in sequential decision-making. Note that other ML techniques can also be utilised in addition to RL. An RL HO algorithm was proposed [167] using non-stochastic bandit theory [168] with HO cost included in the utility function to reduce frequent HOs and the overall system energy cost. A unified HO algorithm was developed for Long-term Evolution Advanced (LTE-A) systems in [169] based on discrete stochastic dynamic programming [170]. This algorithm considered both the UE's measurements of signal quality and BSs overall resource utilisation to produce HO decisions that also balance loads of BSs. DRL algorithms are also utilised for HO decision-making. For example,

Mollel *et al.* developed a deep Q-learning based HO algorithm to calculate adaptive HO decisions in a mmWave vehicular network [171]. Their work used event A2 to trigger HOs as it could indicate a blockage in mmWave networks while also skipping state observations that are relevant to HO conditions as those observations not satisfying event A2 will be filtered out thus not used. A K-means clustering algorithm was developed in [172] to cluster UEs based on the mobility pattern, followed by an asynchronous multi-agent DRL algorithm for optimal HO decision-making. Furthermore, a joint HO and power allocation scheme was developed for heterogeneous networks utilising multi-agent DRL [173]. Using a reward design based on system throughput and introducing a HO penalty, the algorithm calculated the optimised target BS and transmit power for each UE.

There are some HO research that combined different ML-based HO optimisation techniques to form multi-tier learning-based HO solutions. For example, a long short-term memory-based RNN was trained in [174] to predict future signal strength that proactively triggers HOs. After triggering, a HMM was used for the HO decision-making.

3.2 Learning in Resource Management

Wireless communication networks have been continuously pursuing higher throughput, lower latency, higher reliability, and better coverage, while also suffering from a variety of impairments including pathloss, jamming and/or multi-user interference [121, 175]. To achieve these goals and deal with the issues, efforts have been paid to aspects including more efficient modulation and coding design, improved detection and decoding schemes, with dynamic and effective management of communication resources also identified as a promising solution [176]. Resources in communication systems are limited such as the scarce spectrum resource available, making it more important to optimise the allocation of the limited resources to satisfy the service requirements of the drastically increasing devices [175].

3.2.1 Spectrum and Power Allocation

According to Shannon's information capacity theorem [177], power and bandwidth are the two essential resources which determine the capacity of a wireless channel indicating the upper bound of information transmission rate with an arbitrarily small error rate. Therefore, it is critical to allocate frequency bands and/or transmit power levels according to the channel conditions and users' QoS requirements in vehicular networks.

The basic idea of a spectrum allocation problem can be illustrated with a simplified scenario that considers one user accessing a set of N_{sb} sub-bands under different

conditions at time slot t [178]. Each sub-band has a binary condition of "good" (i.e. small interference and noise) or "bad" (i.e. large interference and noise) that can vary with time. Therefore, the user's goal is to select a sub-band under "good" condition for successful data transmission for each time slot $t \in \{1, 2, ..., T\}$ to maximise KPIs such as throughput. This simplified scenario is shown in Figure 3.3 and can be extended to more generalised spectrum allocation situations with multiple users and/or more complicated channel conditions.

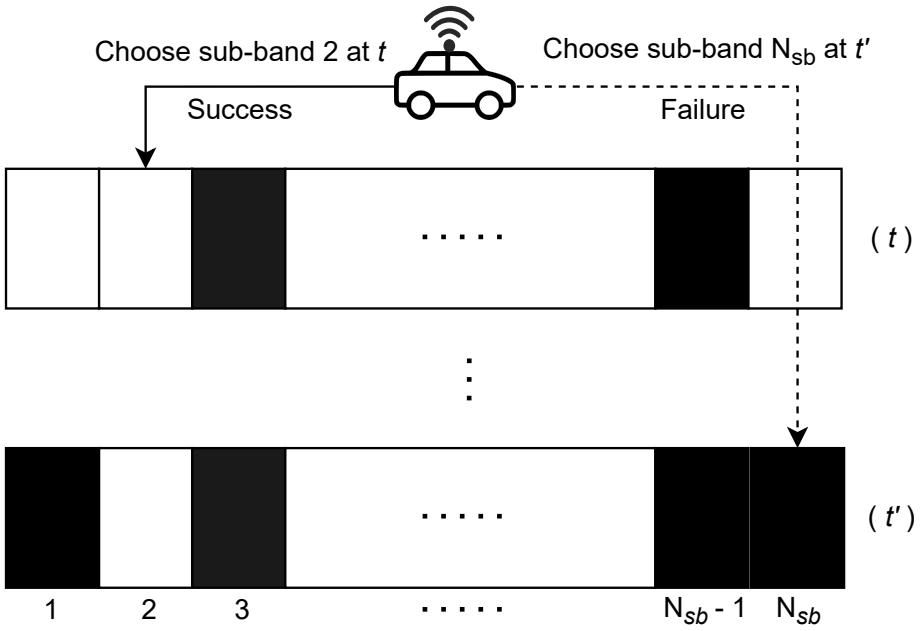


Figure 3.3: A simplified spectrum selection scenario for a single VUE with N_{sb} sub-bands. A white block indicates a sub-band in "good" condition and a black block indicates a "bad" sub-band.

Power allocation concerns the transmit power adaption corresponding to the channel and user conditions to control interference and improve a predefined performance metric [121]. To formulate this power allocation problem, consider a simplified interfering channel with M communication links sharing one spectrum sub-band, aiming to optimise the sum throughput of all links. Denote the transmit power of link m's transmitter as P_m ($m \in \{1, 2, ..., M\}$) with fading channel gain $g_{m,n}$, and the fading channel gain from the transmitter of link n ($n \neq m$) to the receiver of link m as $g_{n,m}$. The received SINR of link m's receiver can then be calculated:

$$SINR_m(\bar{P}) = \frac{P_m g_{m,m}}{\sum_{n \neq m} P_n g_{n,m} + \sigma^2} \tag{3.3}$$

where $\bar{P} = \{P_1, ..., P_M\}$ is the set of transmit power of all links and σ^2 represents the noise power. The power allocation optimisation's goal is for each link m to adjust its transmission power P_m within the maximum power range ($0 \leq P \leq P_{max}$) to maximise the system throughput expressed as:

$$\max_{\bar{P}} \sum_{m=1}^{M} \log(1 + SINR_m(\bar{P})) \tag{3.4}$$

Note that spectrum and power allocations can be jointly considered to improve spectral efficiency, reduce interference, and optimise metrics such as system throughput and transmission latency.

3.2.2 Resource Allocation in C-V2X

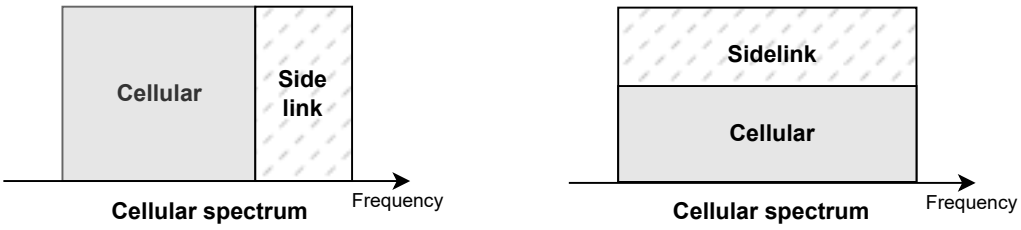
In C-V2X, V2N is supported as conventional cellular communications via cellular links, while direct communication links between UE-type devices such as V2V and V2P are supported by cellular sidelinks [179, 180]. There are two spectrum access modes for cellular sidelink: inband and outband modes [181]. Outband sidelink communications use unlicensed spectrum bands and adopt other wireless technologies such as Bluetooth [182]. In contrast, the inband setting utilises the licensed cellular spectrum bands for both sidelink and cellular link communications, which is the selected type for this thesis for its full control over resource allocation. As shown in Figure 3.4, inband sidelink can be further categorised into two types, summarised as follows [181, 183]:

- **Underlay**: Sidelink communications fully reuse the cellular spectrum bands. The licensed spectrum is not divided for sidelink and cellular communications, resulting in a higher spectrum efficiency but dedicated resource allocation is required between sidelink and cellular users to mitigate impairments such as interference through power control, spectrum allocation, or joint allocation of spectrum and power [183].

- **Overlay**: In this design, the cellular spectrum is divided into non-overlapping parts for sidelink and cellular communications, respectively. As a result, the overall implementation is simpler compared with underlay sidelink, with interference also reduced between cellular and sidelink users. However, there is potential spectrum resource wastage due to the divided spectrum.

In addition to spectrum access modes, C-V2X also has two resource allocation modes in 5G, supporting both centralised and decentralised resource allocation scenarios for vehicular networks [184–186]:

- **Mode 1**: VUEs communicate via sidelink but resource allocation is managed by a BS in a centralised manner. This mode is only available when a VUE is within cellular network coverage.



(a) (b)

Figure 3.4: Different spectrum access modes for inband cellular sidelink. (a) Overlay mode. (b) Underlay mode. Note that the graph is one-dimensional demonstrating how spectrum frequency bands are utilised, and the presentation of the graph is only to improve visuals.

- **Mode 2**: VUEs communicate via sidelink and autonomously select their radio resources pre-configured by BS/network. As a result, Mode 2 supports both under and out-of-cellular network coverage.

3.2.3 Learning-base Resource Allocation: The State of the Art

There are several approaches for learning-based resource allocation and can be classified based on the underlying learning paradigm, i.e., supervised, unsupervised, and RL [121].

Supervised approaches for resource allocation optimisation treat the optimisation problem as a relationship between the input parameter and the output resource allocation decision and train an ML model to represent such relationship for optimised decision-making [121]. As resource allocation optimisation has been previously tackled with conventional optimisation algorithms, the results of these algorithms can be used as labelled data to train the ML model. For example, a deep feed-forward ANN was trained by Sun et al. [133] for power allocation in a basic wireless network with interfering channels utilising the output of the Weighted Minimum Mean Square Error algorithm proposed in [187]. Their results suggested the trained ANN can deliver performance close to the original heuristic algorithm. A GNN model was proposed in [134] for link scheduling, a special case of power allocation that considers on/off power control of communication links, for a dense sidelink communication network. The GNN model was trained using the data generated by a sub-optimal algorithm, FPLinQ [188], and the results showed satisfactory performances with good generalisation ability to different network topologies.

Since another algorithm's results are used to train an ML model in supervised approaches, the performance of such algorithms is crucial for the learning outcome. Moreover, it is expensive to generate sufficient high-quality data [121]. These disadvantages motivated the use of unsupervised approaches to directly optimise the target metric by treating it as the algorithm's cost function using unlabelled data. A deep ANN-based power

control method was proposed in [135] which aimed to solve the non-convex optimisation of maximising the sum rate of a fading multi-user interference channel. The sum rate was directly used as the loss function of their proposed algorithm which is similar to that expressed in (3.4). Similarly, a GNN-based link scheduling solution was proposed in [136] which also aimed to maximise the sum data rate in an interfering sidelink network. Their results indicated close performance compared to a supervised learning counterpart while not requiring expensive labelled data.

Different from supervised and unsupervised approaches, RL for resource allocation problem models the network controller or UEs as RL agents to learn optimal resource allocation by interacting with the environment, and most conducted work on ML-based resource allocation optimisation lies in this category [37]. For example, a game-theoretic model was proposed in [189] to optimise power allocation in a vehicular network, which is solved by a Q-learning-based algorithm to minimise VUEs' energy consumption. A joint sub-band and transmission power level for V2V links was proposed in [137] for a spectrum-sharing vehicular network between V2V and V2N links. A centralised deep Q-learning algorithm was developed for the network controller to decide each V2V link's transmitting sub-band and transmit power. Moreover, Liang et al. developed a DRL algorithm for joint sub-band and V2V transmission power allocation in a spectrum sharing C-V2X with Mode 2 allocation [25], with a multi-agent deep Q-learning-based solution developed for each VUE to select its own transmitting sub-band and power. Similarly, a DRL solution for joint sub-band and V2V transmission power control was developed by Yuan et al. [138]. However, to further improve the adaptability for different communication scenarios, a meta reinforcement learning algorithm [140] is proposed for fast tuning the DRL model for resource allocation scheme.

3.3 Learning in Energy Efficiency

For vehicular and cellular networks in general, energy efficiency is crucial to reduce energy wastage, Operational Expenses, and greenhouse gas emissions to achieve the goal of green and sustainable network deployment. Different approaches exist for energy efficiency in vehicular networks, aiming at two main scenarios: life-span extension for self-powered devices and power efficiency improvement for power grid-supplied Radio Access Networks (RAN) [36]. These aspects aim to improve energy efficiency on the network side. Since VUEs have sufficient energy for communication components, energy efficiency improvement for vehicles focuses on aspects such as improving road traffic efficiency by reducing fuel wastage and optimising driving style for motors to operate more efficiently [36].

The first type of scenario's aim is to increase the operational time of battery-powered devices such as RSUs deployed in rural areas that support local road traffic. Two major

methodologies for life-span extension are energy conservation and energy harvesting [190]. Energy conservation aims to reduce a device's power wastage to extend its battery life as much as possible. In comparison, devices under energy harvesting schemes are equipped with additional equipment (e.g., wind turbines, solar panels, etc.) to collect energy from a secondary energy source [191]. ML techniques have also been applied to this aspect of energy efficiency in vehicular networks [36], with use cases including switching a self-power RSU into sleep mode when idle [192], energy-efficient scheduling for computation offloading with energy-constrained RSUs [193], and wind energy harvesting-based energy trading between RSUs and electric vehicles [194], etc.

Although life-span extension for battery-powered devices is an important scenario to investigate, this thesis focuses on the other category of scenarios to improve the energy efficiency of power grid-supplied RAN components (e.g. BSs). As introduced in Chapter 1, BSs are the major energy consumer in cellular networks and take up 60% - 80% of energy consumption in the network, with 5G BSs consuming more than 1.5 times more energy compared to their predecessors [43, 44]. As a result, it is vital to improving BSs' energy efficiency to achieve green communications and sustainable network development.

3.3.1 Base Station Components

A well-known and fundamental power consumption model for cellular BSs was developed by Auer et al. [45], referred to as the Energy Aware Radio and neTwork tecHnologies (EARTH) power consumption model. According to this model, the typical components that contribute to a BS's power consumption are:

- **Antenna interface**: The antenna is the component for signal transmission and reception. The influence of the antenna type on the power efficiency is modelled by a certain amount of losses caused by the feeder, antenna bandpass filters, duplexers, and matching components [45].

- **Power amplifier**: This is used during the transmission and reception of radio frequency signals to boost the signals' power level.

- **Small-signal radio frequency transceiver**: This consists of a transmitter and receiver for downlink and uplink communications, respectively.

- **Baseband unit**: Responsible for baseband signal processing, including modulation, conversion, filtering, etc. Carrying out such tasks leads to additional energy consumption [45].

- **Power supply and cooling**: This unit contains the mains (Alternating Current (AC)-to-Direct Current (DC)) supply, DC-DC power supply, and the active cooling

system of a BS (mainly for the macro-BS type).

Following the introduction of the above main components and assuming the power consumption of a BS is proportional to the number of transceiver chains N_{TRX}, a BS' total power consumption P_{in} is expressed by the following equation according to [45]:

$$P_{in} = N_{TRX} \times \frac{\frac{P_{out}}{\eta_{PA} \times (1-\sigma_{feed})} + P_{RF} + P_{BB}}{(1-\sigma_{DC})(1-\sigma_{MS})(1-\sigma_{cool})} \tag{3.5}$$

where η_{PA} denotes the power amplifier's power efficiency, σ_{feed}, σ_{DC}, σ_{MS}, and σ_{cool} stand for the coupling losses from the feeders, the DC-DC converter, the mains supply, and the cooling systems, respectively; P_{out} represents the average transmit power for each transceiver link of the BS, P_{RF} and P_{BB} are the power consumptions of the radio frequency and baseband unit, respectively. More detailed BS power consumption breakdown can be found in the original paper of the EARTH model [45].

According to the analysis of the results in Figure 3 of [45], the relations between P_{out} and P_{in} are nearly linear. Therefore, a linear approximation of (3.5) as a simplified BS power model is often used for green radio analysis [195]:

$$P_{in} = \begin{cases} N_{TRX} \cdot (P_0 + \Delta_p P_{out}), & 0 < P_{out} \leq P_{max} \\ N_{TRX} \cdot P_s, & P_{out} = 0 \end{cases} \tag{3.6}$$

where P_0, P_s are the static operational and sleeping power consumption, respectively, Δ_p is the slope of the load-dependent power consumption, and P_{max} denotes maximum transmission power.

3.3.2 Cell Switching-enabled Energy Saving

As BSs are the major power consumers in cellular networks and various techniques have been developed for BS energy-saving. As summarised in [195], energy-saving techniques can be grouped into several domains, namely: Spatial domain, time domain, frequency domain, and engineering domain, corresponding to optimising different variables in (3.5). The methodologies and representative energy-saving techniques from each of these domains are presented as follows:

- **Spatial domain**: This is to minimise the equivalent number of transceiver chains N_{TRX}, i.e., to switch off transceiver chains or entire cells of a BS as many as possible or deploy fewer transmitters in the first place. P_{in} can be proportionally reduced with the decrease of N_{TRX}, and when all cells of a BS are switched off,

the BS can also be switch to sleep mode and operates in P_s which is lower than its active power $P_0 + \Delta_p P_{out}$ as in (3.6).

- **Time domain**: This is to extend the time when a transceiver is switched off or a BS is in sleep mode. Since $P_s < P_0 + \Delta_p P_{out}$ in (3.6), the longer a BS can stay in sleep mode, the more energy can be saved. The cell discontinuous transmission technique [196] is an efficient approach to timely switch off the traffic channels and the related power amplifier, which can significantly extend the sleeping periods [195].

- **Frequency domain**: To reduce the average transmit power requirement P_{out} of a BS and increase transmission efficiency. This straightforward aim is achieved by bandwidth reduction techniques as described in [196]. The reason behind this technique is that the radiated power scales with the bandwidths requirement, while smaller bandwidth also requires less reference signalling, which lowers the power budget. The technique can be accompanied by radio resource management, such as power allocation to maximise its value [195].

- **Engineering domain**: Engineering evolution can also help improve BS energy efficiency by increasing the efficiency of hardware via advanced architectural design and/or improved components as in [197–199]. The aim is to improve, e.g., the power amplifier efficiency η_{PA} and the coupling losses consisting of σ_{feed} and σ_{cool}

From the above summary, switching off components in a BS or putting the BS into sleep mode (referred to as cell switching in this thesis) is a highly effective strategy for the proportional energy saving according to (3.5) that has a high impact among all BS energy efficiency techniques. Additionally, the cell switching technique also stands out for the following reasons: 1) It can be deployed on top of other energy efficiency techniques such as engineering solutions (e.g. improved power amplifier) and transmission efficiency schemes; 2) It is more convenient to deploy with software updates, compared with the engineering solutions requiring component replacement or even new deployment, or frequency domain solutions that need updated communication standardisations BS [200]. Finally, cell switching can be adaptively designed and executed with different levels of sleep mode correlating to the deactivation time for different scenarios. A deep sleep mode can switch off more components (e.g., from transceiver pairs to active cooling) and achieve greater energy saving while having a longer deactivation time which is less flexible. In contrast, shallow sleep mode will result in less saved energy by switching off fewer components (e.g. only a few transceiver pairs) but is more responsive with a shorter deactivation/reactivation time [201].

3.3.3 Learning-base cell switching: The State of the Art

Previously, optimising cell-switching decision-making was developed using mathematical optimisation-based heuristic algorithms. In recent years, cell switching solutions have been adapting ML techniques because of the advantages of ML. These solutions can be classified into two main categories drawing from the state of the art: Direct cell switching decision-making algorithms that mainly utilise RL, multi-tier solutions combining different ML techniques or utilising ML and heuristic algorithms jointly. Since cell switching considers the energy efficiency of BSs which are important components for cellular networks as a whole, research for both general cellular networks and the more specific vehicular networks are included here. Note that energy efficiency considering only vehicular networks also exists such as the aforementioned work of [192–194] which is not the focus of this topic, with more details covered in [36].

Tabular Q-learning is the frequently utilised RL algorithm for ML-based direct decision-making for cell switching. For example, Yu, Chen, and Yin proposed a Q-learning-based, dual-threshold solution to put SC BSs into sleep mode for energy saving while avoiding the SCs' frequent transactions between sleep and operation modes [141]. Assaid *et al.* also proposed a Q-learning-based method [142], which considered energy consumption with CO_2 emission tracing in a 5G network. Another Q-learning-based solution was proposed in [143] to dynamically adjust BSs' load and to switch off redundant SCs in an ultra-dense cellular network setup. Multi-agent RL was also utilised, such as the Q-learning algorithm proposed in [144] that modelled each SCs as an RL agent to jointly learn the best sleep mode option in a multi-sleep-mode setup in order to maximise the network's energy efficiency.

In addition to tabular RL techniques, approximation-based RL techniques were also utilised by recent research. For instance, the linear function approximation technique was utilised by Ozturk *et al.* [54] to develop online training and execution in an ultra-dense network. Another approach is to exploit the strong approximation capability of ANNs, such as the work of Zhang *et al.* in which a double deep Q-network-based solution was developed to determine the optimal sleeping strategy in a heterogeneous radio access network [145] and real-world traffic data was used trained and tested the algorithm.

As for multi-tier solutions, some research combined ML methods to boost the performance of heuristic algorithms or to reduce the problem search space and hence the overall problem complexity. Abubakar *et al.* proposed a two-tier cell switching scheme [146] based on unsupervised learning and the exhaustive search algorithm [202]. Their solution first separated an ultra-dense network into different clusters using the K-means clustering algorithm; after that, the exhaustive algorithm was executed for each cluster to get optimal local cell-switching decisions. A LSTM model was utilised by Jang *et*

al. [147] to predict user traffic for the next few time slots of the network. Based on the predicted traffic, a Lyapunov optimisation problem [203] was formulated to obtain the cell-switching decision to balance between the reduced power consumption and the predicted traffic loads.

Moreover, different learning-based techniques can be jointly utilised for cell-switching decision-making, such as the work in [148] that first combined CNN and RNN to predict future traffic. After that, the cell switching problem was modelled as an MDP and solved by the Deep Deterministic Policy Gradient (DDPG) method [139], a DRL algorithm.

3.4 ML Techniques Adopted for Implementation

With the research gap identified for each research topic from the literature review, this thesis adopts different ML techniques for solution proposals. The adopted ML techniques for the three research questions are centred around Deep Learning and, more specifically, DRL and GNN-based algorithms that belong to the categories of reinforcement learning and unsupervised learning. Detailed ML techniques utilised for implementation are summarised as follows on a per-chapter basis. Note that the Pytorch learning framework is utilised to implement all the proposed learning algorithms [204].

- **Chapter 4**: A centralised DRL algorithm of Deep Q-learning is employed for HO decision-making.

- **Chapter 5**: A decentralised DRL algorithm utilising FL training is implemented for joint spectrum and power allocation. This algorithm has two versions, Deep Q-learning and A2C, to examine the performance of on-policy and off-policy RL algorithms.

- **Chapter 6**: A GNN-based DL algorithm is proposed and implemented in an unsupervised manner for adaptive cell switching.

3.5 Summary

Following the general background introduction in Chapter 2, a more detailed background regarding the three chosen research topics, handover management, resource allocation, and energy efficiency, is presented in this Chapter. For each research topic, such background includes the fundamental concepts (e.g., the definition of handover) and goals, and the related system architecture/model considered. Following this, a literature review for each research topic is presented, drawing the state-of-the-art. The background and literature review will be further elaborated in the following chapters (Chapters 4,

5, and 6) for each research question, motivating the proposed solutions to the research questions. Finally, a brief summary of adopted ML techniques implemented for each research question is covered in a per-chapter manner.

Chapter 4

Intelligent Handover Algorithm with Deep Reinforcement Learning

4.1 Introduction

Network densification is a key solution to meet the rising demand for data traffic and to serve the dramatic increase in demand for cellular services. With the development of 5G and beyond, the substantially increasing numbers of BSs due to ultra-dense deployment and connections with more HO entities to be considered will lead to highly complex HO management. In addition, the high mobility of road vehicles will cause a much shorter connection time between a VUE and a BS, further increasing the frequency of HO. These characteristics bring new challenges to traditional solutions with fixed parameters for HO decision-making.

4.1.1 Background: System Model and Problem Formation of Cellular V2N Handover Optimisation

The considered network architecture of cellular V2N communication in this chapter consists of VUEs and BSs for HO management optimisation. The scenario can be illustrated by the simple scenario in Figures 3.1 and 4.1, which consist of only road vehicles and BSs with V2N as the only communication type. Once the simplified scenarios are understood, they can be extended to include densely deployed BSs, multiple VUEs, and other communication entities to formulate a more realistic real-world scenario. Note that the detailed general background for cellular networks' HO management is already covered in Section 3.1 hence will not be repeated in this section.

4.1.1.1 The Current HO Algorithm for Cellular Networks

In current cellular systems, the HO solution is designed using fixed parameters combined with different measurement events following the introduction of the cellular HO mechanism in Section 3.1 and a standardised HO algorithm implemented in cellular networks is the A3-based HO algorithm [205]. To recapitulate, event A3 will be triggered if the signal strength (e.g. RSRP) of a neighbour BS becomes larger than that of the serving BS by a predefined offset including measurement object specific offset, the event offset, etc., with the mathematical definition of the entering/leaving conditions defined in the corresponding 3GPP specification [158]. In this thesis, however, it is assumed that the offset values are represented by the hysteresis.

Figure 4.1 demonstrates the concept of A3-based HO considering a simple simulated scenario where a VUE travels along a straight road at a constant speed v between two BSs deployed. The scenario shown by the bottom part of this figure largely resembles what is demonstrated in Figure 3.1, while the top part of this figure illustrates the RSRP records of the two BSs along the trajectory of the VUE. The VUE is initially connected to BS_1 and is handed over to BS_2 at around 37 seconds of simulation time when the RSRP of BS_2 is 3 dB larger than that of BS_1 according to the decision calculated using the A3-based HO algorithm. Since this algorithm uses RSRP as the signal quality indicator (the larger an RSRP is, the better the signal quality), it is termed the A3 RSRP HO algorithm.

In this figure, t_0 represents the ideal HO point in the demonstrative case; i.e., the RSRP of BS_2 exceed that from BS_1 after t_0. If the HO can be completed at t_0, this will ensure that the VUE experiences the theoretically best signal quality throughout this simplified scenario. Δ and TTT represent the two HO parameters, hysteresis and TTT. As already discussed in Chapter 3, these two HO parameters are set to ensure that a better BS is selected for HO and to overcome the ping-pong effect [206] by the following process: In this simplified scenario, BS_2 must hold a higher signal quality of Δ than BS_1 after event A3 is entered (this equals to "after t_0" in this example) for at least a time of TTT before executing the HO. This is process is also the fundamental process of the A3 RSRP algorithm when measuring RSRP for signal strength indication. Finally, δ_t represents the delayed time of this HO with respect to t_0. During δ_t, the VUE experiences suboptimal signal strength while waiting for the confirmation of HO by the A3 RSRP algorithm as guarded by the two HO parameters.

4.1.1.2 The Considered Problem of HO Decision-making Optimisation

In cellular networks including LTE and the current 5G implementation, hard HO is applied such that the connection between the UE and its serving BS is terminated before the new connection is established [207] as demonstrated in Figure 3.1. As a

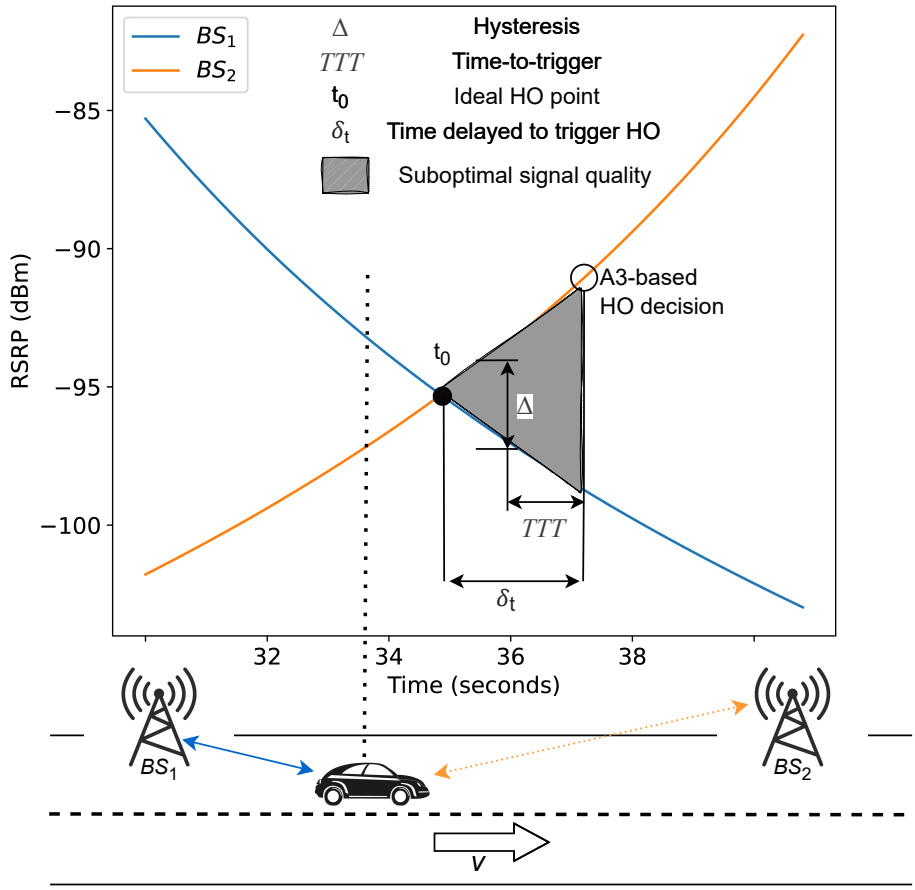


Figure 4.1: Event A3-based HO. In this case, t_0 identifies the optimal HO point or the UE will experience suboptimal signal quality due to HO delay as specified by the grey area. Δ and TTT represent the two parameters, hysteresis and TTT, of the A3 RSRP HO algorithm; δ_t indicates the delayed time (period of suboptimal signal quality due to late HO) for this HO with respect to t_0.

result, the UE's application data transmission will be interrupted for a short period during the hard HO process. This interruption occurs after the HO execution point (e.g., shown by the circle mark in Figure 4.1 for the simplified scenario), is known as the HO interruption time t_d. Note that t_d is different from the delayed time for HO (δ_t) that denotes the time between t_0 and the A3 HO trigger.

The accumulation of t_d will cause a degradation effect on the experienced throughput of the UE due to hard HO. Note that under an idealised condition, t_d will be a constant defined by the system configuration as the time consumption of the HO execution phase specified by steps 7 to 11 in Figure 3.2 is a constant. The accumulation of t_d related to the total number of HO N_{ho} in a given UE trajectory are combined as HO delay cost D_{HO}, and is defined as follows according to [208]:

$$D_{HO} = N_{HO} \times t_d \tag{4.1}$$

For a unit time period T (seconds) of a given moving UE, the normalised HO delay cost β_{HO} can then be derived:

$$\beta_{HO} = min(\frac{D_{ho}}{T}, 1) \tag{4.2}$$

β_{HO} is expressed as a factor between 0 and 1, indicating the percentage of total time consumed on HO operations such as radio link switching between BSs. As β_{HO} tends towards 1, it indicates that the UE has spent almost the whole period T on HOs such that the user will experience no effective connections for service data transmission. Ping-pong effect during HOs that the cellular connection of a UE shifts forwards and backwards between two or more BSs will greatly increase β_{HO} by increasing N_{HO} thus transmission interruption and largely degrade the service quality.

The throughput derived from the Shannon capacity formula when considering β_{HO} can then be expressed as:

$$Throughput = (1 - \beta_{ho}) \times \sum_{t}^{t+T} BW(t) \times \log_2(1 + SINR(t)) \tag{4.3}$$

where $BW(t)$ and $SINR(t)$ are the instantaneous bandwidth and signal-to-noise ratio at time t. Therefore, maximising *throughput* indicates maximising $BW(t)$ and $SINR(t)$ while minimising β_{HO} for T. If constant $BW(t)$ and t_d are considered, $SINR(t)$ and N_{HO} then play a vital role in increasing the average throughput for a defined time period T along a known UE trajectory, which is the fundamental assumptions of this work as the focus is on the HO decision-making. The optimisation of HO execution (beyond step 7 in Figure 3.2) is thus beyond the scope of this chapter. For simplicity, the rest of this chapter will use $SINR$ and BW instead of $BW(t)$ and $SINR(t)$.

Therefore, given T and a trajectory in a hard HO setup, the optimisation objective is to maximise average throughput by optimising the HO decision-making to increase the average $SINR$ by switching to a better-serving BS while also reducing N_{HO} for a constant BW and t_d. It is acceptable to have more HOs to further increase $SINR$ as long as the resultant $Throughput$ also increases following (4.3). Note that the goal to maximise $SINR$ is essentially to minimise δ_t following the assumption of constant BW and t_d to reduce the time for suboptimal signal quality during a HO period of interest. Therefore, δ_t becomes an important KPI in this work as it indicates the period of service quality downgrade due to late HO execution. When using the A3 RSRP algorithm in a vehicular network with ultra-dense BS deployment, frequent HOs will lead to a large accumulation of δ_t that will significantly downgrade the user experience

due to packet loss caused by the suboptimal signal strength. Note that for a defined HO algorithm (e.g., the A3 RSRP algorithm with known Δ and TTT), δ_t will also be a constant following the above assumption as the decision of a HO would also be predefined, hence this optimisation problem is dependent on the algorithmic design and parameter configurations for optimal HO decision-making.

Fixed-parameter-based conventional HO algorithms are hard-coded and fully reactive that cannot adapt to the dynamic environment of vehicular networks and thus, much attention has been paid to HO management optimisation. In recent years, ML-based solutions have been widely explored in various wireless communication research fields, including resource management, mobility prediction, and HO management for different system setups [52]. ML techniques can utilise the rich dataset generated by wireless systems and extract hidden patterns in the dataset that are usually difficult to derive using analytical optimisation techniques [209]. Consequently, this work proposes a DRL-based HO algorithm for V2N communications in cellular vehicular networks to make informed and adaptive HO decisions. By gaining knowledge of the environment using DRL, the proposed algorithm decides when a HO should be executed in an informed manner such that the HO can happen near the optimal point t_0. Also, because the HO points are learned through experiences, the proposed algorithm does not rely on HO parameters (e.g., Δ and TTT) to ensure an optimal target BS is selected.

4.1.2 Related Work

To improve HO performance, various research has been conducted using ML techniques [17, 166] and ML-based solutions for HO optimisation can be classified into three main types: ML-based HO parameter optimisation, predictive HO enabled by ML, and direct HO decision making using ML.

For the scenario in Figure 4.1, it is possible for A3 RSRP HO algorithm to reduce δ_t and hence the grey area representing suboptimal signal quality. This can be achieved by dynamically adjusting the values of Δ and TTT (i.e., reducing Δ or TTT in Figure 4.1 scenario), the HO parameters of the algorithm. ML-based HO parameter optimisation still uses parameter-based HO algorithms for HO decision-making, with a learning algorithm implemented to optimise the value of HO parameters. For example, the Q-learning-based algorithm proposed by Assem *et al.* [125] that optimises the values of TTT and hysteresis for an A3-based HO algorithm by setting the RL reward function to consider the number of HOs, HO delays and throughput system-wise. Another Q-learning-based algorithm was proposed by Goyal and Kaushal [126] to optimise Δ and TTT. This scheme utilised signal strength indicators (RSRP, RSRQ, and SINR), the location and moving direction of a UE, and BS traffic loads to support the Q-learning algorithm to decision optimal values for the two HO parameters.

For predictive HO enabled by ML, ML algorithms are designed to accurately predict future measurement metrics or the future location of a vehicle to trigger HOs. When applied to the scenario in Figure 4.1, this strategy essentially aims to trigger the HO execution in advance to reduce the signal quality downgrade due to Δ_t, i.e. the grey area in Figure 4.1 representing it. An example of this approach in the simplified scenario is that at the 34-second time step in this Figure, the UE sends an accurate prediction of RSRP for the 37-second time step for the BS to start the HO using the A3 RSRP algorithm. In this direction, an FL training setup for future Signal-to-Noise Ratio (SNR) prediction utilising both the macro BS and local UEs was proposed [132]. The predicted SNR values were sent to a conventional HO algorithm to proactively trigger HOs in a mmWave vehicular network. In comparison, the algorithm proposed in [127] could proactively trigger HOs by matching the vehicle's predicted future location with known BS locations to reduce the complexity of HO decision-making as being in close proximity to a BS means better signal quality.

In contrast to the strategy used in ML-based predictive HO, ML-based HO algorithms aim to directly produce optimised HO decisions using an ML algorithm. When applied to the scenario in Figure 4.1, this type of approach tries to identify t_0 such that δ_t can be minimised and hence minimise the signal quality degradation. Various solutions have been proposed in this category, such as a unified HO algorithm for LTE-A [169], based on discrete stochastic dynamic programming [167]. The algorithm considered both UE measurements (RSRP and RSRQ) and overall resource utilisation of BSs to produce HO decisions that also balanced the traffic loads among BSs. A joint HO and power allocation scheme was developed for heterogeneous networks utilising multi-agent DRL [173]. Using a reward design based on system throughput and introducing a penalty for HO, the algorithm optimises BS and power level selections for each UE. Other algorithms also considered the new technologies introduced in 5G cellular networks. For example, a Q-learning-based HO algorithm was proposed in [53] for a mmWave 5G network, where the RSRP is used to trigger the algorithm for HO making. In a similar manner, Mollel et al. proposed a deep Q-learning-based HO algorithm for a mmWave vehicular network [171]. Using SINR as the UE measurement metric, this work adopted event A2 (the serving BS becomes worse than an absolute threshold) to trigger HOs as it could indicate a blockage in mmWave networks while also accelerating ANN training by skipping states that were not points of interest.

Some HO research combined ML HO optimisation techniques to form multi-tier learning-based HO solutions. For example, an LSTM RNN was trained in [174] to predict future received signal strength to trigger HO predictively. After triggering, an HMM [128] was used to optimise the HO decision-making.

4.1.3 Objectives and Contributions

Extensive research has been conducted into ML applications in HO management from different optimisation aspects, with some researchers focusing on specific network setups such as Yajnanarayana *et al.* and Mollel *et al.* who focused on the features of mmWave networks. Most literature considered a scenario of a generic cellular network with slower moving UEs whilst vehicular UE scenarios with much higher moving speed and strict QoS requirements were less common. However, relatively little research evaluating the performance of an ML-based solution with conventional methods using the same input parameters has been considered. Although it is essential to evaluate and compare the performance of different ML-based solutions using standardised datasets and/or test environments [36], little research has considered these aspects of different HO algorithms with only [169] and [174] implementing their proposed algorithms on a full-stack simulator such as ns-2 and ns-3.

Consequently, this work focuses on a DRL-based HO algorithm to tackle the HO optimisation problem in cellular V2N networks based on current cellular network architecture, aiming to explore how DRL may improve the system's HO decision-making performance and evaluate the performance using the well-established network simulator ns-3. The ns-3 simulator is a standardised platform with full-stack configuration for realistic simulation configuration and real-world system configuration emulation. A bespoke simulation is developed with a realistic scenario setup for the Glasgow city centre area, UK, using BS locations and vehicle trajectories that emulate reality. The official ns-3 cellular network module was used in accordance with the 3GPP standards [210]. The proposed algorithm only used the RSRP parameter available in the current measurement reports for HO as the input parameter, and the dataset used for training was gathered directly from the corresponding network layer of the cellular protocol stack via realistic simulations. After offline training, the algorithm was then deployed in the mobility management entity for online performance analysis directly using the ns-3 simulator. Performance comparisons showed a 25.73% packet loss reduction and 3.03 dB SINR gain per HO period of interest (defined as the period starting from t_0 until the HO is executed for both benchmarks, specified by the grey area in Figure 4.1). Moreover, a qualitative comparison of the proposed scheme and the state-of-the-art research utilising ML techniques is provided. The contributions of this work can be summarised as follows:

- A HO algorithm is developed using DRL utilising the standard input parameters list as available in cellular network configurations, hence can be deployed via a software upgrade with small system-level modifications.

- The proposed algorithm is validated on the discrete-event network simulator ns-3 with realistic scenario setups, in contrast to high-level proof-of-concept simulations.

- Performance evaluation against the A3 RSRP HO baseline demonstrates a significant 6.3% improvement in throughput from the proposed algorithm for the considered simulation scenario, while also reaching near-optimal HO point (around t_0) for all HO cases in the simulation, offering improved performance for 5G vehicular networks.

- Through qualitative comparison among state-of-the-art solutions, the proposed algorithm can be positioned to be lightweight with less impact on the system deployment. Combined with the simulation results, this makes the proposed solution a promising candidate for 5G and beyond V2N.

4.2 Proposed Solution

A Double Deep Q-network (DDQN)-based HO algorithm is developed for the above-formulated problem and this section focuses on the design detail of the key elements, including the state space and state observation, action space, and reward design. A step-by-step summary is also included for both training and execution of the proposed solution. The Experience Replay is also deployed along with the proposed DDQN HO algorithm to improve the data efficiency during training while also aiming to improve the overall performance.

Because of DDQN's advantage over the original Deep Q-network (DQN), this research aims to develop a DDQN-based HO algorithm and deploy it in cellular network architecture and compare it with the results of the presently implemented A3 RSRP HO algorithm. A centralised agent is designed that utilises the DDQN HO algorithm, following the same HO decision-making setup as the existing cellular network. The HO process derived from the current Xn-based HO [161] can be found in Figure 4.2.

The state observation and reward design must be based on the cellular measurement report entities from the UE, specifically, the mapped RSRP index values (integer values between 0 and 97 proportional to the raw RSRP measurement in dB, 0 represents the weakest connection quality or no signal, while 97 indicates very strong signal strength) for the serving BS and neighbour BSs and corresponding BS IDs [153, 154]. According to 3GPP, raw RSRP measurement values will first go through layer-3 (the Radio Resource Control layer of the cellular protocol stack) filtering before being reported to the serving BS by a UE. The layer-3 filtering is shown in (4.4), where F_n and F_{n-1} are the current and old filtered RSRP values to report, M_n is the latest received measurement result from the physical layer, and $a = \frac{1}{2^{\frac{k}{4}}}$ where k is the filter coefficient for the corresponding measurement quantity received by the quantity configuration parameter.

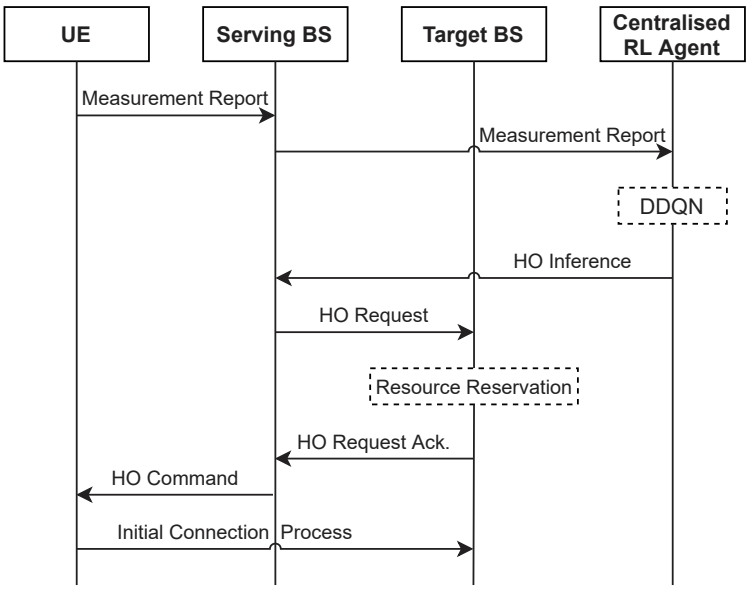


Figure 4.2: The HO process with a centralised DDQN agent. Derived from steps 1 to 7 of Figure 3.2

.

$$F_n = (1 - a) \cdot F_{n-1} + a \cdot M_n \tag{4.4}$$

4.2.1 State Space

Mobility-based BS selection strategies have been intensively studied for vehicular networks, utilising a UE's location and speed [17]. Measuring the exact location of a VUE is expensive and time-consuming, whereas RSRP information can be utilised to estimate a UE's location [211, 212]. This provides a strong mapping between a geographical location in a defined area to a set of RSRP values from the BSs within the area, while more BSs available in the area may further improve the location estimation's precision. Therefore, this study considers the combination of the RSRP values measured by a UE from all surrounding BSs to represent the HO location-of-interest instead of the precise location of the UE (i.e., geo-coordinates of UE's location) and vehicle UEs are assumed to be of the same height to reinforce such representation.

Dependent upon the cellular network's configuration, converted RSRP indexes will be reported by a UE to its serving BS for HO inference [160]. However, in contrast to the configuration deployed in the current network, this study requires that all RSRP indexes of listed BSs within an area to be reported to form a state observation vector.

For a given local area containing n BSs and for a UE at position p, the RSRP measurements of all BSs $\overline{RSRP}_p$ is given as:

$$\overline{RSRP}_p = \{rsrp_p^1, rsrp_p^2, \ldots, rsrp_p^n\} \tag{4.5}$$

Note that the above design is based on the assumption that by deploying the proposed DDQN HO algorithm, the network operator has predefined areas with a known set of BSs for a particular DDQN model to operate, while the network controller is able to switch DDQN models corresponding to the adjacent areas at area boundaries.

The state observation vector s_p is the combination of $\overline{RSRP}_p$ and the serving BS ID $\{\overline{RSRP}_p; BS_{serving}\}$.

However, instead of using the converted decimal value to represent the serving BS ID (as it may be confused with a RSRP index value), this information is designed to be represented via one-hot encoding [213]. For example, if the serving BS of a UE has a local ID of 2, with a total of 5 BSs in the local area, then the serving BS ID after one-hot encoding becomes the vector $\overline{BS}_{serving} = \{0, 1, 0, 0, 0\}$. Therefore, s_p can then be formally defined as:

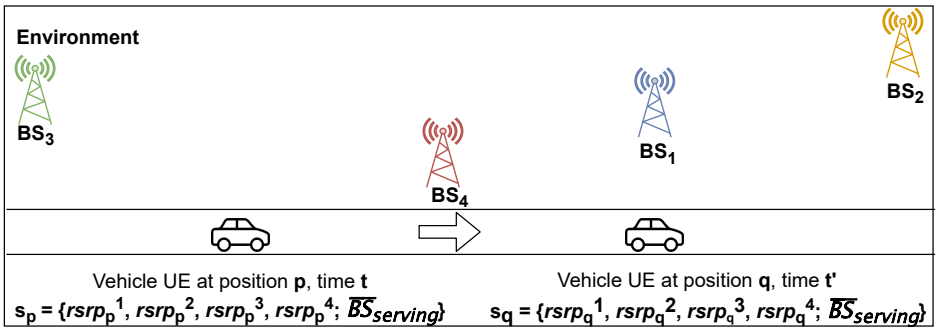
$$s_p = \{rsrp_p^1, rsrp_p^2, \ldots, rsrp_p^n; \overline{BS}_{serving}\} \tag{4.6}$$

and the complete state space S is then defined as the collection of all possible states. To observe the environment fully, the algorithm assumes that a state observation is periodically reported by a UE for both training and performance evaluation.

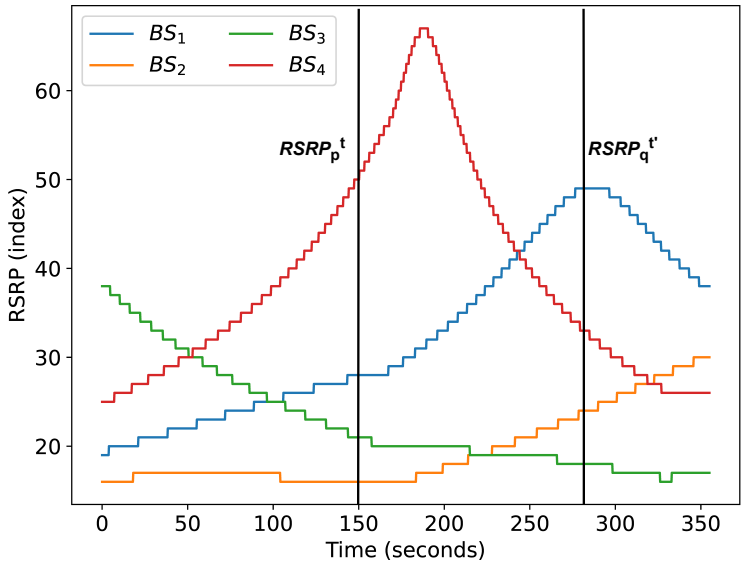
Figure 4.3 demonstrates the environment-state relationship following the state design. At the top of the figure, a conceptual geographic environment is represented by n BSs and a single moving vehicle UE between two locations, p and q, at time t and t', to show the formulation of s_p. The lower part of the diagram shows the $\overline{RSRP}_p$ formulation in a graphical example of an RSRP record assuming $n = 4$.

4.2.2 Action Space

An action a to be taken in each state can be defined as the BS to connect to for the next state of a UE, i.e. all listed BSs in the local area including the serving BS (if the decision is not to HO). Therefore, the action space can be defined as a vector consisting of local BS IDs, $A = \{BS_0, BS_1 \ldots, BS_n\}$. Note: a HO will only be executed if the action specifies a neighbouring BS to connect to, while an action for the serving BS ID indicates no HO required for the UE and to remain on the current the serving BS.



(a)



(b)

Figure 4.3: A graphical demonstration of the state space design. (a) A simplified environment consisting of $n = 4$ BSs and a VUE; (b) An illustrative scenario with $n = 4$ BSs showing the RSRP record of a UE trajectory with the x-axis being the elapsed simulation time.

4.2.3 Reward Design

In line with the design of the state observation space, the reward design should also utilise only information from the measurement report as the design aims to not influence the current HO algorithm deployment wherever possible. A straightforward design uses the RSRP of the new serving BS after the HO (hence state transition) specified by the action taken. The proposed reward design is to normalise this value with the highest reported value to emphasise the RSRP difference between the current BS choice and the local maximum RSRP, following the premise that higher signal strength correlates with a higher SINR and thus larger throughput.

A constant HO penalty is introduced in the reward design to enable the agent to consider the negative impact of performing HOs. The simplified approach of maximising cumulative RSRP-based reward may lead to unwanted HOs resulting in the ping-pong effect due to noisy measurement reports causing delays in data transmission as occur in

the current hard HO implementation of LTE and 5G cellular systems.

As shown in (4.7), $r(s_p, a_p; s_{p+1})$ is the reward gained after taking action a_p in state s_p and observing the next state s_{p+1}. $max(\overline{RSRP}_{p+1})$ is the largest RSRP value from the measurement report in state s_{p+1}. In addition, $rsrp^a_{p+1}$ is the RSRP value of the target cell decided by the action taken, and C_{HO} is the introduced punishment on HO, a positive number with its specific value configuration depending on the environment.

$$r(s_p, a_p; s_{p+1}) = \begin{cases} max(\overline{RSRP}_{p+1}) - rsrp^a_{p+1} - C_{HO}, \text{if HO is triggered} \\ \\ max(\overline{RSRP}_{p+1}) - rsrp^a_{p+1}, \text{otherwise} \end{cases} \tag{4.7}$$

4.2.4 Algorithm Design

The proposed DDQN algorithm implementation consists of two phases: exploration (training) and exploitation (execution). During the exploration phase, the algorithm is trained offline such that the dataset is collected, pre-processed, and used to train the DDQN without directly interacting with the environment. The dataset is collected along different UE trajectories and the ϵ-greedy strategy [214] is used to explore various actions in every state to update the ANN that approximates the optimal HO value function. Algorithm 4.1 summarises the DDQN algorithm's training process during the exploration phase.

The training process completes in the exploitation phase, where the ANN update is terminated, and the trained ANN is used to emulate the optimal HO policy to take HO actions by setting $\epsilon = 0$ in the ϵ-greedy strategy. In order to directly compare with the A3 RSRP baseline, event A3 is also used in the exploitation phase to trigger the HO inference. Figure 4.4 demonstrates the algorithm's workflow in the exploitation phase. Note that the new data generated during the exploitation phase can also be stored and processed to update the DDQN algorithm to learn the underlying patterns in the new datasets. However, the design and implementation of such future updates to the algorithm are beyond the scope of this work.

Algorithm 4.1: Training of the proposed DDQN HO algorithm

Initialise: θ_q - Training Q network; θ'_q - Target Q network;
$\mathbb{B}$ - The replay Buffer; N_{rp} - Replay buffer capacity;
N_{mb} - Mini-batch size; N_f - Target network θ'_q update frequency;
γ - The discount factor; ϵ - Probability to take a random action;

for *episode* $\leftarrow 1, 2, ..., N_{episode}$ **do**
  Set the initial state s_1 ;
  for $i \leftarrow 1, 2, ..., end\ of\ trajectory$ **do**
    Observe s_i ;
    $$a_t \leftarrow \begin{cases} \text{a random action,} & \text{with } \epsilon \\ argmax_a\ Q(s_t, a; \theta_q), & \text{with } 1-\epsilon \end{cases}$$
    Execute a_i and **observe** s'_i and r_i ;
    Store $\{s_i, a_i, s'_i, r_i\}$ in $\mathbb{B}$;
    if N_{rp} *is reached for* $\mathbb{B}$ **then**
      Delete the oldest sample in $\mathbb{B}$;
    end
    if $\mathbb{B}$ *has at least* N_{mb} *samples* **then**
      Sample a mini-batch of (s, a, s', r) from $\mathbb{B}$;
      foreach *(s, a, s', r) sample* **do**
        Construct target value:
        $$y \leftarrow \begin{cases} r, & \text{if } s' \text{ is the terminal state} \\ r + \gamma Q(s', argmax_{a'} Q(s', a'; \theta_q); \theta'_q), & \text{else} \end{cases}$$
        Calculate loss $\|y - Q(s, a; \theta_q)\|^2$;
      end
      Do gradient descent with the calculated loss array ;
      Update parameters of θ_q
    end
    if $mod(i, N_f) = 0$ **then**
      /* Copy θ_q parameters to θ'_q */
      $\theta'_q \leftarrow \theta_q$
    end
  end
end

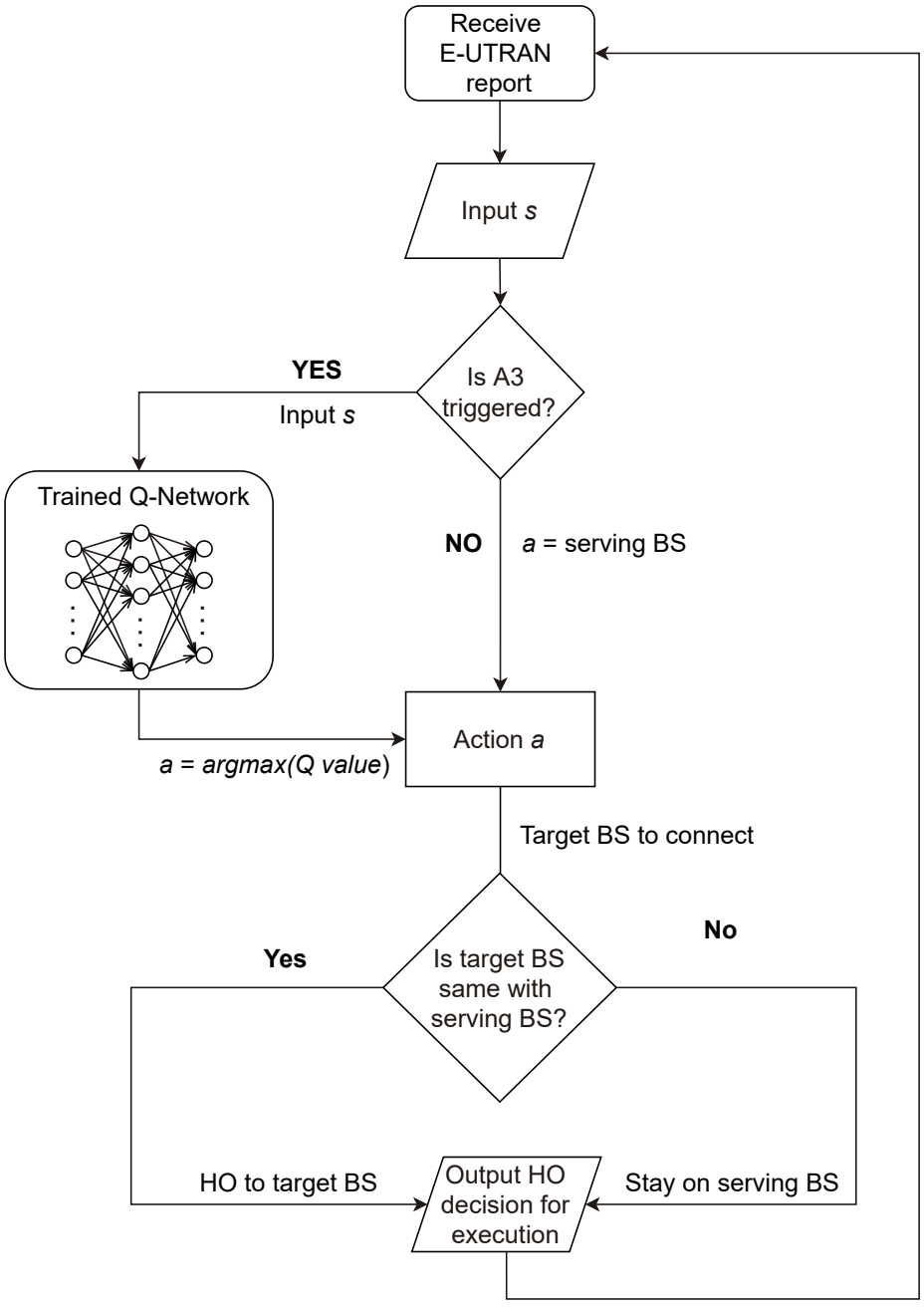


Figure 4.4: A flowchart of the proposed DDQN HO algorithm in the execution phase.

4.3 Performance Evaluation

This section covers the simulation-based performance evaluation and analysis for the proposed DDQN HO algorithm against the A3 RSRP algorithm currently deployed in the network. First, the detailed configuration is introduced which includes the scenario and network setup, as well as the ANN hyperparameter setups, followed by the performance metric introduction. This is because the choice of performance metrics has also been adapted to the selected tools. Finally, the performance analysis contains a case study to demonstrate the evaluation process, and statistical results to show the overall performance. Note that the proposal focuses on using the same input parameters available in the cellular system with the proposal able to be deployed in the current architecture, benchmarking with other existingML solutions is thus beyond the scope of this chapter due to pragmatic reasons and is regarded a future work.

4.3.1 Scenario and ns-3 Simulation Configurations

Due to the complexity of collecting real-world UE handover data, the dataset for this research was generated using the ns-3 network simulator [215]. The ns-3 simulator is an open-source, discrete-event full-stack simulator that allows tracing internal events with flexible configurations and supports multiple communication technologies. The ns-3 official standard-compliant LTE module LENA [216] was chosen to configure an LTE cellular network scenario to investigate a cellular V2N communication network. This approach was adopted as the 5G and LTE network HO mechanisms are very similar and the 5G-LENA [217] (the 5G version of the LENA module) is still under development at present. The main rationale for using ns-3 is that with the LTE module, it models every layer (e.g. physical) of the cellular protocol stack and is able to provide realistic data, while it also enables the evaluation of the proposed algorithm emulating the deployment in the cellular architecture.

The need for realistic simulation data is to train and evaluate the DDQN HO algorithm in an environment that is a close analogue to a real-world network. The simulation scenario was a 2×2 km local area in the city centre of Glasgow, UK, which is a typical European city. For the implementation of a realistic mobility simulation for vehicles, the routes mobility model [218] was selected, which utilised the Google Maps' directions Application Programming Interface (API) [219] and the way-point mobility model provided by ns-3. By specifying the target area and a driving trajectory's start and end points, the trajectory waypoints could be generated by the Google Map's API with the travelling time calculated according to the road traffic prediction of the area when generating the trajectory. The default option "best guess" provided in the API [220] was utilised for trajectory generation and all trajectories were generated within a short period on the same day to keep the traffic model's output consistent. The underlying travel speed of a trajectory is hence dependent on the travel time and

bounded by the speed limit of the area. The routes mobility model could then transform the generated trajectory into a useful format by the ns-3 simulation. For BSs, the location references were taken via the Cell Mapper website [221], which recorded the real-world BS deployment locations based on measured data from participants. For the scenario setup, eight BSs from the UK mobile operator Vodafone was chosen.

Figure 4.5 shows the scenario setups including the environment setup, BS locations, and the network architecture overlaid on a map of the selected area. Each BS is connected to the core network (i.e., the Mobility Management Entity and Serving Gateway for LTE) via the S1 interface and to other BSs via the X2 interface. A sample vehicle trajectory is also included in Figure 4.5, showing the vehicle's current serving BS is the red BS and will be handed over to the green BS during the passage along its route. The red and green circles are simplified indications of the signal strength from the corresponding BSs is the same, while the yellow dot represents the HO location of interest for HO between the two BSs. The VUE should be handed over from the red BS to the green BS after this location for improved signal quality. The VUE's trajectory is shown in black.

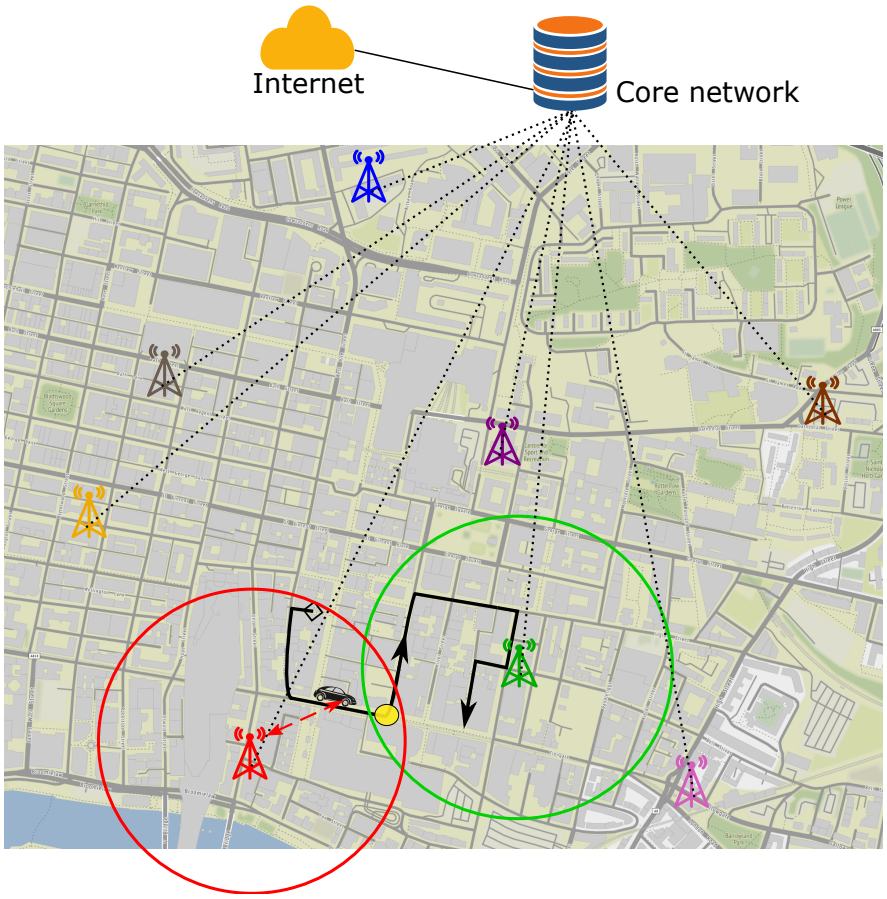


Figure 4.5: Glasgow city centre scenario setup with 8 BSs marked in different colours. The route marked in black shows 1 illustrative trajectory of a road vehicle used for the case study in Section 4.3.3, where a potential HO along the trajectory is also demonstrated (from the red BS to the green BS specified by the yellow dot).

After establishing the scenario, the detailed network configuration is listed in Table 4.1.

An isotropic antenna model is used at the current stage, demonstrating the same HO strategy while simplifying the scenario, and hence the state space to be considered by the DDQN HO algorithm. Other network configurations, such as the carrier frequency and noise figures of BS and UE, are set according to the 3GPP standards [180, 222].

In addition to the considered 3GPP UMa (urban macro) path loss model, small-scale fading should also be considered for a realistic simulation as it is recognised that multi-path fading and Doppler spread can introduce significant variations during propagation, particularly in vehicular networks of high UE mobility. Therefore, trace-based fading was generated via the supplemental Matlab script provided in the LENA module [223], which used the fading model implementation derived from the Jakes' model [224] for Rayleigh fading generation used in [225], and loaded into ns-3. This approach has been chosen by the ns-3 developers to limit the computation complexity of the simulation [223]. The "Vehicular" mode with nodes' moving speed of 60 km/h was chosen to generate the Rayleigh fading traces and hence introduce small-scale fading to the simulation scenario. As the speed limit of the considered area is 30 mph (or 48.28 km/h) with a large 20 mph (or 32.19 km/h) zone [226], the chosen 60 km/h speed for fading generation thus leads to the worst possible case for fading (maximum small-scale fading) for the area. Note that 60 km/h speed is not the actual vehicle moving speed, which was already defined by the Google Map's direction API when generating the trajectories. For the configured simulation, a higher vehicle moving speed will increase the fading effect by introducing the higher Doppler frequency shift besides the more frequent change in signal strength (e.g. RSRP) caused by the more rapid position change of a UE for a known trajectory. The opposite hold true for a lower vehicle moving speed.

To train the performance of the proposed DDQN HO algorithm, 18 trajectories covering the major routes across the selected area of Glasgow city centre were generated via the routes mobility model [218] for the ns-3 simulation, to form a training dataset for the exploration phase of the DDQN algorithm. The maximum simulation time is set to 800 seconds (13.3 minutes) so that all vehicles have sufficient time to complete their trajectories. The User Datagram Protocol (UDP) packet size is set to 4096 bits to guarantee that the system can transmit all packets when not in a HO period-of-interest to help analyse the impact of HO on packet loss, which is a selected performance metric.

For performance evaluation, overall the same trajectories were used for the exploitation phase of the algorithm, while slight modifications were applied to the detailed way-points representing the trajectories to test the algorithm's generalisation ability. Performance evaluation was implemented online using the ns-3 simulator directly and the connection between a UE-BS pair was terminated shortly after the UE reached its endpoint to avoid redundant data collection.

Table 4.1: Simulation configuration

Setup properties	**Parameters and values**
Scenario	A 2×2 km^2 street blocks of Glasgow city centre
Cellular network	8 BS sites, UK operator Vodafone [221]
Vehicle mobility model	Waypoint mobility model generated by Routes mobility model [218], speed controlled by Google Maps' Direction API.
Number of vehicle trajectories	8
Antenna model	Isotropic antenna model
Antenna height	**UE**: 1.5 m; **BS**: 25 m
Pathloss model	3GPP UMa (urban macro)
Fading model	Rayleigh fading
Scheduling algorithm	Proportional Fair
BS transmission power	40 dBm
Carrier frequency	2115 MHz, downlink only
Noise figure	**BS**: 5 dB; **UE**: 9 dB
HO algorithm	**A3 RSRP (baseline)**: hysteresis: 3 dB, TTT: 320 ms; **DDQN (proposed)**: HO penalty: 3 (equivalent to dB)
Application setup	UDP, downlink only **packet interval**: 20 ms (50 packets / sec) **individual packet size**: 1024 bits
Maximum simulation time	800 seconds

For the DDQN setup, a fully connected feed-forward ANN with 3 hidden layers was used, and the detailed hyperparameter configurations are summarised in Table 4.2. The Gaussian Error Linear Units (GELU) were used as the activation function of hidden layers [227], while the optimisation algorithm was set to the Adam with Decoupled Weight Decay (AdamW) optimiser [228] for their better performances in general in ANN training compared with their predecessors (other linear unit activation functions and the original Adam optimiser). The initial learning rate was set to 0.001, and an exponential learning rate decay was configured with the decay ratio set to 0.98 to stabilise the ANN training convergence. The replay buffer was set to have a maximum capacity of 100,000 observation samples, and a mini-batch sampling size of 1024 was configured for ANN training. After completing the exploration phase, the trained ANN was then turned into the exploitation phase and deployed to directly interact with the ns-3 simulator using the ns3-ai module [229] for direct online performance evaluation.

4.3.2 Data Collection and Evaluation Metrics

To train and evaluate the DDQN HO algorithm, RSRP data of all BSs need to be collected, following the design in Section 4.2. The data is collected directly from the ns-3 LENA module's Radio Resource Control (RRC) layer of the VUE where Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) measurements are performed and reported [155]. An RSRP index record of all 8 BSs is generated for each trajectory

Table 4.2: ANN configuration

Hyperparameters	**Values**
Hidden layers, Neuron size	3, $256 \times 128 \times 64$
Hidden layer activation function	GELU [227]
Output layer activation function	Linear
Target network update frequency	Every 100 episodes
Optimiser	AdamW [228]
Learning rate	0.001
Learning rate decay ratio	0.98
Replay buffer size	100,000
Mini batch size	1,024

to form a dataset to train the algorithm.

To evaluate the performance of the proposed HO algorithm, several metrics are chosen to compare performances between the proposed algorithm and the A3 baseline:

- **The SINR gain** $\mathbb{G}_{SINR}$: According to (4.3), in order to maximise throughput, it is essential to maximise the $SINR$ by switching to a better-performing neighbour BS where applicable while minimising the N_{HO} given a constant t_d in (4.1). Therefore, the SINR traces of serving BSs are collected for signal quality (hence indirect throughput comparison) comparison between the proposed DDQN HO algorithm and the A3 RSRP baseline during the HO period of interest, while the number of HOs N_{HO} is also recorded for each trajectory. The SINR traces are recorded by the ns-3 LENA module and presented as raw linear-scale values [230], and then converted to decibels for performance comparisons. This metric is defined as The normalised SINR value using the DDQN HO algorithm with respect to the A3 RSRP baseline during a HO period of interest, calculated as $\mathbb{G}_{SINR} = \sum_{HOPeriod} SINR_{DDQN} - SINR_{A3}$. The sum operation is used because a discrete-time simulator is used hence discrete data. The baseline result for this metric is 0 since $\sum_{HOPeriod} SINR_{A3} - SINR_{A3} = 0$.

- **The Packet Data Convergence Protocol (PDCP) packet loss**: The PDCP layer in the LTE protocol stack is responsible for the transfer of data on the control/user planes [231] hence is the best metric to directly evaluate throughput performance. A smaller packet loss during a HO period of interest suggests higher throughput during that period. The PDCP packet loss is calculated by the ns-3 simulator with the statistics recorded by the LENA module [216] while the calculation of packet loss follows ns-3's data plane error model utilising the link-to-system technique and block error rate mapping [232].

- **The throughput gain** $\mathbb{G}_{packet}$: Corresponding to the PDCP packet loss, this metric indicates the improvement in throughput during a HO period of interest

using the proposed algorithm with respect to the A3 baseline. For a given HO, this metric is calculated as $\mathbb{G}_{packet} = \frac{packet_{DDQN} - packet_{A3}}{packet_{transmit}}$, where $packet_{DDQN}$ and $packet_{A3}$ stand for the received PDCP packets using the proposed DDQN algorithm and the A3 baseline, respectively, while $packet_{transmit}$ indicates the transmitted PDCP packets in total during this period.

- **The gain in time delayed for HO** $\mathbb{G}_{\delta_t}$: The time delay δ_t as expressed in Figure 4.1 indicates when a HO decision is made with respect to the optimal time instant. Therefore, $\mathbb{G}_{\delta_t}$ becomes a clear metric to measure how much faster the DDQN HO algorithm makes the HO decision towards the optimal HO with respect to the A3 RSRP algorithm. For each HO case, this is calculated as $\mathbb{G}_{\delta_t} = t_{DDQN} - t_{A3}$, where t_{DDQN} is the HO decision time step for the proposed algorithm, while t_{A3} represent that of the A3 baseline and the baseline value for this metric is 0 ($t_{A3} - t_{A3}$). Note that this metric becomes a logical indicator in this work as the proposed algorithm and A3 RSRP algorithm are the only benchmarks, with $\mathbb{G}_{\delta_t} > 0$ suggesting improvement in HO performance while a negative $\mathbb{G}$ represents inferior performance compared with the A3 baseline.

The HO period of interest used in the above metric is defined as "for each HO case, starting from t_0 until both considered HO algorithms (the proposed solution and the A3 baseline) decide to execute this HO", which is equivalent to the grey area presented in Figure 4.1. To calculate these metrics, the HO period of interest is first defined for each HO identified in the simulation in a case-by-case manner.

The presentation and analysis of $\mathbb{G}_{\delta_t}$ and $\mathbb{G}_{SINR}$ for all 18 trajectories are not easily presented in a graphical manner in the simulation scenario due to a number of HOs occurring. Therefore, for the sake of simplicity and clarity, the results from 1 exemplar trajectory selected from the full simulation are first presented in Section 4.3.3 to give a detailed graphical demonstration of metric analysis for a single HO instance. Then, the statistical results for the whole simulation scenario (i.e. all 18 trajectories including the exemplar one) are then presented in Section 4.3.4 with some edge-case discussions. With the exception of the reference results, all presented results are collected via the same round of training and evaluation (hence the same trained DDQN model). Note that the HO period of interest and the values of the performance metrics are dependent on the moving speed of VUEs because, under the same trajectory and application configuration, a larger moving speed will result in shorter travel time hence shorter HO periods of interest for A3 baseline (due to RSRPs changing faster thus earlier hysteresis satisfaction) and fewer transmitted packets in total. The opposite applies when considering a slower moving speed.

4.3.3 Result Analysis for One HO Case

Following the scenario and simulation setup, this subsection presents an exemplar performance analysis of a single trajectory within the 18 trajectories in the simulation scenario. The visualisation for this trajectory's geographical information is shown in Figure 4.5 (the illustrated trajectory), and the performance of the proposed DDQN HO algorithm was evaluated against the A3 RSRP baseline with all other network settings kept constant. In contrast to the learning-based algorithms, the A3 RSRP HO algorithm triggers a HO based on event A3, when a neighbour BS's RSRP becomes greater than that of the serving BS by a predefined offset value, as shown in Figure 4.1. The two parameters of this HO algorithm, hysteresis and TTT, are used to avoid the ping-pong effect. To permit easy comparison in this performance evaluation, the A3 RSRP baseline used the same parameter configuration as current cellular networks as stated in Table 4.1.

The RSRP record for all BSs throughout this trajectory is shown in the top half of Figure 4.6. For improved visual clarity, this part of Figure 4.6 is presented without fading. Accordingly, 1 HO should be triggered for this trajectory, which also corresponds with the simulation results. This HO period-of-interests is highlighted by a black rectangle in the top half of Figure 4.6, and the zoomed-in of this area is presented in the bottom half of the figure. The bottom half of Figure 4.6 is presented with fading enabled to reflect the actual simulation.

The optimal triggering instant of this HO is around 116 seconds after the simulation starts (0 seconds) based on Figure 4.6. This is when the RSRP of BS_3 becomes greater than that of BS_4, the initial serving BS, and remains so until the end of the simulation. Due to the 3 dB hysteresis and TTT, the A3 RSRP baseline must wait until both thresholds are satisfied to trigger the HO at 126.72 seconds (with a time step interval of 0.02 second), while the DDQN-based HO algorithm triggers the HO as soon as the optimal HO point is reached (116.68 seconds). As a result, the DDQN-based HO reduces the delayed time of HO by having a $\mathbb{G}_{\delta_t} = 10.04$ seconds compared to the A3 RSRP baseline for this exemplar case for the considered scenario. As $\mathbb{G}_{\delta_t} > 0$, this means that a positive performance gain is achieved by the proposed algorithm compared to the A3 RSRP baseline while combining the information in Figure 4.6, it can be judged that the proposed algorithm reaches a near-optimal HO decision-making.

After presenting $\mathbb{G}_{\delta_t}$, Figure 4.7 includes the normalised serving BS SINR ($\mathbb{G}_{SINR}$) using the proposed DDQN HO with respect to the A3 baseline to show the SINR gain. When not in the HO period, the normalised SINR is 0 dB as the experience SINR using both HO algorithms are the same, while the differences are clearly evident during the HO period of interest. As the simulation setup includes fading that introduces fluctuations in Figure 4.7a, a reference curve of normalised SINR is also included in Figure 4.7b

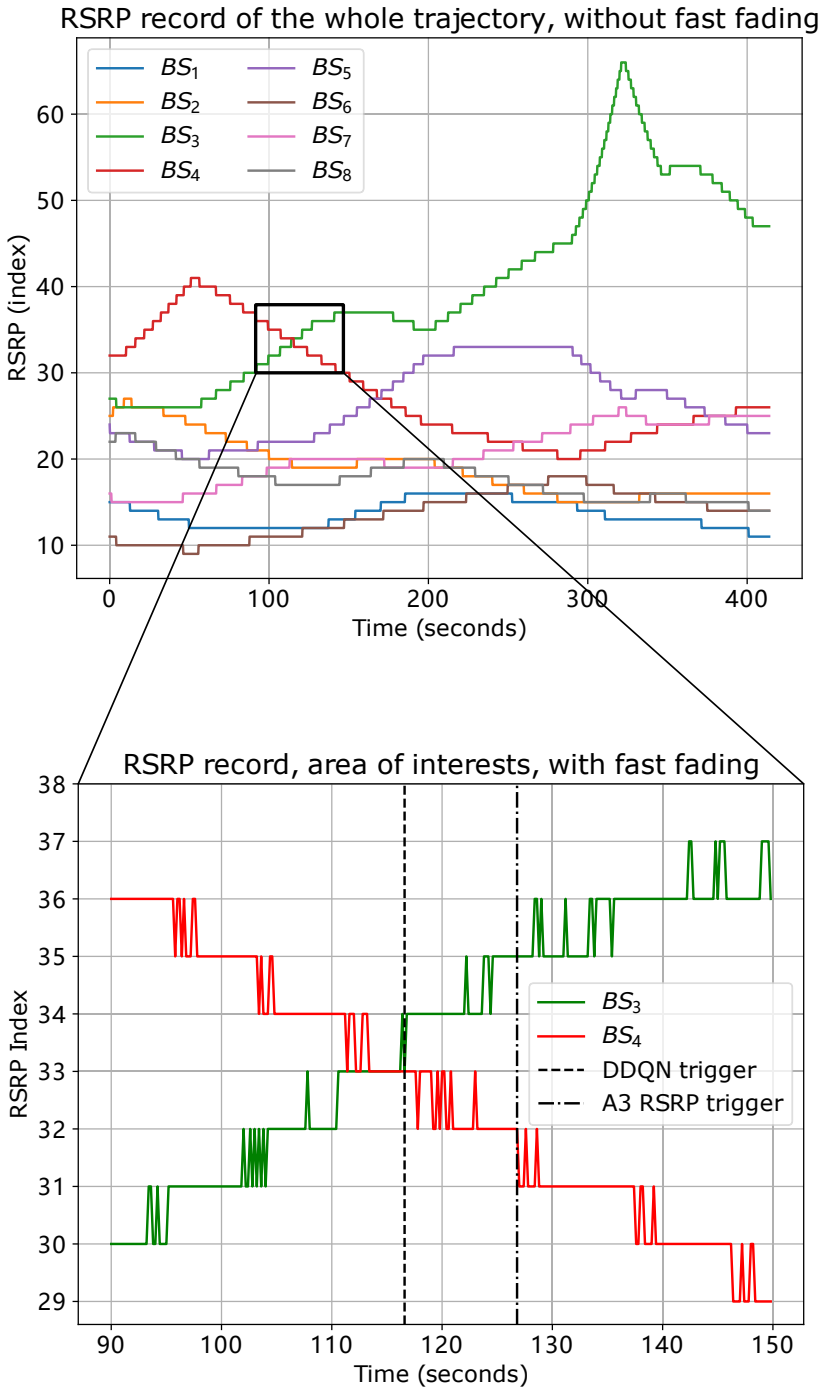


Figure 4.6: Case study: The corresponding RSRP record for the selected trajectory. The top half of the figure includes the RSRP record for all 8 BSs (without fading for a clear visual presentation). The bottom half of the figure is the zoomed-in period of interests when a HO was triggered, with the triggering instants of DDQN and A3 baseline plotted as vertical lines (fading enabled, with configuration introduced above).

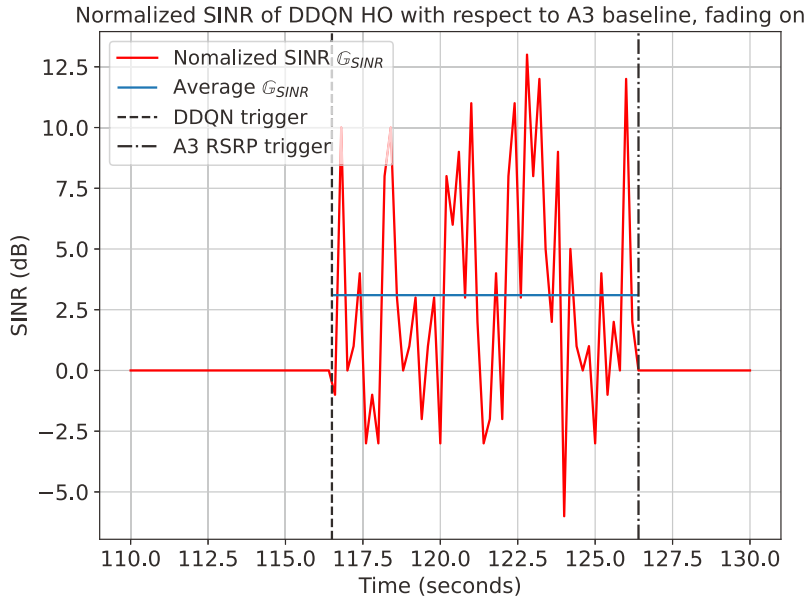
using the same simulation configuration but disabling fading to give a clearer visual presentation of $\mathbb{G}_{SINR}$. Note that the same DDQN trained using the dataset generated in the scenario with fading is deployed to produce results in this reference, and in both sub-figures, the average $\mathbb{G}_{SINR}$ is calculated and plotted as well. Also, by disabling fading, the RSRP record becomes smooth, leading to some state changes (as the DDQN algorithm is trained using data that included fading) and hence different HO triggering instances when using the two HO algorithms. However, the DDQN's HO start at 115.8 seconds is still an optimal trigger shown by Figure 4.6. It can be inferred from above that a "bad" HO decision, either a suboptimal one like the A3 triggering or a wrong HO decision (i.e., HO to a worse performing BS) can lead to an overall negative $\mathbb{G}_{SINR}$ and thus service downgrade with a smaller throughput and more dropped packets due to inferior link budget and potential ping-pong effect. However, the proposed solution has resilience against such bad decisions and maintains optimal decision-making by gaining environmental knowledge during the offline training stage, with the reward design guarding the agent against making wrong decisions (since the agent aims to maximise the long-term accumulated reward). This is further backed by the statistical results in the following section.

The results shown in Figure 4.7 demonstrate an obvious HO performance gain with respect to $\mathbb{G}_{SINR}$ during the HO periods of interest, with a maximum SINR gain of over 12.5 dB compared with the A3 RSRP baseline. According to the reference curve without fading, an average $\mathbb{G}_{SINR}$ of 3.51 dB is achieved by the DDQN HO algorithm during the HO period-of-interests, which is between 116 and 127 seconds as indicated by the two vertical dotted lines.

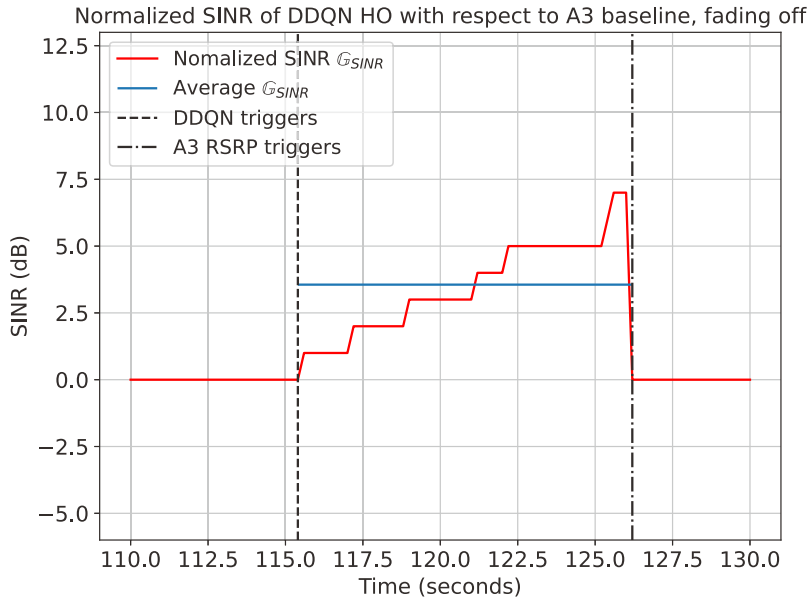
The improvement of HO performance is also reflected in packet loss as shown in Figure 4.8, where the PDCP packet loss is counted for this HO for the A3 RSRP baseline and the DDQN algorithm, respectively. For the HO period-of-interest between 116 and 127 seconds shown in Figure 4.7, overall 550 packets are sent. When using the A3 RSRP baseline, the UE experiences a packet loss of 27 packets due to waiting for HO to trigger and the connection drop during HO. This number for DDQN is 9 packets, indicating a 66.7% better performance in packet loss and an equivalent throughput gain $\mathbb{G}_{packet}$ of 3.3%.

4.3.4 Results Analysis for All Trajectories

In total, there are 46 HOs using the A3 RSRP baseline, and 47 HOs using the proposed DDQN HO algorithms for all 18 trajectories throughout the simulation over the scenario. The edge case for the additional HO that happened is shown by the black rectangle area in Figure 4.9, where the RSRP difference between the best neighbouring BS and the serving BS could not satisfy the A3 RSRP baseline's 3 dB hysteresis. In contrast, the DDQN algorithm learned the RSRP features along the whole trajectory and performed



(a)



(b)

Figure 4.7: The SINR gain $\mathbb{G}_{SINR}$ during the HO period-of-interests of the DDQN HO with respect to the A3 RSRP baseline for the trajectory. Both results were produced by the same DDQN trained using data with fading. (a) The original result with fading enabled (b) Reference result with fading disabled. Note that before 116 and 127 seconds, the serving BS is the same for both HO algorithms and thus the normalised SINR is 0 dB.

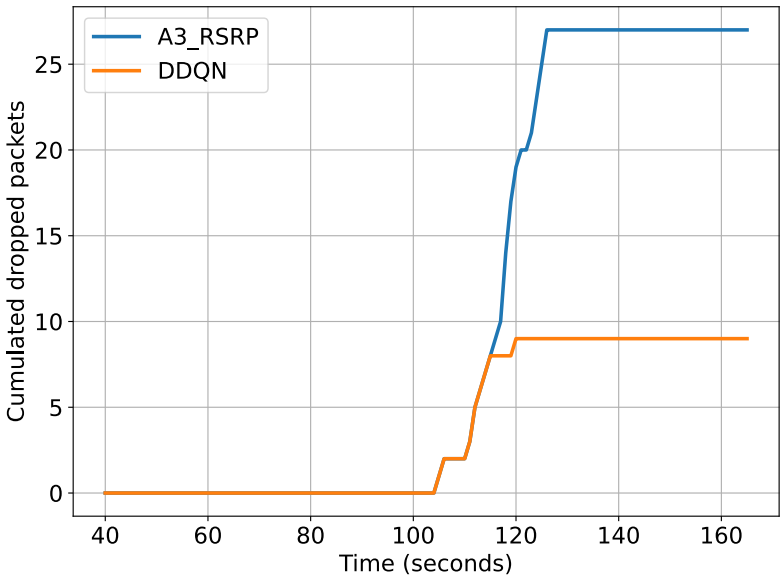


Figure 4.8: Accumulated packet loss comparison between the proposed DDQN HO algorithm and A3 RSRP baseline for the demonstrative trajectory. 550 packets were sent during the considered HO period of interest between 116 and 127 seconds. The recorded time scale is extended in this figure for clearer visuals.

an additional HO to improve optimal signal strength, demonstrating the superiority of having knowledge of the environment. Excepting this one edge case, both HO algorithms have the same number of HOs, indicating that DDQN manages the ping-pong effect as effectively as the A3 RSRP baseline does. However, instead of using predefined parameters (hysteresis and TTT) as in A3 RSRP HO, the proposed DDQN algorithm learns with the HO cost punishment of unnecessary HOs while being able to avoid staying connected to a BS with suboptimal signal strength until the static conditions are satisfied as the A3 baseline.

Throughout the whole evaluation, there are 27400 packets transmitted for all 47 HO periods of interest. Using the DDQN-based HO algorithm results in 1736 fewer accumulated lost packets over the 47 HOs compared to the A3 RSRP baseline, showing an improvement in cumulated packet loss by 42.54%, as shown in Figure 4.10. This corresponds to a throughput gain $\mathbb{G}_{packet}$ of 6.3% for the whole simulation. As the HO mechanism of 5G is very similar to that of the current LTE networks, the evaluation results suggest a potential for significant performance gain in a dense 5G mmWave network that will have many more HOs due to the short range of mmWave beams. The significantly smaller packet losses using the proposed HO algorithm in the considered simulation scenario will result in more packet loss reduction in a dense 5G network.

As demonstrated in section 4.3.3, using the DDQN-based HO algorithm makes the HO triggering instant closer to an optimal HO point compared to the A3 RSRP baseline. After checking the HO triggering point between the two algorithms, positive $\mathbb{G}_{\delta_t}$ is achieved for all HOs that happened in the simulation scenario, suggesting positive gains

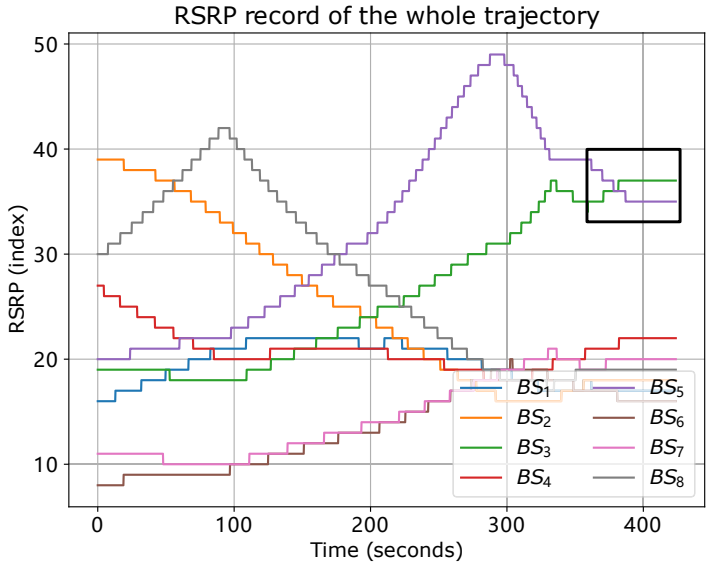


Figure 4.9: The edge case that leads to DDQN's 1 more HO trigger, marked in the black rectangle. Fading is disabled for a clear visual presentation.

achieved for these HO cases. Moreover, the performance trend of the exemplar HO case remains the same for all HO cases after this check, suggesting that the proposed algorithm can reach a near-optimal (t_0) HO decision-making point for the considered scenario. For metric $\mathbb{SINR}$, the proposed DDQN algorithm achieves an average of 3.03 dB gain, with a minimum gain being 1.84 dB while the maximum gain being 3.41 dB across all HO cases. For the considered scenario, this is a significant gain in SINR compared to the baseline of 0 dB according to the definition of this metric. Both metrics suggest significant improvement compared with the A3-based solution deployed in the current cellular network.

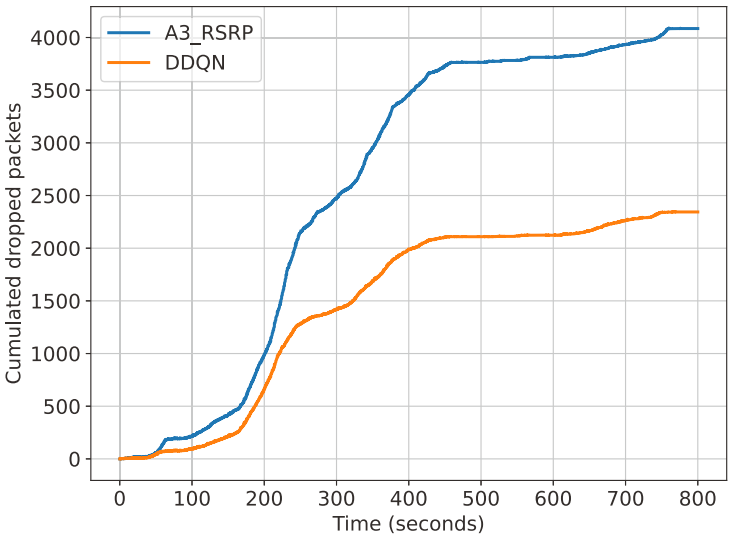


Figure 4.10: Accumulated packet loss comparison for all trajectories between the A3 RSRP baseline and the proposed DDQN HO algorithm. For all the HO periods of interest, 27400 packets have been sent in total for these statistical results.

4.4 Discussion on Deployment Aspects

After presenting the performance evaluation results, some insights are provided in this section regarding the deployment aspects, which also point out potential future research for this work. In addition, a qualitative comparison of the proposed algorithm with some state-of-the-art research is also provided.

4.4.1 Discussion on the Effect of Vehicles' Moving Speed

As HOs are for a moving VUE, for the same trajectory and system setup, different moving speeds of the VUE will influence the state observation of the proposed RL algorithm due to the resultant variation in fading, while also changing the value of the performance metrics.

For a given trajectory, the recorded results of the considered metrics can vary depending on the UE's moving speed. The reason for this is that the slower a UE moves along its passage, the longer it takes to satisfy the A3 RSRP baseline's hysteresis and TTT, resulting in a larger δ_t for the baseline and thus higher $\mathbb{G}_{\delta_t}$ and $\mathbb{G}_{SINR}$. Additionally, it is also anticipated that the pack loss would be larger for the A3 RSRP algorithm with a slower moving VUE as more packets are transmitted during the longer elapsed time while a further delayed HO leads to a prolonged period for suboptimal signal strength using the A3 baseline. The opposite of the above holds true considering a VUE with a larger moving speed. In contrast to the A3 baseline, this aspect does not influence the proposed algorithm since the state observation defined in (4.6) suggests a position in the radio environment that corresponds to a geographical one, which remains constant facing different VUE moving speed. Additionally, as the proposed DDQN is based on the model-free Q-learning technique that does not require a model of the environment and is trained using data generated via known trajectories, the state transition variation caused by a UE's moving at different speeds along a given trajectory will not influence the overall decision making. Therefore, after the initial training, the proposed solution would still make near-optimal HO (the optimal is defined to be at t_0 as in Section 4.1) decisions along the trajectory for different VUE moving speeds.

Nevertheless, the above discussion may not be true when considering moving speeds' effect on fading and thus state observation of the DRL algorithm. Fading in wireless channels can be broadly classified as large-scale and small-scale fading, and Doppler spread from the small-scale fading has a positive correlation to the UEs' moving speed [233]. Consequently, a large moving speed indicates a more significant fading and thus more noisy state observation for the proposed DRL algorithm. The fading setup in Section 4.3.1 and the results in Figures 4.6 and 4.7 indicate that the same DDQN is able to produce the optimal HO decision in both the theoretical worst fading situation and in the ideal situation where only path loss is considered. The fact that the DDQN was

only trained once using the noisy data and the satisfying results in such extreme cases suggest that the proposed DDQN has greater resilience against the speed-dependent Doppler spread within the considered UE speed range. However, it may be necessary to train the algorithm using data generated at the speed limit to provide the DDQN knowledge of the worst fading case, which requires further investigation.

4.4.2 Discussion on the Deployment Costs

As this research looks at the deployment aspect of an intelligent HO algorithm, it is essential to consider the implementation cost alongside the performance gain. For the proposed DDQN algorithm, the fundamental assumption is that the RSRP measurement of all listed BSs in a local area to be reported for HO decision-making. Because the related configuration options are available in the current cellular network setup [153, 155, 158], implementing the algorithm locally requires only a software patch.

Compared to the current HO solution (A3 RSRP), the proposed algorithm requires proportionally larger data input to operate. In the proposed scenario of 8 BSs, the A3 RSRP baseline usually gets a report of two entries, the RSRP and IDs of the serving BS and the best neighbouring BS. In contrast, the current design DDQN requires the reports for all 8 listed BSs to operate, resulting in quad times the data input. Linear complexity yields a moderate cost with respect to computation and storage. Furthermore, the UE measurement reporting uses the dedicated control channel [158, 159], and the increase in data transmission will result in a larger communication overhead in the control channel. However, without the knowledge of the exact data structure (hence data size) of measurement reports in the cellular system, an accurate quantity for the communication overhead is thus unavailable for discussion. This aspect requires further investigation and is considered a limitation of this work. With the deployment of 5G, ultra-dense network deployment is also scheduled and in progress. The future large number of BSs also adds to this potential issue of linear complexity and communication overhead for the proposed DDQN, the effect of which demands further research and quantitative analysis. Similar issues in resource allocation have previously attracted attention and research to reduce input state space of a learning-based algorithm to reduce communication overhead [234], which may also provide a valuable exploration for learning-based HO algorithms.

For the real-world deployment, the HO algorithm is designed to work on a defined local area, leading to the requirement of dedicated local area specifications and individual ANNs to be trained and stored at the network controller for HO decision-making in each local area. Moreover, this work has also assumed that the network controller can automatically switch DDQN models for adjacent areas when a UE crosses the boundaries. These aspects are heavily dependent on mobile operators' physical network deployments. Finally, same as all learning-based solutions, the DDQN-based HO algorithm requires a

significant amount of data to form an effective training dataset. Therefore, the related data collection and processing for the individual local area that influences the scalability of the algorithm remains an important aspect of the algorithm's implementation.

In summary, the proposed DDQN-based HO algorithm, after learning from collected data of the environment, can improve the performance of handover significantly compared to the A3 baseline. Importantly, it only requires a small change to the existing network architecture setup. Therefore the implementation cost should be small, requiring only a software patch. However, other important deployment aspects need to be considered, including state space reduction, control signal overhead assessment, and data collection and processing.

4.4.3 Qualitative comparison with the State-of-the-art Research

After discussing the deployment aspects, a comparison of the proposed DDQN HO algorithm with the A3 RSRP baseline and some of the state-of-the-art research are presented. The contents of this comparison include the methodologies and key design concepts, followed by the input parameters required to operate the algorithms, and what level within the network the algorithms are executed. Impact on the cellular network architecture (LTE and 5G) is also compared at a qualitative level based on the proposed algorithm and system architectures in the original literature. Finally, the communication overhead using the selected algorithms is compared by analysing the type and amount of information required to be transferred on wireless channels for operation. The full comparison is presented in Table. 4.3.

With the exception of the A3 RSRP baseline, all state-of-the-art works are ML-based solutions and are selected based on their ML application types for HO optimisation as discussed in Section 4.1.2. Algorithm designs that considered multiple optimisation objectives such as joint optimisation of HO and radio resource management [169] have not been selected in this comparison in order to focus on HO optimisation.

According to Table. 4.3, ML-based HO parameter optimisation [125] and HO decision-making [172] exploit information from various input parameters for optimisation and require some network adjustment for the HO algorithms to operate. In contrast, predictive HO triggering considers one type of input parameter for accurate predictions, while the prediction is usually performed at the UE level. However, using additional training setups (i.e., the FL setup in [132]) will require a major change to the network architecture for deployment, while also demanding that the ANN model is transmitted through the wireless channel among participants for global model aggregation. Similarly, the two-tier design suggested by Aljeri and Boukerche [174] requires the RNN model to be transmitted via wireless channels and may lead to a large communication overhead, while its HMM-based centralised HO decision requires less modification to the network

architecture compared to the design of Qi *et al.* [132]. In comparison, the proposed DDQN HO design, although aimed at HO decision-making, requires only 1 type of input parameter while maintaining a low impact on the existing network architectures (utilising existing deployment design and input parameters). Therefore, it is a lightweight upgrade considering real-world deployment aspects while delivering near-optimal decisions.

Table 4.3: Comparison between the proposed DDQN HO algorithm and some state-of-the-art

Algorithm	**A3 RSRP [235]**	**Proposed DDQN HO**	**Q-learning-based HO parameter optimisation [125]**	**Multiuser DRL [172]**	**FL proactive HO trigger [132]**	**Two-tier proactive HO optimisation [174]**
Methodology	Static control scheme	RL	RL	Unsupervised learning, RL	Supervised learning	Supervised learning, RL
Key idea	Fixed parameter-based HO trigger to strongest BS	HO decision optimisation	Parameter optimisation for A3-based HO	Mobility-based UE clustering, cluster-level optimal HO policies	SNR prediction for HO trigger	RSSI prediction for HO trigger, RL-based HO decision optimisation
Learning technique setups	/	Deep Q-learning	Traditional Q-learning	K-means clustering, A3C policy gradient	Feedforward ANN-based prediction, Federated Learning	RNN (prediction), HMM (HO decision)
Input parameters	RSRP	RSRP	Throughput, Packet delay, HO frequency	UE mobility, RSRQ, Throughput, HO frequency	SNR	RSSI
Execution agent	Core network	Core network	Core network	Core network	Distributed	UE (prediction), Core network (HO decision)
Impact on network architecture	No	Low	Moderate	High	Very high	High
Communication overhead	Very low	Moderate	Low	Moderate	High	Moderate

4.5 Summary

Vehicles have higher mobility compared to a conventional cellular UE, which leads to reduced connection time between a VUE and a BS. In addition, an increasing number of BSs are being deployed to meet the demand of network densification, leading to more target BSs to consider during HO. Moreover, as the reduction of BS coverage range will cause more HOs for V2N communications due to vehicles' high mobility, it has become much more challenging for conventional HO algorithms to satisfy the requirements in vehicular networks. As a result, ML-based HO optimisation research has utilised various input parameters and enabling technologies to improve HO performance. However, relatively less attention was paid to implementation in unified test environments as well as real-world deployment aspects for ML-based solutions.

A double deep Q-network-based HO algorithm is proposed and evaluated using a realistic bespoke simulation built with the ns-3 full-stack network simulator with the LENA module. The results analysis from 47 HOs throughout the simulation show that the proposed algorithm can reach near-optimal HO decision-making, with a 42.62% reduction in accumulated packet loss (or 6.3% throughput improvement) compared to the A3 RSRP HO algorithm baseline and an average SINR improvement of 3.03 dB for the HO periods of interest. The proposed algorithm also aimed to utilise the existing cellular network configuration with only minor additional information requirements, reporting all neighbouring BS's RSRP instead of only those satisfying predefined conditions (as the A3 RSRP baseline). This makes the algorithm implementable via a software patch.

Chapter 5

Federated Learning-enabled Adaptive Resource Allocation

5.1 Introduction

Allocating resources efficiently and adaptively in vehicular networks is a substantial topic to address the following major challenges: 1. The dynamically changing radio environment of vehicular networks due to their high mobility causes serious impairments with short-lived allocation policies; 2. V2X applications comprise a variety of services with highly differentiated QoS requirements; and 3. The exponential increase in connected vehicles accessing the already congested radio frequency bands demands more advanced allocation algorithms. Spectrum and power are two fundamental radio resources in wireless networks and it is critical to allocate frequency bands and transmit power according to the channel conditions and users' QoS requirements to improve spectral efficiency, mitigate interference, and fulfil the various stringent V2X services.

V2V and V2N are two essential communication types in C-V2X and are exploited to support different applications [15]. The real-time safety-critical message transmission among proximally close vehicles is accomplished via V2V links, with strict latency and reliability requirements, while V2N supports on-board infotainment services via Internet access, requiring high capacity with large bandwidth between VUEs and BSs. Such diverse QoS requirements are challenging for conventional centralised resource allocation approaches to satisfy, which motivated the 3GPP organisation to develop advanced resource allocation approaches for cellular V2X based on the dedicated resource pool design for V2V and V2N links [236]. With a limited frequency spectrum and a significantly increasing number of VUEs, the underlay sidelink architecture provides more opportunities for the coexistence of V2V and V2N links on a shared resource pool. However, this also complicates the network design for interference management,

necessitating efficient resource allocation optimisation for proper spectrum sub-band allocation and transmission power control to meet the diverse QoS requirements from both V2V and V2N links.

Traditionally, model-based analytical algorithms have been widely utilised to optimise resource allocation and interference management for V2X communications [121, 237]. Such methods have clear limitations for vehicular network applications as vehicles' high mobility makes it difficult to obtain precise information such as the Channel State Information (CSI) to solve the formulated problem due to the rapid radio environment changes. Furthermore, analytical solutions are computationally demanding due to the complex problem formation for the V2X environments, and thus face challenges for real-time applications with tight latency requirements [36].

In comparison, ML techniques extracting the underlying patterns from data generated from the wireless environment can cope with vehicular networks' uncertainty and dynamics to provide more intelligent and flexible solutions which are usually hard to derive using conventional optimisation [209]. As a result, ML techniques have been widely explored in various wireless communication research for different system setups in recent years [52]. Deep learning techniques utilising ANNs are capable of further exploiting data patterns and improving task-specific performances, motivating existing research's proposal of both single-agent and multi-agent DRL algorithms for resource allocations.

However, these DRL solutions introduce large communication overheads as information needs to be sent to the central controller or shared between vehicular agents to train the algorithms or for allocation decision-making. Furthermore, it is inconvenient for existing designs to reuse trained ANN models for newly joined agents without any knowledge of the environment. FL is a decentralised ML architecture that enables knowledge sharing among agents through model aggregation and reduces frequent direct information sharing, while also preserving participating agents' data privacy. These advantages of FL motivate this work for an algorithm proposal for decentralised joint spectrum and power allocations in vehicular networks using FL-enhanced DRL.

5.1.1 Related Machine Learning Background

As this work focuses on DRL algorithmic design considering the deployment aspects, some essential background that may significantly impact the deployment of algorithms will first be presented.

5.1.1.1 The Settings of Reinforcement Learning

According to the number of decision agents, RL can be classified into single-agent and multi-agent RL. Only one decision-making agent exists in single-agent RL and the agent takes actions purely based on its own state observations of the environment (either the complete state or a partially observed one). The setting of single-agent RL is considered centralised since both information gathering and action-taking are executed by the decision-making agent, which is equivalent to a central controller. In contrast, multi-agent RL (MARL) contains two or more decision-making agents. MARL also includes different settings, corresponding to the cooperation level among agents via reward design or the information structure as summarised in Figure 5.1.

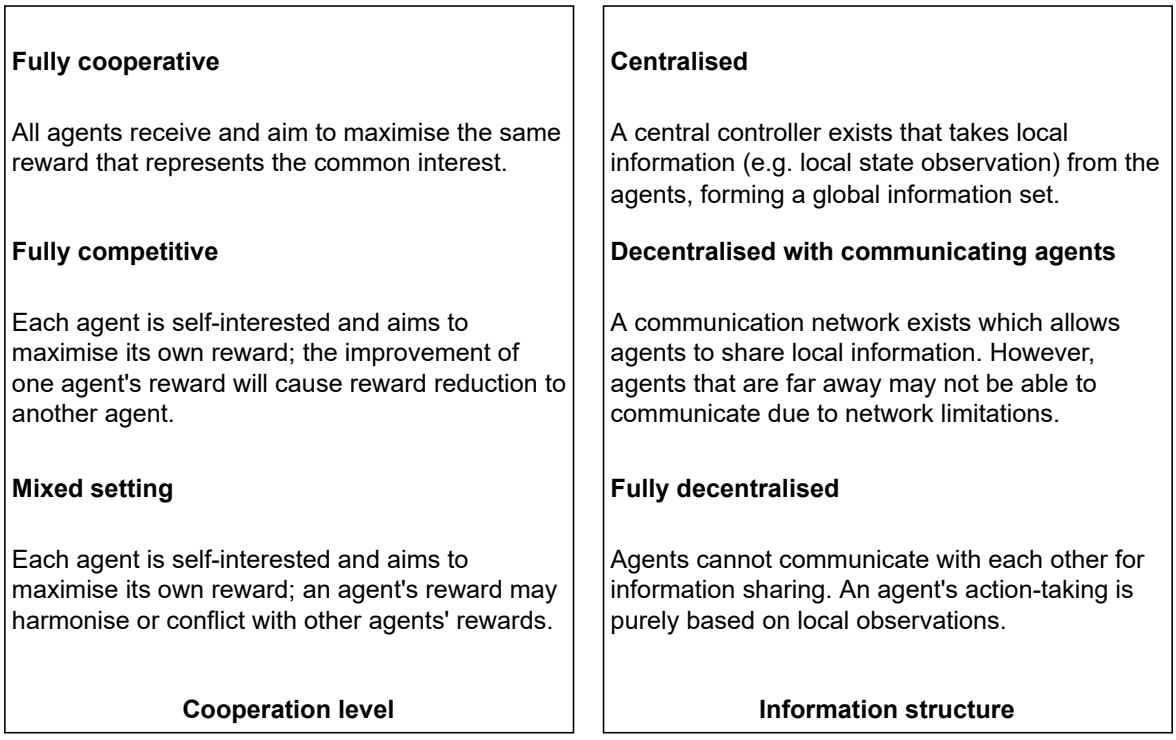


Figure 5.1: Different settings of multi-agent reinforcement learning, recapitulated from the related details in Chapter 3.

Regarding the cooperation levels, fully competitive and mixed settings can be implemented using all three information structures depending on the specific problem. However, the fully cooperative MARL setting requires either the "centralised" or "decentralised with communicating agents" information structure as local information from all agents is required to calculate a shared reward for all agents. Note that only a high-level summary is presented here as more details have already been introduced in Chapter 2.

For deployment in vehicular networks, single-agent RL has large communication overheads as all information needs to be sent to the central controller for decision-making. In comparison, the more decentralised an MARL information structure is, the smaller communication overheads will be due to less information sharing. However, less information sharing causes agents to have reduced global information access, leading to

more uncertainty in agents' state observations hence action-taking and resulting in less stable training.

5.1.1.2 Transfer Learning for MARL

A typical MARL problem has a defined set of agents within the considered environment. However, this assumption or precondition cannot be satisfied when considering the deployment in a vehicular network as vehicles frequently enter and leave local areas. Since learning a complex task from scratch, especially in real-time, is impractical due to the complexity of RL algorithms, reusing the knowledge from another agent or previous experiences becomes vital for learning acceleration and scalable RL deployment [238].

Transfer learning [239] is a technique for knowledge reuse in ML systems. Traditionally, an ML system trains a specific model for the identified task, while transfer learning utilises knowledge extracted from the original task (source task) and updates the ML model for a target task with additional data gathered in the target task. Figure 5.2 illustrates the differences between these two learning processes. Readers interested in the detailed background of transfer learning can refer to the comprehensive surveys and tutorials of [239–241].

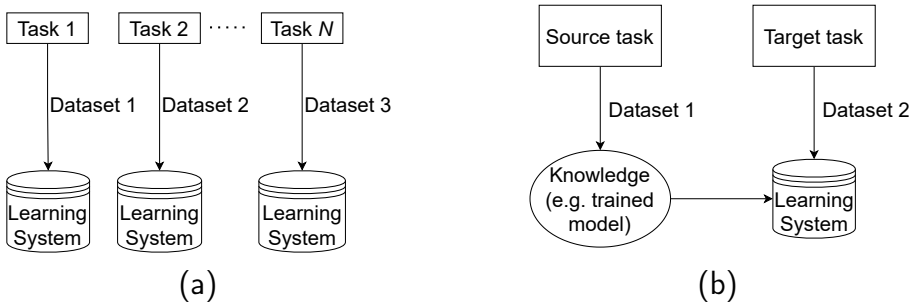


Figure 5.2: The process of traditional ML and transfer learning. (a) Traditional ML. (b) Transfer learning.

For MARL deployment for decentralised resource allocations in vehicular networks, a newly joined vehicular agent can download a copy of the trained model from another agent. The newly jointed agent can then update the copied model with its locally collected data to help faster learning convergence and thus improve its decision-making, avoiding the expensive training frame scratch and performance vacuum. The disadvantage of this approach, however, is that the newly joined agent requires another agent to be nearby and share a trained model, which may not always be satisfied. Additionally, the shared model trained by the other agent's local data may not be suitable for the new agent's situation as the other agent only has its local state observations during its own training. An example scenario for this is two cars in the same street block (possible for model sharing) but travelling through very different routes and thus having different local data (knowledge).

5.1.2 Related Work

For resource allocation optimisation in vehicular networks, ANN-enabled DRL algorithms have made significant advances in applications with high complexity and variability [121]. A sub-band and transmission power allocation scheme for V2V links was proposed in [137] for a spectrum-sharing vehicular network between V2V and V2N links. A centralised deep Q-learning algorithm was developed for the network controller, with V2V pairs taking turns to send their local state observation at every time step for resource allocation decisions to simultaneously minimise V2V transmission latency and interference to V2N links. Also utilising deep Q-learning, a centralised resource allocation scheme was developed by Li *et al.* [242] for MEC [63] in a cellular vehicular network. By jointly considering the MEC server's computation resources and VUEs' transmission power, the centralised agent aims to minimise the overall delay and power consumption for all VUEs.

Centralising the learning setup in vehicular networks has the obvious limitation of communication overheads as information needs to be sent to the centralised agent for decision-making while the agent is under heavy computation demands for both training and execution. Moreover, the centralised agent usually suffers from partial observations of the environment. These factors motivate the multi-agent DRL setup to directly exploit local information at each vehicle to better explore the environment. Liang *et al.* developed a multi-agent deep Q-learning for joint sub-band and V2V transmission power selection in a spectrum-sharing cellular V2X [25]. Each V2V link is modelled as an agent that trains a local ANN model for decision-making while cooperating to maximise V2V payload delivery for all V2V links and the sum V2N transmission rate. Similarly, Yuan *et al.* developed a hierarchical DRL solution for joint sub-band and V2V transmission power selection [138]. They utilised two DRL algorithms, deep Q-learning and DDPG [139], for spectrum allocation and power control, respectively. On top of the DRL resource allocation scheme, a meta-reinforcement learning algorithm was further implemented to adaptively tune the trained allocation scheme in response to different communication scenarios. Furthermore, Gu *et al.* [243] proposed a multi-actor-attention-critic algorithm based on resource allocation in C-V2X. Each VUE was modelled as an agent in a 3GPP standard-compliant sensing-based scheduling setup and selected spectrum resources to reduce V2V packet collisions, and all VUEs shared their state transition observations with the RSU for centralised policy training.

Although achieving satisfying performance, these multi-agent DRL solutions are based on the setting of cooperative learning with the centralised information structure and configured in a "centralised training, distributed execution" manner. The cooperative setting for RL-based decentralised resource allocation in vehicular networks has been preferred because all agents aim to achieve a common objective for optimisation, e.g., system throughput, without being self-interest and competing with each other for radio

resources. However, such a setting requires frequent information sharing between agents and the controller or among agents during the training stage to calculate a cooperative reward or to formulate a global state observation for all agents, and thus have large communication overheads as the centralised single-agent setting.

To reduce the communication overheads in this setting, an additional ANN model was proposed for CSI compression to reduce communication overheads [234]. However, the additional ANN model leads to additional computation resource demand for VUEs during deployment. Furthermore, the above multi-agent DRL solutions provided in the existing research will encounter difficulties for online model updates after deployment. This is because only the agents during the original training phase have a working DRL model. After deployment, newly joined agents must either train a new DRL model from scratch or by using transfer learning techniques, e.g., to update a copy of another agent's model, which has the disadvantages of potential unavailability of another agent or performance downgrade as introduced above. These issues lead to high deployment costs and need to be addressed for real-world applications.

FL is an alternative decentralised learning architecture utilising data distributed across different local learners. As introduced in Section 2.4, each participating learner trains a local model using its local training data, with an aggregator combining all local models' parameters to form a global model and then redistributing the complete model to all learners. Since only the ML models are transmitted between the aggregator and learners, FL does not require frequent data sharing in real-time among learners which also protects the learners' data privacy [244] and has attracted attention for applications in vehicular networks [56]. For resource allocation, FL has been combined with extreme value theory [245] and Lyapunov optimisation [203] for joint power and spectrum allocation in an ultra-reliable low-latency V2V network as designed by Samarakoon *et al.* [246]. FL has also been utilised with DRL algorithms. For instance, a federated multi-agent DRL solution was proposed by Zhang *et al.* [236] for each VUE cluster in a local area for sub-band and V2V transmission power allocation under a cooperative setup via a shared reward design. As the cooperative setting is still configured in their work, local observations for each VUE must be shared to calculate the shared reward, leading to even higher communication overheads combined with the requirement of sending ANN models in the FL training process.

5.1.3 Objectives and Contributions

Compared with a cooperative setting, a mixed setting with self-interested agents does not require direct information sharing among agents to attain a common reward and thus can reduce communication overheads during training. However, a fully distributed information structure in MARL with a mixed learning setting may face the extreme case of independent learning, which can cause non-convergence in the learning process

[86]. FL, as an alternative decentralised learning architecture, can provide another form of information sharing via global model aggregation while avoiding other data transmission. As ANN models are transmitted less frequently, FL is a promising framework for developing multi-agent DRL-based resource allocation algorithms to reduce communication overheads. Moreover, FL enables newly joined agents to directly utilise the global model which avoids the inconvenience of training a new one, further reducing the deployment cost and is another advantage over traditional MARL with transfer learning setups.

However, FL-enhanced non-cooperative multi-agent DRL solutions for resource allocation have received less attention than the cooperative setup due to their inherent challenges and require further investigation. Consequently, a decentralised joint subband and transmission power allocation scheme for cellular vehicular networks is proposed for spectrum-sharing V2V and V2N links. The goal is to match or exceed the performance compared with a cooperative setting while reducing the overall deployment cost. The proposed scheme aims to maximise the successful V2V safety payload delivery ratio while also maximising the overall V2N rate in the system. The main contributions of this chapter are as follows:

- The proposed design utilises FL to form a new decentralised RL information structure and is configured under a mixed setup for the multi-agent DRL algorithmic design. As a result, the proposed solution does not require direct data sharing among participating agents during training, while being fully distributed for execution after training.

- The proposed solution is implemented with two different DRL algorithms to explore the effectiveness of different types DRL algorithms in the considered problem. Simulation results for training show that both algorithms are effective and can successfully converge under the FL-enhanced DRL setup.

- Through extensive simulation implemented following 3GPP specifications through the Python programming language, the proposed FL-enhanced DRL solution outperforms the state-of-the-art MARL benchmark, especially regarding the generalisation ability and online model updates.

- The proposed solution has a much smaller communication cost compared with a state-of-the-art benchmark under the cooperative setting. The communication overheads can further be greatly reduced with little performance sacrifice through quantisation-based model compression in a proof-of-concept analysis.

- The proposed federated-DRL has a much more convenient model-sharing mecha-

nism compared with model transfer while also requiring much less frequent data transmission for online model updates.

5.2 System Model and Problem Formation

This work considers a cellular vehicular network consisting of one BS and multiple VUEs as shown in Figure 5.3, forming J V2N and K V2V links in the network. As discussed in 3GPP's Release 15 and 16 specifications for cellular V2X enhancement [247], V2N and V2V links provide simultaneous support for high data rate infotainment services, and reliable safety message sharing for essential safety applications and advanced driving services, respectively. Note that mode selection for safety messaging is disabled, i.e. VUEs transmit safety messages only via V2V links and V2N links are for infotainment services only. Specifically, this work considers Mode 2 introduced in Section 3.2 for C-V2X architecture as a distributed mechanism for spectrum selection of V2V links, where each vehicle can autonomously select radio resources for its V2V link rather than depending on the BS to allocate resources in a centralised manner.

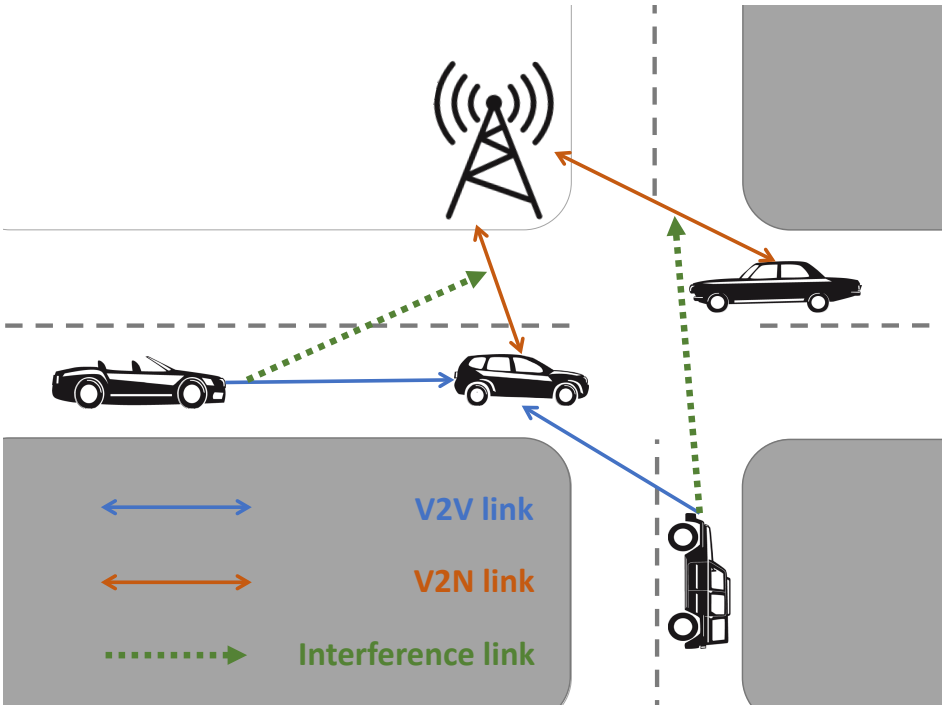


Figure 5.3: A cellular vehicular network of spectrum sharing V2N and V2V links.

Define the set of V2N and V2V links as $\mathcal{J} = \{1, 2, ..., J\}$ and $\mathcal{K} = \{1, 2, ..., K\}$, respectively, while the total bandwidth is divided into M orthogonal spectrum sub-bands, forming the set $\mathcal{M} = \{1, 2, ..., M\}$. Next, denote $\rho_{j,m}, \rho_{k,m}$ $(\rho_{j,m}, \rho_{k,m} \in \{0, 1\})$ for the sub-band selection decisions to V2N and V2V links, i.e. $\rho_{j,m} = 1$ or $\rho_{k,m} = 1$ if the j-th V2N link and k-th V2V link select the m-th sub-band. Each V2V link can select one sub-band at a time, while multiple V2V and V2N links can share the same sub-band to improve the spectrum utilisation. It is further assumed that the orthogonal spectrum sub-bands are uniformly preassigned to each V2N link with fixed

transmission power, i.e. the j-th V2N link always occupies sub-band $\bar{m}_j$ or $\rho_{j,\bar{m}_j} = 1$, $\bar{m}_j = mod(j, M)$.

The channel power gain consisting of both large-scale and small-scale fading components is considered. The channel gains for the j-th V2M link over sub-band $\bar{m}_j$ and the k-th V2V link over sub-band m are represented by $h_{j,\bar{m}_j}$ and $g_{k,m}$, respectively. The interference to the j-th V2N link over the $\bar{m}_j$-th sub-band is $I_{j,\bar{m}_j}$ and similarly, the interference to the k-th V2v link over sub-band m is $I_{k,m}$. The received SINR of the j-th V2N link, and that of the k-th V2V link over sub-band m can be expressed as:

$$SINR^{v2n}_{j,\bar{m}_j} = \frac{p^{v2n} h_{j,\bar{m}_j}}{I_{j,\bar{m}_j} + \sigma^2} \tag{5.1}$$

$$SINR^{v2v}_{k,m} = \frac{p^{v2v}_{k,m} g_{k,m}}{I_{k,m} + \sigma^2} \tag{5.2}$$

where p^{v2n} and σ^2 stand for the fixed V2N transmission power and the noise power, respectively, while ${p^{v2v}}_{k,m}$ represents the k-th V2V link's transmission power over sub-band m. $I_{j,\bar{m}_j}$ and $I_{k,m}$ denote the total interference power to V2N link j over sub-band $\bar{m}_j$ and V2V link k over sub-band m, calculated as:

$$\begin{aligned} I_{j,\bar{m}_j} = & \sum_{j'=1}^{J,\ j' \neq j} p^{v2n} h_{j',\bar{m}_{j'}} \\ & + \sum_{k=1}^{K} \rho_{k,\bar{m}_j} p^{v2v}_k g_{k,\bar{m}_j};\ \bar{m}_{j'} = \bar{m}_j \end{aligned} \tag{5.3}$$

$$\begin{aligned} I_{k,m} = & \sum_{j=1}^{J} p^{v2n} h_{j,\bar{m}_j} \\ & + \sum_{k'=1}^{K,\ k' \neq k} \rho_{k',m} p^{v2v}_{k'} g_{k',m};\ \bar{m}_j = m \end{aligned} \tag{5.4}$$

where $h_{j',\bar{m}_j}$ and $g_{k,\bar{m}_j}$ represent the interfering channel gains at the j-th V2N receiver from the transmitter of the j'-th V2N link and the k-th V2V link over sub-band $\bar{m}_j$. Similarly, $h_{j,\bar{m}_j}$ and $g_{k',m}$ stand for the interfering channel gains at the k-th V2V receiver from the j-th V2N link and k'-th V2V link over the m-th sub-band.

With the above definitions, the data rate of the j-th V2N link C^{v2n}_j and k-th V2V link

C_k^{v2v} can then be obtained by (5.5) and (5.6), according to Shannon capacity theory:

$$C_j^{v2n} = BW \log(1 + SINR_{j,\bar{m}_j}^{v2n}) \tag{5.5}$$

$$C_k^{v2v} = \sum_{m=1}^{M} \rho_{k,m} \cdot BW \log(1 + SINR_{k,m}^{v2v}),\ \rho_{k,m} \in \{0, 1\} \tag{5.6}$$

where BW is a constant representing sub-bands' bandwidth. Note that as V2V links can only choose 1 sub-band at a time step, (5.6) essentially finds sub-band m such that $\rho_{k,m} = 1$ since $\rho_{k,m'} = 0$ where $m' \neq m$.

As discussed above, V2V and V2V links need to satisfy differentiated service requirements. V2N links aim to provide infotainment services requiring a high data rate, forming the clear goal of maximising the V2N sum rate $\sum_{j=1}^{J} C_j^{v2n}$.

In comparison, V2V links focus on the reliable transmission of safety-critical messages, which leads to the network goal of maximising the average successful transmission probability among all V2V links. For the k-th V2V link, the successful transmission of payload u is represented by a binary value $\zeta_{k,u} \in \{0, 1\}$, calculated via the following Boolean expression:

$$\zeta_{k,u} = \sum_{t=t_u}^{t_u + t_{max}/\Delta t} \Delta t \cdot C_k^{v2v}(t) \geq B_u \tag{5.7}$$

where B_u is the size of the u-th payload for V2V link k. t_u and Δt stand for the starting time of payload u's transmission and a time step's duration, respectively. t is the time step index corresponding to the transmission time, and t_{max} is the maximum delay tolerance for the V2V payload transmission. Based on (5.7), the delivery of V2V payload u for V2V link k is successful ($\zeta_{k,u} = 1$) only if the delivery time consumption does not exceed t_{max}, and unsuccessful ($\zeta_{k,u} = 0$) otherwise. Note that t_u of each V2V link to transmit payload u is the same for the synchronous transmission setup considered in this work, and may differ if an asynchronous setup is considered. Therefore, the successful transmission probability η_k of V2V link k during a given transmission period, can be expressed as follows, assuming $N_{payload}$ V2V payloads have been transmitted during this period:

$$\eta_k = \frac{\sum_{u=1}^{N_{payload}} \zeta_{k,u}}{N_{payload}} \tag{5.8}$$

Following the above definitions, the optimisation problem for the considered spectrum-sharing scenario can be formulated as follows at each resource allocation time step:

$$\max_{\hat{\rho},\hat{P}^{v2v}} \; E = \frac{\sum_{k=1}^{K} \eta_k}{K} \tag{5.9a}$$

$$\text{s.t.} \; \max \Lambda = \sum_{j=1}^{J} C_j^{v2n}, \tag{5.9b}$$

$$\sum_{m=1}^{M} \rho_{k,m} \leq 1, \forall k \tag{5.9c}$$

$$0 \leq p_{k,m}^{v2v} \leq P_{max}^{v2v}, \forall k \tag{5.9d}$$

where E and Λ are the system-level V2V payload delivery ratio and V2N data rate, respectively. $\hat{\rho} = \{\rho_{1,1}, ..., \rho_{k,m}, ..., \rho_{K,M}\}$ is the set of the sub-band selection indicators, while $\hat{P}^{v2v} = \{p_{1,1}^{v2v}, ..., p_{k,m}^{v2v}, p_{K,M}^{v2v}\}$ is V2V power allocation indicator set, and P_{max}^{v2v} is the maximum transmission power of a V2V transmitter. For power control, this work considers discrete power control [248] and assumes the transmit power of VUEs has N_{pow} levels.

The formulated resource allocation problem in (5.9) is a multi-objective that aims to maximise both the V2V payload delivery probabilities expressed in (5.9a) and the V2N sum rates expressed by (5.9b) at the same time. Note that without (5.9b) the optimisation will only aim to maximise V2V delivery probability regardless of V2N service quality, while vice versa for the lack of (5.9a). Furthermore, as the problem involves sequential decision-making over multiple resource allocation time steps, this problem is NP-hard [138] and is highly challenging to solve using a conventional optimisation solution, especially considering the time limit set to produce resource allocation decisions for real-time applications.

5.3 DRL-based Resource Allocation Algorithm

The considered spectrum sharing scenario is modelled as a MARL problem, and an FL-enhanced decentralised DRL solution is proposed to the original optimisation problem in Eq. (5.9).

To utilise DRL algorithms, the problem is first formulated as a Markov Game [85] with multiple decision-making agents, as described in Figure 2.2. In a Markov Game, multiple decision-making agents take actions according to their observations of the current environmental state at time step t, forming a joint action. After the action execution, the agents receive the observations of the resultant state at $t+1$ and corresponding immediate numeric rewards generated by the environment. It is assumed that the scheduling period of V2V pairs is set to be 1 time step for the whole set of V2V pairs. Through many rounds of training, the agents can develop an optimal policy to adopt in taking action within the given environment to maximise the long-term accumulated reward.

This section covers the design of the three key elements of the Markov Game model, i.e., state space, action space, and immediate reward, while also proposing two DRL algorithms (1 on-policy and 1 off-policy) for resource allocation to explore the performance between the different types of DRL algorithm.

5.3.1 State and Observation Space

For the considered problem, the true environmental state at any time step $\mathcal{Z}_t$ may include the global channel information, agent behaviours, and other related information. However, individual agents cannot gather such global information and can only receive state information via its observation, forming an observation space denoted as $\mathcal{S}_t$ for all agents. To reduce communication overhead, the state observation of an agent at each time step t should only include local information at each V2V receiver, which includes the sensed V2V link SINR at each sub-band $\overline{SINR}_t$, the sub-band and transmission power selection for the last time step $\bar{\rho}_{t-1}$ and $\bar{p}^{v2v}_{t-1}$, the remaining V2V payload B'_t, and the remaining time budget $T_{b,t}$.

For N_{sb} sub-bands, $\overline{SINR}_t = \{\text{SINR}_1, ..., \text{SINR}_{N_{sb}}\}$. The information of $\bar{\rho}_{t-1}$ and $\bar{p^{v2v}_{t-1}}$ is designed to be represented via one-hot encoding [213] to avoid potential confusion with a decimal SINR value. For example, if an agent selects sub-band 2 with the 5 sub-bands in total, then such action after one-hot encoding becomes the vector $\bar{\rho} = \{0, 1, 0, 0, 0\}$, while $\bar{p}^{v2v}$ is also represented in this manner. In addition, B' and T_b are two scalar values. As a result, s_t of an agent can then be formally defined as:

$$s_t = \{\overline{SINR}; \bar{\rho}_{t-1}; \bar{p^{v2v}_t}; B'_t, T_b, t\} \tag{5.10}$$

which has the size of $2N_{sb} + N_{pow} + 2$ which is a constant for a defined system setup, while the observation space S_t is the set of s_t for all agents at time step t.

5.3.2 Action Space

The problem formation (5.9) defines the actions for each agent to be the spectrum sub-band selection and V2V transmission power control at each time step. Consistent with the notation in Section 5.2, such an action is defined as $\{m, p^{v2v}\}$. The spectrum naturally breaks into M disjoint sub-bands, while the V2V transmission power is defined to have N_p levels as previously defined, such that $m \in \mathcal{M}$ and $p^{v2v} \in \{0, ..., \frac{p^{v2v}_{max}}{N_p - 1}\}$, forming an action space sized MN_p representing all possible $\{m, p^{v2v}\}$ combinations. Therefore, the action space is defined as $a_t = \{1, ..., MN_p\}$, with each action in a_t associated with a combination of spectrum sub-band and power selection.

5.3.3 Reward Design

RL is advantageous for decision optimisation by designing a reward function to represent the hard-to-optimise objectives and constraints with a high degree of flexibility. After an agent takes an action based on its policy for a state observation, it receives an immediate reward generated by the environment, and the reward can indicate the performance of this decision execution.

Following the optimisation objective in (5.9), an optimal outcome for each V2V agent maximises the successful delivery probability of a V2V payload within the time tolerance while sacrificing the V2N transmission rate as little as possible, which is the core concept for the reward design. Note that the dynamic vehicular networks may require frequent updates to the ANN after deployment, which makes a cooperative reward design expensive with respect to communication overheads as in the work of Liang *et al.* [25] since the reward needs to be transmitted and shared among all participating agents. As a result, the immediate reward is also designed to only utilise the local information of each agent, which is expressed as:

$$r_t = \begin{cases} C_t^{v2v} - \Delta B, & \text{if } B'_t > 0 \\ C_t^{v2v} + C_t^{v2n}, & \text{else} \end{cases} \tag{5.11}$$

where ΔB is the average transmission rate for a given V2V payload with size B under a defined network setup, while C^{v2v} and C^{v2n} are the transmission rate of the agent's corresponding V2V link and V2N link at the same receiver, consistent with the notions in Section 5.2. The first case of (5.11) is the normalised V2V rate with respect to ΔB to give the agent a clear baseline to satisfy the V2V payload tolerance, while the second case of it stands for the sum V2N rate maximisation goal when the V2V payload is delivered, plus an additional reward element of V2V rate to further encourage the agent to achieve the V2V payload delivery as the second case leads to a much higher reward than the first case.

Since RL aims to find an optimal policy that can attain the expected reward from the state in the long term, the cumulative discounted reward can be defined as follows, recapitulated from Eq. (5.12):

$$G_t = \sum_{l=0}^{\infty} \gamma r_{t+l} \tag{5.12}$$

where $\gamma_i \in [0, 1]$ stands for the discount factor used to balance weight between the future rewards and the immediate reward, and the cumulative reward G_t for state s_t equals the immediate reward r_t when $\gamma_i = 0$.

5.3.4 DRL Algorithmic Implementation

Two DRL algorithms are proposed to solve the formulated Markov Game for the considered resource allocation optimisation problem. For the off-policy method, the well-known DQN method initially proposed by Mnih *et al.* [105] is utilised. To be specific, this work develops the algorithm based on the DDQN variant of the DQN [107]. For the on-policy DRL algorithm, this work adopts the A2C algorithm [110]. The essential mechanisms of these algorithms are introduced in Chapter 2 and thus not repeated here.

The DDQN and A2C algorithmic implementations for the proposed solution are summarised in Algorithm 5.1. Note that these presented algorithms are for 1 time step t, which by calling upon multiple iterations can represent the complete training phase.

5.4 Decentralised Resource Allocation by Federated DRL

The proposed DRL algorithm in Section 5.3 to solve the considered problem in (5.9) is under a mixed setting without any information exchange among the V2V agents. Although this helps the reduction of control signal overheads in communication, this setup leads to the extreme case of independent learning, where agents learn on local state observation, action, and rewards and suffer from non-convergence in general [86]. To deal with this potential issue, this work explores the federated learning (FL) setup among the agents to indirectly share their local information and improve the convergence probability. The fundamentals of FL are introduced in Chapter 2.

The core process of federated DRL is the global aggregation that defines how local models are combined to form a global model. This work adopts the federated averaging method [249] for global aggregation. With N_{agent} V2V pairs participating in the federated DRL,

Algorithm 5.1: DRL-based Decentralised Algorithms

/* t - current time step; s_t - state observation; */
/* T - total time step. */

Function TrainingA2C(t, s_t, T):
  Data: θ_a - Actor-network; θ_c - Critic-network;
  γ - Discount factor; $\mathbb{M}$- Memory for n-step TD;
  Execute a_t according to policy $\pi(a_t|s_t;\theta_a)$;
  Observe s'_t and r_t;
  if $t < T$ *or* s_t *not terminal* **then**
    Store (s_t, a_t, s'_t, r_t) in $\mathbb{M}$;
  else
    $$R \leftarrow \begin{cases} 0, & \text{for terminal state} \\ V(s_i;\theta_c), & \text{for non-terminal } s_i \end{cases}$$
    for $i \leftarrow 1$ **to** T **do**
      $R \leftarrow r_i + \gamma R$;
      Calculate loss for θ_a: $\| \log \pi(s_i, a_i; \theta_a) \cdot (R - V(s_i;\theta_c)\|^2$;
      Calculate loss for θ_c: $\|R - V(s, a; \theta_c)\|^2$;
    end
    Update parameters of θ_a and θ_c;
  end

Function TrainingDQN(t, s_t, T):
  Data: θ_q - Training DQN; θ'_q - Target DQN;
  $\mathbb{B}$ - Reply buffer; N_{rp} - Replay buffer capacity;
  N_{mb} - Mini-batch size; N_f - θ'_q update frequency;
  γ - Discount factor; ϵ - Random action probability.
  $$a_t \leftarrow \begin{cases} \text{a random action}, & \text{with } \epsilon \\ argmax_a\ Q(s_t, a; \theta_q), & \text{with} 1 - \epsilon \end{cases}$$
  Execute a_t and **observe** s'_t and r_t;
  Store (s_t, a_t, s'_t, r_t) in $\mathbb{B}$;
  if N_{rp} *is reached for* $\mathbb{B}$ **then**
    Delete the oldest sample in $\mathbb{B}$;
  end
  if $\mathbb{B}$ *reaches* N_{mb} *samples and* $t = T$ **then**
    Sample a mini-batch of (s, a, s', r) from $\mathbb{B}$;
    foreach *(s, a, s', r) sample* **do**
      Construct target value:
      $$y \leftarrow \begin{cases} r, & \text{if } s' \text{ is the terminal state} \\ r + \gamma Q(s', argmax_{a'} Q(s', a'; \theta_q); \theta'_q), & \text{else} \end{cases}$$
      Calculate loss $\|y - Q(s, a; \theta_q)\|^2$;
    end
    Update parameters of θ_q
  end
  if N_f *is reached* **then**
    $\theta'_q \leftarrow \theta_q$; // Copy θ_q parameters to θ'_q
  end

the ANN weights of the global model θ_g can be updated by:

$$\theta_g \leftarrow \frac{1}{N_{agent}} \cdot \sum_{id=0}^{N_{agent}} \theta_{agent_{id}} \tag{5.13}$$

where $\theta_{agent_{id}}$ is the ANN weights of $agent_{id}$'s local model. Note that this equation (5.13) is a simple global aggregation method that assumes the same aggregation weight from every participant is utilised in this work. However, as the focus of this work is to explore how federated DRL performs compared with a cooperative DRL setup, we assume that the scenario during training contains a predefined set of participants without newly added and dropped participants. Therefore, the related aspects of FL training are beyond the scope of this work, which makes the aggregation method suitable for the considered scenario. With the important aspects introduced, the proposed federated DRL-based algorithm for resource allocation is summarised in Algorithm 5.2. The DDQN and A2C algorithms with this FL setting are termed FL-DQN and FL-A2C, respectively.

Algorithm 5.2: Federated DRL algorithm for training

Initialise : Global model: Actor-Critic pair $\{\theta_a^g, \theta_c^g\}$ or DQN θ_q^g. BS aggregator distributes the global model to each participant

Data: N_{agg} - Global aggregation step size;

Output : Trained global model

foreach *communication round* **do**
  Get V2V payload size for each V2V link;
  for $t \leftarrow 1$ **to** T **do**
    All agents **observes** the state s_t ;
    All agents performs `TrainingA2C(`t, s_t, T`)`
      or `TrainingDQN(`t, s_t, T`)`;
  end
  if N_{agg} *is reached* **then**
    All agents upload local models to BS ;
    `/* Global model aggregation as in (5.13) */`
    $\theta^g \leftarrow \frac{1}{N_{agent}} \cdot \sum_{id=0}^{N_{agent}} \theta^{agent_{id}}$;
    BS distributes the global model to all agents;
  end
end

5.4.1 Model Quantisation to Reduce Communication Overhead

The proposed federated-DRL framework enables knowledge sharing among the learning agents via global aggregation without direct communication and data transmission. However, the global aggregation requires local ANN models to be transmitted from participating V2V agents to the BS aggregator, which also causes communication

overheads. Therefore, effective ANN model compression that reduces model size while maintaining as much performance as possible becomes valuable for the proposed federated-DRL framework for communication cost reduction.

Model compression has attracted much research attention for model size reduction and acceleration, with techniques such as pruning, quantisation, low-rank factorisation, etc. [250]. This work adopts the quantisation technique for model compression, which is effective for fully connected ANNs usually used for DRL algorithms and has high robustness [251]. At present, ANN weights are stored as 32-bit floating-point numbers. Therefore, reducing the number of bits used to represent the weights and activations can lead to a significant reduction in the size of trained deep ANNs, e.g. decreasing the 32-bit storage to 8 bits can directly result in a 4-time model size reduction. More detail regarding the quantisation implementation is covered in Section 5.5. This work focuses on proof-of-concept exploration of whether quantisation preserves the overall performance for the proposed algorithms while bringing significant model size and thus, communication overhead reduction. As a result, aspects such as combining different kinds of compression techniques for compression optimisation and detailed real-world deployment are beyond the scope of this work.

5.5 Evaluation Configurations

In this section, the experimental setup, benchmarks, and performance metrics are presented to evaluate the effectiveness of the proposed federated-DRL algorithm in a single-cell cellular vehicular network. Simulations have been implemented via Python 3.10 using Numpy [252] and Pytorch [204] on the Archlinux platform (ANN model size based on the corresponding file system), and result visualisation is generated via Matplotlib [253].

5.5.1 Simulation Setups

The simulation setup is based on the urban case's evaluation methodology as defined in [180]. This TR includes detailed vehicle types and mobility models, V2V and V2N channel models, V2V data traffic models, etc. Major simulation parameters are summarised in Table 5.1. Specifically, the simulation arena is set to contain four urban grids as defined in Table A-1, Annex A. in [180] to focus on a local area centred at a crossroad intersection. Moreover, the network traffic model is based on Model 2 (medium traffic intensity) in Section 6.1.5 of [180], while changing the inter-packet arrival time and latency requirement to 50 ms. Also note that as in [25], the BS transmission power is set to 23 dBm to mitigate V2V interference and the V2V power control options in this work are set to four levels, i.e., [23, 10, 5, -100] dBm, where the -100 dBm represents "zero" V2V transmission power (since -100 dBm is actually finite

but is very small).

Table 5.1: Simulation configurations based on 3GPP TR 37.885 [180]

Setup properties	**Parameters and values**
Simulation time	30 Seconds
Carrier frequency	6 GHz
Total bandwidth	140 MHz
Number of V2N links	5
Number of V2V links	5
Antenna height	BS: 10 m VUE: 1.5 m
Antenna gain	BS: 8 dBi VUE: 3 dBi
VUE mobility model	Option A for urban case in [180]: VUE speed: 60 km/h; VUE goes straight, left, right at an intersection with probability: 0.5, 0.25, 0.25.
Pathloss model	V2N: 3GPP Urban Micro V2V: 3GPP Urban case
Transmission power	BS: 24 dBm UE: [23, 10, 5, -100] dBm
Fast fading	Rayleigh fading
Shadowing	V2V: As in Table A.1.4-1 of [254] V2N: As in Table 7.4.1-1 of [222]
Receiver noise figure	BS: 5 dB VUE: 9 dB
Noise Power	-114 dBm
Network traffic model	Periodic traffic, 10 ms per packet arrival
V2V payload size	1200 bytes (20% probability), 800 bytes (80% probability)
V2V latency requirement	10 ms

The ANNs are configured to be the same wherever possible for all considered algorithms for the performance evaluation, including the DRL-based benchmarks introduced in Section 5.5.2. These configurations are summarised in Table 5.2. To be specific, the Rectified Linear Unit (ReLU), i.e., $f(x) = max(0, x)$, is used as the activation function for DQN and Critic Network, while the Tanh and Softmax are used as the activation for the actor-network to sample categorical actions. Moreover, the "Adam with decoupled weight decay" (AdamW) optimiser [228] is used to train the ANN parameters with an initial learning rate of 0.0001, and an exponential learning rate decay of 0.97 applied every episode.

In addition, the global aggregation is set for every 30 seconds for the federated-DRL solutions, while the Reply buffer is configured to have the capacity of $2e^6$ with a mini-batch sample size of 1280 for the DDQN, which is also true for the considered benchmarks using the DQN algorithm.

Table 5.2: Neural network hyperparameter configurations

Hyperparameter	**Value**		
	DQN	Actor Network	Critic Network
Hidden layer number	3		
Neuron size	$512 \times 256 \times 128$		
Hidden layer activation function	ReLU	Tanh	ReLU
Output layer activation function	Linear	Softmax	Linear
Optimiser	AdamW [228]		
Learning rate	10^{-4}		
Learning rate decay	0.97		
γ in (5.12)	0.98		

5.5.2 Benchmarks

The proposed FL-DQN and FL-A2C's performances are compared against two established DRL-based benchmark methods in the literature and the baseline performance from a fully random strategy. Furthermore, the performances are also compared with the theoretical performance upper bounds for V2N and V2V links, respectively. Together, these benchmarks are introduced as follows:

- **C-MARL**: This is the multi-agent DRL-based resource allocation scheme developed in [25] where each V2V link works as an agent. C-MARL is implemented in a "centralised training, distributed execution" manner, that each agent makes its own actions based on the state observations using only local information, while a reward based on both V2V and V2N rates for all agents is shared among the agents during training.

- **Centralised**: This is the centralised DRL-based resource allocation proposed in [137]. At each time step, the centralised agent decides the action for the spectrum sub-band and V2V power level for one V2V link, while the allocation decisions for other V2V links remain the same (turn-based resource allocation). State and reward information is sent to the centralised agent from each V2V pair.

- **Random**: This is the baseline that chooses the spectrum sub-band and V2V transmission power for each V2V link in a random manner, i.e., without any resource allocation optimisation strategy.

- **max_v2n**: This gives the maximum V2N sum rate upper bound for a given scenario, calculated by disabling all V2V links for theoretical V2N performance. In this case, the packet delivery rates for all V2V links are zero, which is equivalent to a conventional cellular network with all UEs being vehicles.

- **max_v2v**: This produces the successful V2V delivery ratio upper bound. Following the problem formation, this work essentially tries to solve a combinatorial optimisation with discrete spectrum sub-band and V2V transmission power options. As a higher V2V transmission rate leads to a higher probability of successful V2V payload delivery, this method exhaustively searches the action space of every V2V agent in each time step to maximise the V2V transmission rate while completely ignoring the V2N service requirements and regardless of its computational complexity.

Note that the two theoretical upper bounds are achieved by ignoring important factors and thus are for reference only.

5.5.3 Performance Metrics

For the performance evaluation, all learning algorithms are first trained in an episodic manner that runs the same environment (vehicle initial positions and trajectories, fading behaviours) multiple times. After the training, the algorithms are then evaluated in a testing environment whose data has not been experienced by the algorithms. The evaluation is centred around the two network performance metrics:

1. **V2V payload delivery ratio** E: This is the primary metric as safety applications are more essential and critical for V2X applications as defined in (5.8) and $0 \leq E \leq 1$.

2. **Sum V2N transmission rate** Λ: This is the sum transmission rate of all V2N links defined in (5.9b).

3. **Aggregated gain** $\mathbb{G}$: As the optimisation problem is a max-max problem with two optimisations expressed by the above two metrics, an aggregated gain is defined to achieve a more grounded comparison, defined as:

$$\mathbb{G} = \frac{E_{drl} - E_{rnd}}{E_{rnd}} + \frac{\Lambda_{drl} - \Lambda_{rnd}}{\Lambda_{rnd}} \tag{5.14}$$

where E_{drl} and E_{rnd} stand for the V2V payload delivery ratio obtained by a considered DRL algorithm, e.g., FL-DQN, and the Random benchmark, respectively. The same applies to Λ_{drl} and Λ_{rnd}, which are the sum V2N transmission rate obtained by the corresponding algorithms. This metric indicates how much gain considering both the above metrics an algorithm can achieve compared with the Random benchmark's bottom line performance. Without losing the generalisation ability, E and Λ are set to have the same weight in the summation of (5.14).

In addition to the above performance metric, the communication cost penalty and the convergence behaviours are also analysed for the considered learning algorithms.

5.6 Experimental Results

To produce performance results, an offline training stage under the same configuration was first carried out for all learning-based algorithms, i.e., Centralised, C-MARL, and the two versions of the proposed solution, FL-DQN and FL-A2C. After training, the algorithms are tested in the following scenarios:

- The training environment for direct performance evaluation. This shows the raw performance after training in the completely same environment with VUEs' moving trajectories unaltered, which does not reflect reality and is presented for completeness.

- A direct generalisation test where the algorithms are deployed in a different scenario, i.e., VUEs have different moving trajectories compared with the training scenario.

- An online update scenario where the number of VUEs is not changed while one of the agents does not have a trained model and is treated as newly joined. This scenario considers the online model update and leaving/joining agents, which is considered to most emulate a real-world situation.

In addition to the performance comparison, a proof-of-concept comparison between C-MARL and the proposed solution regarding the communication cost is also presented.

5.6.1 Performance Results: The Training Scenario

5.6.1.1 Convergence Analysis

Since all considered learning algorithms are DRL algorithms, the convergence behaviour will give valuable insights into how each algorithm performs during the training process, which is shown in Figure 5.4.

From Figure 5.4a, the overall convergence trends of the proposed solution's two implementations are highly similar, which is expected as they have the same Markov Game modelling. Nevertheless, two aspects are worth noting, which are at the beginning of the training process and after Episode 20. After Episode 1, FL-A2C has double the cumulative return than FL-DQN. The main reason for this is that FL-DQN follows the ϵ-greedy policy selection that a random action will be taken with a probability of ϵ. While ϵ is set to 1 for Episode 1, FL-DQN essentially follows a random policy during

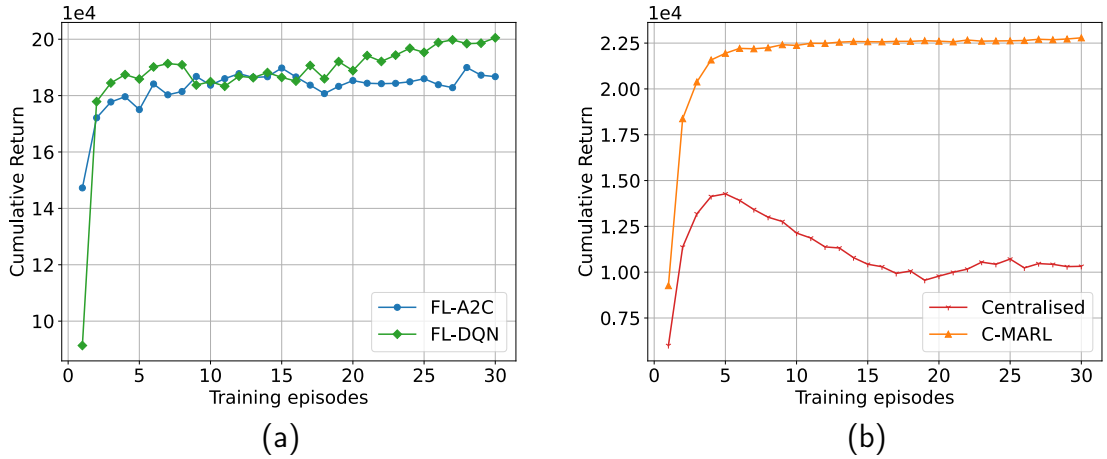


Figure 5.4: Cumulative return per episode for all considered DRL algorithms. (a) The proposed FL-A2C and FL-DQN. (b) The benchmarks.

this episode. In comparison, FL-A2C follows behaviour policy during the training while also improving the policy, leading to a much higher return. After Episode 20, FL-DQN has an increasing cumulative return while the improvement plateaued for FL-A2C. A possible reason for this is that FL-A2C can only update its policy with the latest data, while FL-DQN has a higher data efficiency by exploiting data stored in the experience replay buffer and can find a more deeply hidden pattern to optimise.

Regarding the two benchmark results, C-MARL has a very smooth convergence line, as in Figure 5.4b, due to its cooperative and direct information-sharing design with all agents aiming to achieve a common goal. It has almost no fluctuation compared with FL-A2C and FL-DQN, which are implemented with a mixed setting where self-interest agents may conflict with each other, causing system-wise performance downgrade and thus reduced return. In comparison, the Centralised benchmark has a diverging behaviour starting from Episode 5 although in the same learning configuration. This is because, for the Centralised benchmark, the network controller makes decisions for each V2V link by taking turns, which causes instability in the RL environment.

5.6.1.2 Performance Results

The performance results regarding the two performance metrics, E and Λ, are presented in Figure 5.5. As can be expected from the convergence analysis, C-MARL has the highest E for its cooperative setting with smooth training. This is followed by FL-DQN whose $E = 0.83$, 0.06 lower than that of C-MARL. Also shown in the convergence analysis, FL-A2C has a smaller E than FL-DQN due to having a lower cumulative return throughout the training. Furthermore, since the Centralised benchmark goes diverging during training, it also has the lowest E, which is very close to the bottom-line performance generated by the Random baseline. As for Λ, all considered algorithms have values within 10% of the Random baseline's performance. Note that the results

obtained by max_v2v and max_v2n only provide the theoretical upper bounds for both E and Λ, which is not practically achievable for the considered optimisation problem following their definition in Section 5.5.2 above.

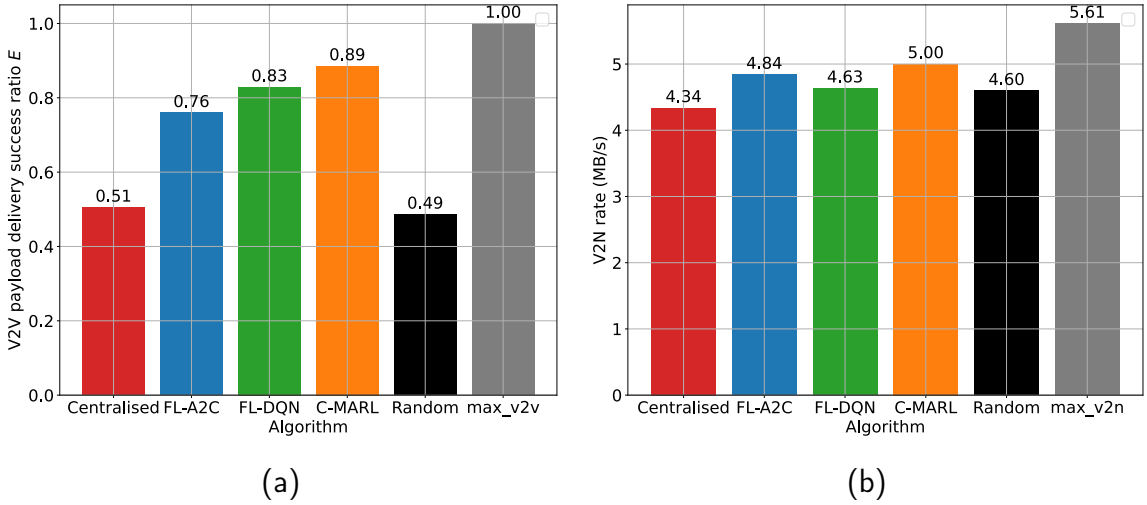


Figure 5.5: The performance results of different benchmarks in the training scenario. (a) System-wise Successful V2V payload delivery ratio. (b) Sum V2N rate system-wise.

Table 5.3 shows the aggregated gain $\mathbb{G}$ (defined in Section 5.5.3) of all four DRL algorithms with respect to the Random baseline. Consistent with the results in Figure 5.5, C-MARL has the highest $\mathbb{G}$, and the second best performing algorithm in the training scenario is FL-DQN, followed by FL-A2C. As the Centralised benchmark becomes divergent during training, it ends up with a negative $\mathbb{G}$ in this scenario. However, the results of the training environment can only show the performance in an idealised situation (the same scenario for training), which is incomplete and needs further elaboration by the generalisation tests.

Table 5.3: Aggregated gain $\mathbb{G}$ with respect to the Random baseline for all considered DRL algorithms in the training scenario.

Algorithm	**FL-A2C**	**FL-DQN**	**C-MARL**	**Centralised**
Aggregated Gain $\mathbb{G}$	0.603	0.700	0.903	-0.016

5.6.2 Performance Results: The Testing Scenario

Similar to the performance evaluation in the training scenario, another round of performance evaluation is executed for generalisation tests. To obtain the results, all considered benchmarks are deployed into a new scenario of the same environment, i.e., the trajectories of each VUE, and thus the radio environment, are different from those in the training scenario. However, other settings such as the simulation parameters are kept the same. Moreover, the trained ANN models of all learning algorithms are directly used in this testing scenario without any online updates. Therefore, it is anticipated that all DRL algorithms will experience performance decline because changing the VUEs' trajectories will lead to a state observation and transaction change that the algorithms did not learn during training.

Figure 5.6 shows the performance results with respect to E and Λ for the testing scenario. Regarding E, all DRL algorithms have met performance downgrade, with the best performing algorithm in this scenario changing from C-MARL to FL-DQN, whose $E = 0.8$ has the lowest performance decline. This is followed by FL-A2C, which has $E = 0.69$ with the second lowest drop. In comparison, C-MARL encounters a stiff performance drop of 0.26 regarding E. Regarding the metric Λ, a similar pattern can be found compared to that in the training scenario, with all learning algorithms having Λ within the 10% range of the Random baseline's value. As for $\mathbb{G}$, Table 5.4 shows the corresponding results of all four DRL algorithms with respect to the Random baseline, with all DRL algorithms having a reduced performance compared with the same metric in the training scenario. Also, as expected, FL-DQN now has the highest $\mathbb{G}$ of the four DRL algorithms, followed by FL-A2C.

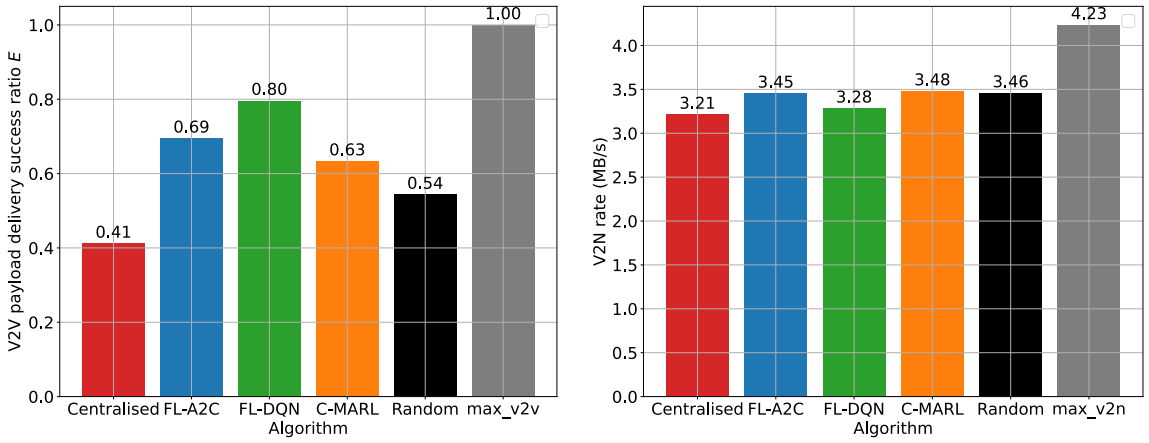


Figure 5.6: The performance results of different benchmarks in the testing scenario without ANN online updating. (a) System-wise Successful V2V payload delivery ratio. (b) Sum V2N rate system-wise.

It is anticipated that the DRL algorithms will have a performance downgrade in this direct generalisation test. The reason is that none of the DRL algorithms have previously experienced the testing scenario due to the change of vehicles' moving trajectories and hence environmental state transitions, leading to the lack of knowledge for trained ANN models. However, the proposed solution (both implemented versions) has a relatively smaller reduction regarding the performance metrics, while the best-performing algorithm in the training scenario, i.e., C-MARL, has the steepest performance decline. A possible explanation for this is that the proposed scheme has self-interest agents while C-MARL has cooperating agents through different reward designs. For the proposed FL scheme, the agents map the state observations to corresponding decisions that maximise their own reward. When directly applied to a new environment, this direct mapping is less affected by the different state transition patterns as agents only focus on their own interests. For C-MARL, however, the reward considers other agents' service quality. When the environment changes without ANN model updates, such cooperative design

leads to impairment between the new environmental information and existing knowledge learned by the models and thus a more significant performance downgrade. Therefore, it can be concluded that without further model updates, the proposed solution has a better generalisation ability.

Table 5.4: Aggregated gain $\mathbb{G}$ with respect to the Random baseline for all considered DRL algorithms in the testing scenario.

Algorithm	**FL-A2C**	**FL-DQN**	**C-MARL**	**Centralised**
Aggregated Gain $\mathbb{G}$	0.245	0.429	0.172	-0.313

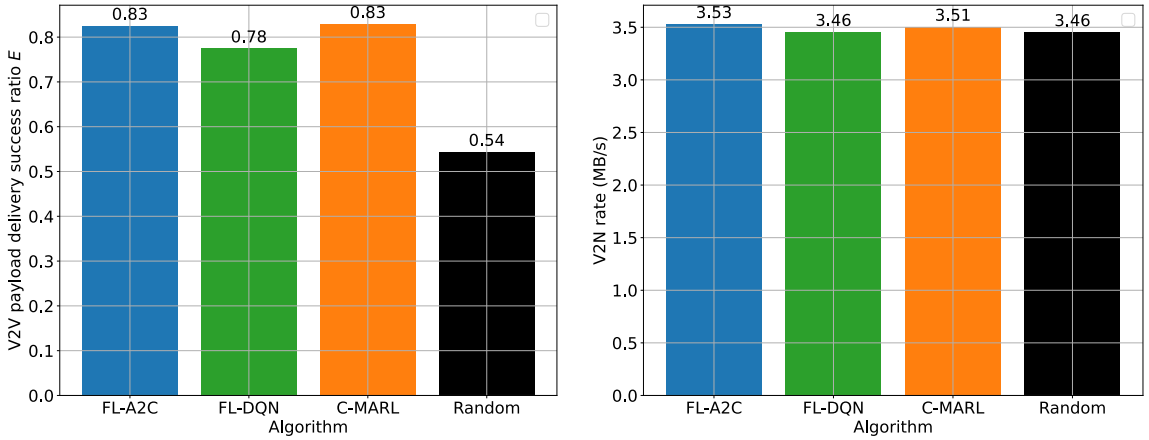
5.6.3 Online Model Update: Results of the Newly Joined Agent

The radio environment in vehicular networks is highly dynamic, with vehicles travelling in and out of a local area frequently. As a result, it is important for resource allocation schemes to adapt to the changing environment which corresponds to the online updating of ANN models for DRL algorithms. Therefore, the testing scenario is extended to emulate the situation with leaving and newly joined VUEs. For the extended testing scenario, it is assumed that one agent with a trained local model has left the local area, with another agent joining the area without a trained model. Hence it is important for the newly joined agent to have a proper ANN model for efficient resource allocation in sidelink Mode 2, which necessitates VUEs for decision-making.

For the proposed federated-DRL, the new agent can download the global model from the BS and further update it with its local data. As for the C-MARL, it is assumed that the new agent manages to get a copy of another agent's trained model upon entering the area to avoid training a new ANN model from scratch. To simulate reality, the episode number for the online ANN update is set to 1 as the same VUE will not travel through the same trajectory multiple times. The trajectories in the testing scenario are used in this extended scenario with all agents except the newly joined agent having their trained local ANN models, and online model update is enabled. Besides, the ϵ-greedy policy for FL-DQN and C-MARL is disabled to avoid random action-taking affecting the performance statistics. Also note that the extended scenario focuses on the Mode 2 resource allocation, and thus the Centralised benchmark is not considered. The performance results for all three performance metrics are presented in the same manner as the above sections.

Figure 5.7 shows the performance results regarding E and Λ. FL-A2C and C-MARL both come to the top with the same $E = 0.83$, with FL-DQN having a slightly lower $E = 0.78$. As for Λ, all three MARL algorithms have very similar performance compared to the Random baseline, with only a little improvement made to this metric (no improvement made by the FL-DQN). As a result, FL-A2C and C-MARL have the same $\mathbb{G}$, while FL-DQN has a 0.11 lower $\mathbb{G}$ in response to its lower E and Λ. In this extended scenario, FL-A2C shows impressive online update training results, in

line with the C-MARL benchmark, which was the best-performing algorithm in the training scenario. Besides, FL-DQN also reaches a much closer performance to C-MARL compared with the results in the training scenario. Although only slight performance gain is achieved by the proposed scheme, this satisfies the performance expectation of this research as discussed in Section 5.1.3. These results support the superiority of the proposed solution, especially regarding the real-world deployment aspects, which will be further elaborated by the communication cost analysis below.



(a) (b)

Figure 5.7: The performance results of different benchmarks in the testing scenario with online updating. (a) System-wise Successful V2V payload delivery ratio. (b) Sum V2N rate system-wise.

Table 5.5: Aggregated gain with respect to the Random baseline for all considered DRL algorithms in the extended testing scenario for online updating.

Algorithm	**FL-A2C**	**FL-DQN**	**C-MARL**
Aggregated Gain $\mathbb{G}$	0.557	0.444	0.557

5.6.4 Discussion on Communication Costs

The online model update requires information sharing to train the ANN models. This will lead to communication costs for the deployed multi-agent DRL algorithms, i.e., FL-A2C, FL-DQN, and C-MARL. Therefore, a communication cost analysis for the above-extended test scenario is presented. Note that this analysis is purely based on the size of data that needs to be transmitted for online model updates.

For the proposed federated-DRL algorithms, the ANN models are transmitted during the online update. On the implemented platform, the ANN model size is 696.0 KB in storage. For the 30-second simulation time, there is only one round of model transmission, i.e., when the newly joined agent first downloads the global model, leading to a system-wide communication cost of 23.2 KB/s for the scenario. This is also true for C-MARL with model transfer as a model copy is downloaded from another agent.

For C-MARL, the throughput information of all agents needs to be shared to enable online model updates, and it is assumed that the corresponding message sent from each agent is 4 bytes, i.e., the reward element consisting of an agent's throughput information is assumed to be represented by a 32-bit floating point number. As the control message needs to be sent every 10 ms according to the simulation configuration in Table 5.1, this will lead to an additional communication cost of 16 KB/s system-wise (as the other four agents need to send their reward element to the newly joined agent). Combined with the 23.2 KB/s for model copying, C-MARL has a communication cost of 39.2 KB/s for the online update scenario. The above analysis is summarised in Table 5.6.

Table 5.6: The communication cost analysis for the online update scenario.

Algorithm	**Proposal**	**C-MARL**
Communication overheads (KB/s)	23.2	39.2
Communication frequency	per 30 s	per 10 ms

From this analysis, the proposed federated-DRL solutions have a 40.8% smaller communication cost compared to C-MARL with model transfer, while the related data is sent at a much smaller frequency, i.e., 30 seconds compared to 10 ms in the considered scenario. Additionally, the proposed solution has a much more convenient setting for ANN model sharing via the FL aggregator designed to be deployed at a BS. In contrast, a newly joined agent must copy another agent's model to avoid training from scratch for C-MARL. This setting of C-MARL is much less stable when applied to a real-world scenario for optimised performance as it cannot guarantee another agent with a trained model to appear near the newly joined agent. Moreover, there is also a potential performance downgrade as the copied model did not have access to other agents' knowledge as those in FL did.

5.6.4.1 Quantisation for Model Size Reduction

Although with a much more convenient global model-sharing architecture, the large model size will cause significant communication overheads when many agents need to update their local models at the same time due to multiple local model uploads and global model downloads. Therefore, effectively compressing the ANN model size while preserving as much performance as possible becomes important for the proposed federated-DRL solution.

The quantisation model compression technique is utilised in this section to achieve this goal for the federated-DRL solution. It is implemented using the Post-training Dynamic Quantisation (PTDQ) functionality provided in the Pytorch library [255] to prove the concept. PTDQ converts all hidden layer weights from the 32-bit floating-point representation to 8-bit integers, while keeping the floating-point representations for biases and activations. FL-DQN is used for this proof of concept and the scenario of the original testing scenario is chosen. The corresponding results are summarised in

Table 5.7. The results indicate that PTDQ reduces the FL-DQN's ANN model size by 73.9% with only a 3% performance loss and a 1% reduction in E compared to the results without PTDQ, suggesting great trade-offs for communication cost reduction for the proposed federated-DRL solution.

Table 5.7: Model size and performance before/after PTDQ for FL-DQN in the testing scenario.

Algorithm	**Model size**	**Successful V2V delivery ratio E**	**Sum V2N rates Λ**	**Aggregated Gain $\mathbb{G}$**
FL-DQN	696.0 KB	0.8	3.28 MB/s	0.429
FL-DQN (PTDQ)	181.1 KB	0.79	3.30 MB/s	0.417

5.6.4.2 Complexity

For computational complexity, the evaluation configuration has set all ANN models to have the same hyperparameters (i.e., number of layers, number of neurons, activation functions and learning rate.) where possible. As a result, the computational complexity from the ANN models are anticipated to be theoretically equal, which was also the motivation for such a configuration.

5.7 Summary

The dynamic radio environment, differentiated QoS requirement, and substantial increase in the numbers of VUEs in vehicular networks have led to the topic of optimised resource allocation for C-V2X, especially for an underlay sidelink configuration. For decentralised resource allocation in Mode 2 setup, existing research focuses on a cooperative MARL setting that has satisfying performance while requiring frequent direct information sharing among agents. There are also difficulties in sharing a trained model with newly joined agents to avoid training from scratch, and thus performance downgrade. FL is an alternative decentralised ML architecture that allows model sharing among participating agents while not requiring direct data transmission, being a promising candidate to solve the above challenge. Additionally, the mixed MARL setting requires much less information to be directly shared among agents to reduce communication overheads. However, it depends on the algorithmic design to match or even exceed the performance of the cooperative setting. In this work, a federated-DRL solution is proposed to jointly allocate spectrum and power in Mode 2, underlay sidelink C-V2X system. With its mixed MARL setting, the only information transmitted during training is the local/global models. The proposed solution is also implemented with two different DRL to explore the effectiveness of different DRL algorithm types.

Through extensive simulation, the proposed solution outperforms the state-of-the-art with respect to the generalisation capability. To be specific, the two implemented algorithms, FL-DQN and FL-A2C, outperform the best-performing C-MARL benchmark in the direct generalisation test by 17% and 6% regarding the metric of V2V payload delivery ratio, and 149% and 42.4% regarding the metric of aggregated gain $\mathbb{G}$, respectively. For the online model update scenario, the proposed FL-A2C can match the performance of the C-MARL benchmark with respect to both the V2V payload successful delivery ratio E and $\mathbb{G}$ whilst FL-DQN has a close performance (5% and 20% inferior for the two metrics compared with C-MARL). For the communication cost analysis, the proposed algorithm has a 40.8% smaller communication overhead compared to C-MARL with model transfer. Furthermore, the proposed solution requires much less frequent data transmission, i.e., 30 seconds compared with the 10 ms required by the C-MARL benchmark. In addition, a proof-of-concept communication cost reduction scheme is discussed via ANN model compression. Through the PTDQ implementation, the ANN model of the proposed solution can be reduced to 1/4 of its original size with only 3% performance loss. To sum up, the proposed federated-DRL solution is a highly promising candidate regarding both performance and communication costs to deploy in future vehicular networks for decentralised resource allocation.

Chapter 6

Adaptive Cell Switching Using Graph Neural Networks

6.1 Introduction

Base stations (BSs) are the major energy consumers in cellular networks and account for 60% to 80% of cellular network's total power consumption [42]. The proliferation of BS deployment for network densification has significantly increased the environmental and economical burden on society. Therefore, it is vitally important to optimise the energy efficiency of BSs for vehicular networks to reduce OpEx and be more environmentally friendly, combined with other energy efficiency schemes introduced earlier in Chapter 3. Green radio is a set of technologies that jointly consider service quality and power consumption to deliver energy-efficient wireless communications. With the development of green radio technologies, various options of energy efficiency schemes have been proposed to reduce BS energy consumption [195]. These BS energy-saving strategies can be classified into four main categories according to their target parameter to minimise in Eq. (3.5), which is summarised in Figure 6.1 with examples. A more detailed background introduction to green radio techniques for BSs is already covered in Section 3.3.

Cell switching is a spatial approach for BS energy efficiency by turning a cell or a BS into sleep mode, which is a promising approach to optimise energy efficiency on the network side. The conventional strategy has been to maintain constant BS operation even when no active users are using the BS's coverage, resulting in significant energy wastage. As data traffic in cellular networks shows both temporal and spatial variation, load adaptive network operation can be executed such that BSs could be switched to sleep modes during periods of low/no traffic to optimise energy efficiency. Additionally, sleep mode can be applied along with other green radio techniques for energy efficiency

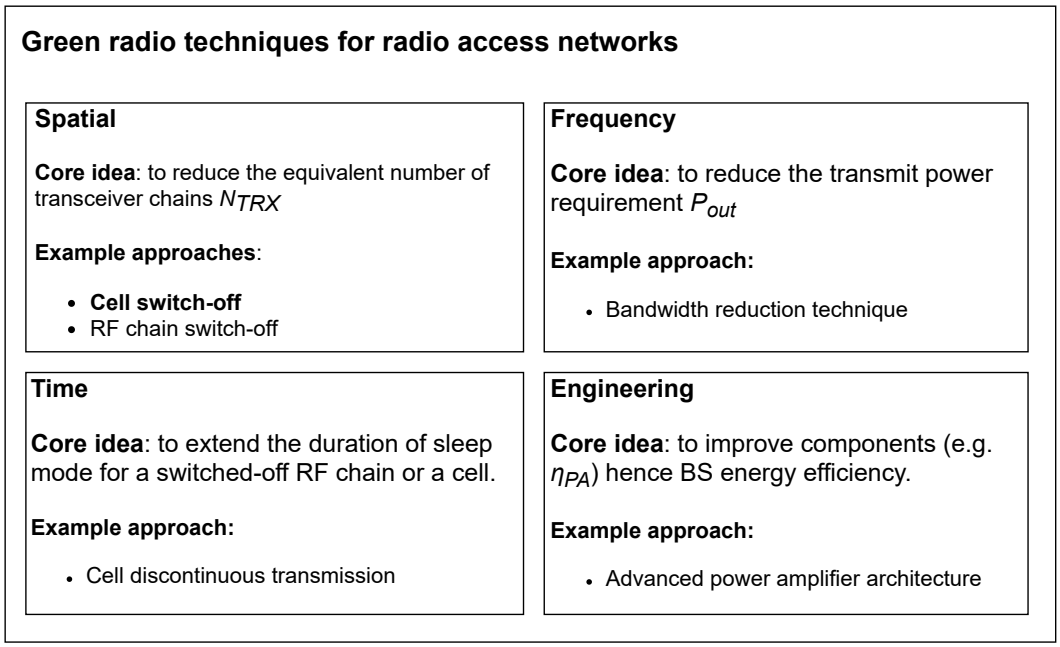


Figure 6.1: A brief summary of green radio techniques for radio access networks, recapitulated from the related details in Chapter 3.

such as engineering higher-efficiency power amplifiers and reducing the transmit power while keeping service QoS via link adaptation, etc.

However, it may not always be feasible to completely switch off BSs in the cellular network architecture due to potential coverage holes that would inevitably downgrade the users' QoS. Moreover, a sleeping BS cannot transmit signals needed by UEs to establish connections such as cell discovery and channel estimation [256]. Separating the control signals requiring full coverage from those supporting high data rate transmissions, the Control Data Separated Architecture (CDSA) is a crucial network architecture to the above challenge [256]. In Control Data Separated Architecture (CDSA), a Macro Cell (MC) control BS provides constant coverage, general data services, and handles signalling tasks, while Small Cell (SC) data BSs provide high data rate services to support various mobile applications. By appointing an MC to ensure the service coverage and the backhaul connection between the MC and SCs, CDSA brings the possibility to switch SCs within the MC's cover into deep sleep mode without impacting the users' QoS during cell switching operation and traffic load re-association. Figure 6.2 demonstrates the considered Heterogeneous Network (HetNet) with CDSA and the concept of cell switching. Figure 6.2a shows such a CDSA HetNet unit consisting of one MC and four SCs while Figure 6.2b presents that after cell switching, one of the SCs is turned into sleep mode and the MC takes over that SC's connected cellular UEs.

To ensure maximum energy efficiency gain (trade-off between service and energy saving) via sleep mode, it is essential for cell-switching solutions to reach optimal decision-making. Research has been conducted for optimised cell switching solutions in CDSA HetNets, and analytical models and heuristic algorithms were developed with *a priori*

knowledge of the environment [257–259]. However, such approaches usually face the NP-hardness solving issue due to the problem formation complexity and computational overhead for complex scenarios and have limited generalisation capability adapting to the dynamic environment of wireless networks [36, 51].

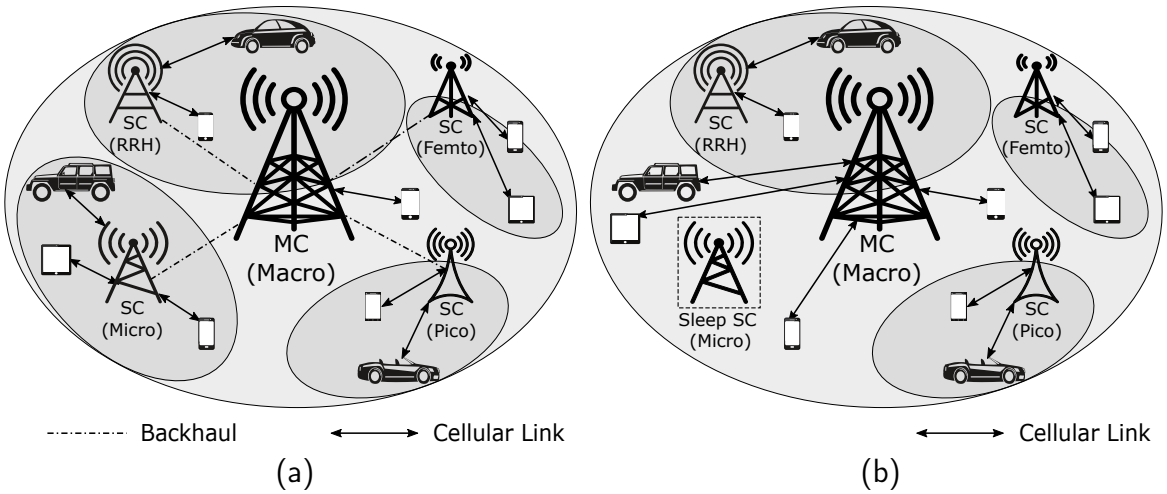


Figure 6.2: A CDSA HetNet unit consisting of an MC and densely deployed SCs within the coverage of the MC. (a) Before cell switching, all users within an SC connect to it. (b) After cell switching, the MC takes over the users of the sleeping SC; the backhaul connections are omitted in this part for simplicity. Note that there are four types of SCs according to [45], RRH: Remote Radio Head.

In comparison, Machine Learning (ML) techniques are able to extract knowledge from historical and real-time collected data for cell switching decision optimisation. Reinforcement Learning (RL) based algorithms can directly optimise cell switching strategies [54, 143–145] while other ML techniques for prediction, classification, and clustering are capable of assisting cell switching solutions for improved performances [146–148]. Furthermore, deep learning techniques utilising the strong approximation capability of ANNs can accommodate highly complex scenarios by directly learning patterns from the rich datasets generated by the communication networks.

In recent years, Graph Neural Network (GNN) has received much attention from the research community, achieving successes in different research areas such as in computer vision, chemistry, and social networks [260, 261]. It has also been applied to wireless network research including traffic prediction, power control, etc. [262]. This is because the topology of communication networks can be naturally modelled using graphs, a data structure comprising nodes and edges with relationship information embedded. A detailed background of graph-structured data and GNN can be found in Section 2.3.3.

Compared to existing deep learning-based solutions, GNNs show an advantage over traditional feed-forward ANNs of better generalisation capability through learning the network topology via the graph data structure with node size invariance [136, 263]. The ability to learn the underlying topology on graph-structured data can increase

dataset utilisation efficiency and learning robustness, while node-size invariance is a significant advantage that reduces computational and time cost for retraining time after deployment to differing scenarios when compared to other deep learning techniques such as Deep Reinforcement Learning (DRL), which needs retraining when the action space size changes. Both advantages make GNN a powerful candidate for cell-switching decision optimisation. However, little research has been conducted exploring how GNNs perform in cell switching problems to date. As a result, a cell-switching solution using a GNN model is proposed for adaptive cell-switching decision-making.

6.1.1 Related Work

Cell-switching decision optimisation must find the best combination of SCs to offload traffic and switch off in order to maximise power saving while maintaining user QoS. Such a problem is naturally combinatorial and may be formulated as mixed integer programming with multiple sets of variables to consider trade-offs among metrics [42] as too aggressive cell switching may lead to user QoS sacrifices when maximising power saving, while too mild cell switching leads to delivering a service capacity exceeding the available users' demands and causes energy wastage. Various approaches exist in the literature to implement cell switching optimisation in CDSA HetNets for energy optimisation. These methods can be broadly classified into heuristic algorithms and ML-based direct cell switching decision-making (mainly RL-based). Some research also developed multi-tier solutions combining heuristic algorithms with ML methods, or developing combinations of different ML methods (e.g. supervised learning and RL).

For heuristic algorithms, the Exhaustive Search (ES) algorithm is guaranteed to produce the most optimal cell switching results by traversing the whole search space to find the best SC combination(s) based on the objectives while satisfying the constraints. However, the complexity of ES grows exponentially and is only practical to apply to small search spaces (small networks for V2X) [54]. To improve the search efficiency towards an optimal solution, a suboptimal greedy SC on/off strategy was proposed in [257] to determine the SC switching patterns for a BS cluster in a green ultra-dense HetNet. This heuristic algorithm tried to maximise the network energy efficiency while considering the traffic load of the SCs and user QoS requirements. Similarly, a firefly algorithm was developed in [258], where joint optimisation of the area spectral efficiency and energy efficiency was formulated to determine the optimal system parameters for a two-tier ultra-dense HetNet. Moreover, a cooperative energy optimisation scheme for 5G ultra-dense HetNet using graph theory was proposed in [259], where a graph representation of the network was first developed, followed by applying graph theory to determine the order of SC nodes to which power-off/on procedures are applied.

Heuristic algorithms are hard-coded with limited generalisation ability, and recurrent applications are often required when network conditions change significantly. As a result,

some research combined ML methods to boost the performance of heuristic algorithms, or to reduce the problem search space and hence the overall problem complexity. An example of such approaches is the work of Abubakar *et al.* [146] that presents a two-tier cell switching based on unsupervised learning and the ES algorithm, where the search space for the ES algorithm is reduced by separating an ultra-dense HetNet into different smaller groups using a clustering algorithm.

To develop adaptive cell switching solutions, another approach is to design ML-based direct cell switching algorithms. For example, a dynamic SC load adjustment algorithm was proposed in [143] that used Q-learning to train an optimal offloading and load-balancing policy to switch off redundant SCs in an ultra-dense HetNet. A distributed Q-learning technique was utilised in [144] that modelled all SCs as learning agents that cooperate to jointly learn the best sleep mode for each SC in a multi-sleep-mode HetNet setup, in order to maximise the network's energy efficiency. However, such tabular RL methods require a large state-action table (or Q-table) to represent the optimal policy when the HetNet scale rises, which leads to considerable memory consumption. As a result, approximation-based RL algorithms become promising candidates, such as the SARSA algorithm with linear function approximation proposed by Ozturk *et al.* [54]. Deployed for online training and execution in an ultra-dense HetNet, the feature space of the SARSA algorithm contains all BSs' traffic loads with the total network-wise power consumption for optimal binary cell switching policies for SCs. Another approach is to exploit the strong approximation capability of ANNs, such as the work of Zhang *et al.* [145] who developed a DDQN-based cell switching algorithm to determine the optimal sleeping strategy in a heterogeneous radio access network. The algorithm was trained and tested using real-world traffic data to minimise the energy consumption of the HetNet while maintaining the user QoS within the network.

Moreover, different learning-based techniques can jointly be utilised for cell switching decision making, such as the work in [148] which combined CNN, RNN and DRL to form a multi-tier solution. CNN and RNN were leveraged to exploit the geographical and semantic spatial-temporal correlations of mobile traffic for future traffic prediction, while the deep deterministic policy gradient algorithm [139] was used to compute optimal cell switching decisions.

Different learning-based solutions have been proposed in the literature for cell switching optimisation, while GNN techniques received relatively little attention although with strong expressive capability and explored to be effective in solving a similar problem of link scheduling [134, 136]. For instance, Lee *et al.* [134] proposed a graph representation design for device-to-device communication and utilised graph embedding combined with neural networks to learn an optimal link scheduling decision without requiring channel state information. Their performance results showed that graph representation

learning is competitive in performance optimality (over 90% of optimal system sum rate), generalisation ability and scalability. However, their design centred around communication links cannot be directly adapted to the cell switching use case, which also motivated this work to explore GNNs on cell switching.

6.1.2 Research Contributions

GNN has the following advantages over other learning-based techniques for the cell switching problem:

- GNN learns on graph-structured data, which include relationship information among modelled BSs, which is absent from other techniques while being useful to the NN model to learn the features with extra information and hence reduce the training epochs.

- GNN is capable of being extended to different-sized networks without the need of retraining, which considerably reduces the cost for deployment to different HetNets compared to other learning-based techniques such as deep Q-learning.

Consequently, this chapter focuses on a first attempt to develop a GNN-based cell-switching solution for CDSA HetNets that can be deployed at each MC of the network to provide cell-switching decisions for SCs within its coverage at a system-level in a locally centralised manner. The proposed solution consists of the graph representation of individual HetNet units, GNN computational model building, and loss function design for unsupervised training. The performance of the proposed solution is evaluated using a dataset based on real-world cellular network traffic information. The performance results are compared to the theoretical optimal results calculated by the exhaustive search algorithm, a state-of-the-art RL-based solution, and the All-on method representing no cell switching deployment. Note that although high-level discussions on how the proposed cell switching algorithm may be deployed in the cellular protocol stack, this work focuses on the algorithmic development and the detailed deployment aspect is beyond the scope of this work. The contributions of this chapter are summarised as follows:

- A graph representation of a CDSA HetNet unit considering BSs' traffic loads and power consumption, and a GNN-based Cell-switching Solution (GBCSS) for CDSA HetNets. GBCSS has a much lower computational complexity during execution compared to the ES algorithm hence is scalable and tractable for large deployments for beyond 5G super connectivity.

- The proposed GBCSS is evaluated using a well-established telecommunication dataset[264] that is based on real-world Call Detail Records (CDR) information

in the city of Milan, making the results more realistic.

- Evaluation results show a 10.41% power efficiency gain using the GBCSS with respect to the baseline without cell switching. Compared to the ES algorithm used for the upper bound baseline, the GBCSS achieves 75.76% of the optimal performance results with less than 0.5% of user QoS sacrificed. In addition, the average energy efficiency of GBCSS outperforms that of the other learning-based benchmark algorithm by 11.90%.

- Generalisation tests for different dates, times, and node sizes show the GBCSS' strong generalisation ability that makes the method highly promising for practical deployments.

6.2 System Model and Problem Formulation

6.2.1 Network Architecture and Power Consumption Model

This work considers an ultra-dense HetNet with a CDSA architecture [256], formulated by multiple HetNet units comprising of one MC and multiple SCs of different types within the coverage of the MC. For each HetNet unit as shown in Figure 6.2, the MC serves as the control BS for signalling, and provides constant coverage and data services, while the SCs only handle data services based on user-specific requests for network capacity enhancement. The MC also acts as a centralised controller within its coverage area for switching SCs in/out of sleep mode and takes over the traffic of sleeping SCs as shown in Figure 6.2b. This task contains traffic load observations on all local SCs, and decision-making on the set of SCs that should be switched into sleep mode during periods of low traffic intensity, with the available capacity of the MC also taken into consideration.

Using the above system model, the instantaneous power consumption of a HetNet unit P_{tot} containing 1 MC and N_{SC} SCs (with SCs' BS types implied) in a CDSA HetNet is given by:

$$P_{tot} = \sum_{i=1}^{N_{SC}+1} P_{BS}^{i} = P_{MC} + \sum_{i=1}^{N_{SC}} P_{SC}^{i} \tag{6.1}$$

where P_{BS}^{i} is the power consumption of the i-th BS (BS_i) in the HetNet unit, BS type implied, while P_{MC} and P_{SC}^{i} denote the power consumption of the MC and the i-th SC respectively.

The calculation of P^i_{BS} for every BS type follows (3.6) according to the EARTH power consumption model [45]. In (3.6), P_{out} is the transmission power that is proportional to the maximum transmission power P_{max} based on a BS's factorised traffic load. In this chapter, P^i_{out} and P^i_{max} are used to denote the transmission power and the maximum transmission power of the i-th BS. The factorised traffic load is expressed as [265]:

$$P^i_{out} = \lambda^i_t P^i_{max} \tag{6.2}$$

where $\lambda^i_t \in [0, 1]$ is BS_i's factorised traffic load at time step t, defined as:

$$\lambda^i_t = \frac{d^i_t}{C_i} \tag{6.3}$$

where d^i_t represents the radio resources of BS_i utilised by its served users at t, and C^i is the radio resource capacity of BS_i.

It is also assumed that BSs of a given type (e.g. all micro BSs) are configured with identical hardware for this general problem formation, such that every type of BSs will have constant P^i_0, P^i_{max}, and Δ^i_p. Therefore, BS_i's power consumption P^i_{BS} depends only on its traffic load λ^i and BS type. For real-world applications, values of these parameters can be specified based on individual BS setups.

6.2.2 Problem Formation

Following the above system model, the goal is to determine the optimal BS switching strategy (i.e. the optimal set of SCs to switch on/off) for each time step t (in minutes) in a given time period $\mathcal{T}$ (in minutes), to minimise energy consumption while maintaining user QoS in a HetNet unit. The switching strategy at t is defined as $\Gamma_t = \{\gamma^1_t, \gamma^2_t, ...\gamma^{N_{SC}+1}_t\}$, where $\gamma^i_t \in \{0, 1\}$ indicates the switching decision for BS_i at t, with 1 denoting ON and 0 representing OFF. In this work, the MC of each HetNet unit is defined to be at index 1 (BS_1) and is considered always ON according to its functionality, i.e. $\gamma^1_t = 1, \forall t \in T$.

By deciding the switching strategy Γ_t at each time step t, a traffic re-association stage is carried out before the cell switching execution, during which the MC takes the traffic loads from, or allocates traffic loads to an SC within its coverage if that SC was switched to sleep mode or brought back in service, i.e. moving d_i from BS_i to BS_1, where $i \neq 1$. However, as SCs and the MC may have different capacities, it is essential to consider such differences for traffic re-association when using the factorised traffic loads λ during this process. To represent such capacity differences, ϕ_i is introduced as the ratio of

BS_i' capacity to that of BS_1 (the MC). Note that ϕ_1 is always 1 as it means the MC's capacity comparing to itself:

$$\phi_i = \frac{C_i}{C_1} \tag{6.4}$$

Therefore, for $BS_i\,(i \neq 1)$, $\phi_i \lambda_t^i = \frac{C_i}{C_1} \times \frac{d_t^i}{C_i} = \frac{d_i}{C_1}$ represents the factorised traffic load of this BS with respect to the capacity of BS_1 (the MC), and the original factorised sum traffic load Λ_t at time step t before executing cell switching Γ_t can then be defined as (6.5). Note that Λ_t is based on the capacity of BS_1 (the MC):

$$\Lambda_t = \lambda_t^1 + \sum_{i=2}^{N_{SC}+1} \phi_i \lambda_t^i = \lambda_{t+1}^1 + \sum_{i=2}^{N_{SC}+1} \frac{d_i}{C_1} \tag{6.5}$$

Moreover, each BS' traffic load after re-association and cell switching can be calculated as follows, starting with BS_1 (the MC):

$$\hat{\lambda}_t^1 = \lambda_t^1 + \sum_{i=2}^{N_{SC}+1} [\gamma_t^i \phi_i \lambda_t^i - (1-\gamma_t^i)\phi_i \lambda_t^i], \quad \text{if } \gamma_t^i \neq \gamma_{t-1}^i \tag{6.6}$$

and for all SCs (for $i \geq 2$):

$$\hat{\lambda}_t^i = \begin{cases} 0 + (1-\gamma_t^i)\lambda_t^i, & \text{if } \gamma_t^i \neq \gamma_{t-1}^i \\ \lambda_t^i, & \text{else} \end{cases} \tag{6.7}$$

where λ_t^i and $\hat{\lambda}_t^i$ are the traffic loads of BS_i at time step t before and after the execution of traffic re-association and cell switching.

Note that after the cell switching execution, BS_i's power consumption P_{BS}^i will also change to $\hat{P}_{BS}^i$ upon λ_t^i changes to $\hat{\lambda}_t^i$. Following Eq. (3.6), $\hat{P}_{BS}^i$ is hence calculated as:

$$\hat{P}_{BS}^i = \gamma_t^i (P_o^i + \Delta_p^i \hat{\lambda}_t^i P_{max}^i) + (1-\gamma_t^i) P_s^i \tag{6.8}$$

The factorised sum traffic load after cell switching $\hat{\Lambda}_t$ of the HetNet unit is then defined as (6.9). It is noteworthy that $\hat{\Lambda}_t \leq \Lambda_t$ as switching off SCs after the MC reaches its

capacity ($\hat{\lambda}_t^1 = 1$) will lead to sacrifices of the original traffic loads:

$$\hat{\Lambda}_t(\Gamma_t) = \hat{\lambda}_t^1 + \sum_{i=2}^{N_{SC}+1} \phi_i \hat{\lambda}_t^i \tag{6.9}$$

Denote $\hat{P}_{tot}$ as the energy consumption of the HetNet unit after executing Γ_t, the optimisation objective is hence to choose an optimal Γ_t for the HetNet unit to maximise the energy efficiency for all $t \in T$, i.e. to minimise $\hat{P}_{tot}$ while maximising $\hat{\Lambda}_t$ (to maintain Λ_t as much as possible and thus preserve the original user QoS) in the HetNet unit. Combining (6.1), (6.8), and (6.9) this optimisation can be formulated as follows, with N_{SC} independent variables ($\gamma_t \in \Gamma_t$) and two constraints:

$$\min_{\Gamma_t} \ \hat{P}_{tot}(\Gamma_t) = \sum_{i=1}^{N_{SC}+1} \hat{P}_{BS}^i = \sum_{i=1}^{N_{SC}+1} [\gamma_t^i (P_o^i + \Delta_p^i \hat{\lambda}_t^i P_{max}^i) + (1 - \gamma_t^i) P_s^i] \tag{6.10a}$$

$$\text{s.t.} \ \max_{\Gamma_t} \ \hat{\Lambda}_t(\Gamma_t) = \hat{\lambda}_t^1 + \sum_{i=2}^{N_{SC}+1} \phi_i \hat{\lambda}_t^i, \tag{6.10b}$$

$$\hat{\Lambda}_t \leq \Lambda_t, \quad 0 \leq \hat{\lambda}_t^i \leq 1. \tag{6.10c}$$

where Eq. (6.10a) defines the optimisation objective to minimise a HetNet unit's power consumption $\hat{P}_{tot}(\Gamma_t)$ given a switching decision Γ_t at time step t. Eq. (6.10b) represents the optimisation constraint which can be regarded as another objective of maximising the system-wide traffic. Eq. (6.10c) defines the optimisation constraints where $\hat{\Lambda}(\Gamma_t)$ is defined by (6.9), which is calculated as the sum of all factorised loads of local BSs with respect to the MC's capacity. $\hat{\lambda}_t^i$ denotes BS_i's load factor after switching, as defined above.

Note that the optimisation constraint $\max_{\Gamma_t} \ \hat{\Lambda}(\Gamma_t)$ has an upper bound of Λ_t which is the original traffic load of the HetNet unit at every time step before executing cell switching as discussed above. Moreover, the value of $\hat{\lambda}_t^i$ should be between 0 and 1 to not exceed a BS's capacity at each time step after switching following the definition of λ. As a result, Eq.(6.10b) is equivalent "to maintain the original traffic load after cell-switching as much as possible" when combined with (6.10c).

Although the formulated cell switching optimisation problem appears to be relatively simple, it is a min-max trade-off problem that needs to consider both the MC's and all SCs' traffic loads within a HetNet unit, while also needing to take the power consumption of different SC types into account (e.g. 4 SC types are considered in the

experiments covered in this work as in Table. 6.1), which is not directly presented in Eq. (6.10a). Therefore, the search space for an optimal cell switching decision is much larger with a highly complex underlying scenario, especially for a large number of SCs. As Γ_t is a discrete set of binary values, the defined min-max optimisation is naturally combinatorial, with $2^{N_{Sc}}$ possible combinations for every time step t for a given HetNet unit.

Such combinatorial optimisation can be considered as a variation of the Knapsack problem, which is a well-known NP-hard [54, 265, 266]. The Knapsack problem considers a set of N_{obj} indivisible objects with integer labels $id = 1, 2, ..., N_{obj}$. Each object is associated with a real number value v_i and a positive real number weight w_i. The goal of the problem is to select a subset of these objects to achieve a maximum sum value while maintaining the total weight within W units, and the mathematical formulation of the problem is expressed as "find o_i, such that":

$$\max \sum_{id=1}^{N_{obj}} o_{id} \cdot v_{id}, \tag{6.11a}$$

$$\text{s.t.} \sum_{id=1}^{N_{obj}} o_{id} \cdot w_{id} \leq W, \tag{6.11b}$$

$$o_{id} \in \{0, 1\}, \quad id = 1, 2, ..., N_{obj}. \tag{6.11c}$$

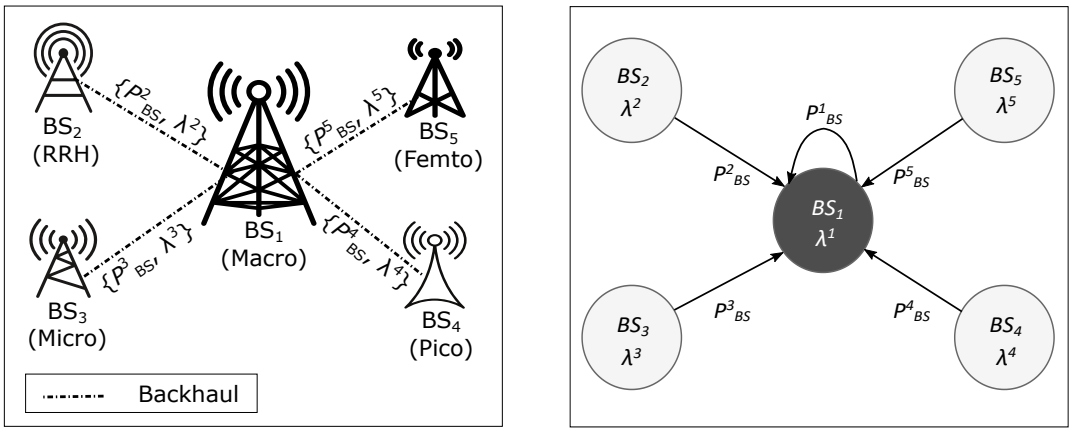
where the objective function (6.11a) can be directly related to the cell switching objective (6.10a) after transforming the minimisation in (6.10a) to a maximisation form by treating BSs' power consumptions as negative values, with the binary Knapsack decision o_{id} representing γ_i. As for the constraint, the $\hat{\Lambda}_t \leq \Lambda_t$ part of (6.10c) represents the Knapsack constraint in (6.11c), while an additional maximisation is added in the formulated cell switching problem, making the cell switching problem overall a variation of the Knapsack problem.

6.3 Cell Switching via Graph Representation Learning

GNN models learn on data represented by the graph data structure. Formally, a graph $g = (\mathcal{V}_g, \mathcal{E}_g, \mathcal{X}_g, \mathcal{A}_g)$ is composed of a set of vertexes/nodes $\mathcal{V}_g$ and a set of edges/arcs $\mathcal{E}_g$ connecting pairs of nodes [117]. When the node pairs in g are unordered, g is referred to as an undirected graph, while ordered node pairs in g make it a directed graph. To enrich the graph g, additional node and edge information can be included in $\mathcal{X}_g$ and $\mathcal{A}_g$ respectively. Each node $v \in \mathcal{V}_g$ is associated with a node feature $x_v \in \mathcal{X}_g$, while an edge $(\overline{u, v})$ connecting a pair of nodes u, v $(u \neq v)$ holds an edge attribute $a_{uv} \in \mathcal{A}_g$.

A graph is a powerful data structure to model a set of objects (as nodes) and their relationships (as edges).

Following the general GNN design pipeline [267], this section presents the GBCSS from the following aspects: graph representation design, GNN computational model building, and learning task confirmation with loss function design.



(a) (b)

Figure 6.3: The proposed graph representation process. (a) System model. (b) Graph representation model. Note that superscripts are for indexing purposes to match the notations in problem formation.

6.3.1 Graph Representation of a HetNet Unit

Omitting the wireless communication links, the network architecture described in Figure 6.2 can be expressed as the system model shown in Figure 6.3a, where SCs within the MC's coverage are connected to the MC through the backhaul. It is assumed that backhaul connections are achieved by high-speed optical fibre whose capacity satisfies all necessary data transmission and the backhaul's own energy consumption is omitted in the considered optimisation problem. For cell-switching, each BS (SCs and the MC) contains the essential information regarding their current traffic load λ and instantaneous power consumption P_{BS}, which is sent to the HetNet unit's local controller located at the MC. Additionally, the type of each SC (e.g. micro or pico) should also be known by the local controller. For each time step, the local controller decides the set of SCs to offload and switch off according to the received information, and then the MC sends the corresponding control signals to the SCs.

The proposed graphical modelling of the system model in Figure 6.3a starts with treating each BS in the HetNet unit as a node, with the BS's traffic load λ modelled as the node feature x_v, while the power consumption P_{BS} of each BS is treated as the edge attributes $a_{1,v}$ that connects the BS at node v and the MC at node 1. Figure 6.3b demonstrates the proposed graph representation model. The neighbour design of this model is based on the system-level assumption that an SC sends its current

load and power consumption data via the backhaul to the local controller deployed at the MC for cell-switching decision-making. Note that the traffic loads and power consumption of the MC should also be sent to the local controller, hence another edge is added to the proposed graph design connecting the MC node to itself. Additionally, edges in this graph representation model are designed to be directed from each SC node to the MC node following the above information flow. For example, the edge connecting node BS_3 and node BS_1 has the direction of $BS_3 \rightarrow BS_1$, and this directed edge hence is denoted as $(\overrightarrow{BS_3, BS_1})$. Directionality reflects that different BSs have differing power consumptions based on the BS types and traffic loads. If an undirected graph representation is used, an edge feature is then shared by its connected node pair, which is not suitable to represent this differentiated power consumption and thus the relationship between an SC and the MC. Therefore, the directionality also allows distinct edge weights to be utilised by the graph convolutional operator introduced in the following section. The node and edge sizes of the proposed graph representation model are both identical to the total number of BSs within a HetNet unit (e.g. tens to hundreds) and thus denoted as n.

This graph representation should be considered as a dynamic graph; after cell switching, the state of all node and edge features change to $\hat{x}_v$ and $\hat{a}_{u,v}$, following the change of λ to $\hat{\lambda}$ and the resultant P_{BS} for all BSs calculated by Eq. (6.6), Eq. (6.7), and Eq. (3.6). It should be noted that other graph representation designs may have differentiated learning outcomes combined with different GNN models. However, investigating the performance of different modelling designs is beyond the scope of this work.

6.3.2 GNN Computational Model for Cell-switching

The graph g serves as the underlying topology for a GNN backbone and is taken as the GNN's input. The GNN then learns and produces a state embedding for each node in g, containing the node's own information and its neighbourhood. Specifically, the GNN processes the set of node features $\mathcal{X}_g$ through a sequence of L hidden ANN layers. At each layer $l \in \{1, ..., L\}$, the feature vector x_v of each node $v \in \mathcal{V}_g$ is updated as:

$$x_v^l = \mu_l \langle x_v^{l-1}, \{x_u^{l-1}, e_{u,v}\} \rangle \tag{6.12}$$

where $\mu_l\langle\cdot\rangle$ is a parametric combination function (operator) with learnable parameters that are updated by the objective (loss) function's gradients through the ANNs' back-propagation. The variable $u \in \mathcal{V}_g, u \neq v$ is a neighbouring node of v within g, such that u and v are connected by edge $(\overrightarrow{u, v})$, and $e_{u,v} \in \mathcal{A}_g$ is the attribute of edge $(\overrightarrow{u, v})$. When $l = 1$, $x_v^{l-1} = x_v^0$, which denotes the original node features $\mathcal{X}_g$. After all L layers, the resulting output feature x_v^L is the node embedding of the original graph g. This work

utilises the local extremum operator (LEConv) proposed in [268] for μ_l, which finds the importance of nodes with respect to their neighbours using the difference operator, and thus benefits from the distinct edge weight of directed edges. The combining function in Eq. (6.12) for LEConv is expressed as:

$$x_v^l = \psi_l\langle\Theta_1^{l-1}x_v^{l-1} + \sum_{(\overrightarrow{u,v})\in\mathcal{E}_g} e_{u,v}(\Theta_2^{l-1}x_v^{l-1} - \Theta_3^{l-1}x_u^{l-1})\rangle \tag{6.13}$$

where $\psi_l\langle\cdot\rangle$ represent the activation function of layer l, which is a configurable hyper-parameter providing nonlinearity, while Θ_i^l ($i \in \{1, 2, 3\}$) denote different learnable parameters.

The main objective of cell-switching is to find the optimal strategy Γ_t at every time step to determine the best set of SCs to switch on or off to increase energy efficiency. Therefore, the node features after node embedding will be passed through a final output layer with another parametric function that maps x_v^L to binary values $\gamma_v \in \{0, 1\}$, while this function needs to be continuous to calculate gradients for GNN's backpropagation. The solution is to first have a function $\Psi\langle\cdot\rangle$ that maps x_v^L to the continuous values ranging between [0, 1] to provide the final output of the GNN, followed by another function mapping such continuous GNN output values to binary ones. In practice, $\Psi\langle\cdot\rangle$ can be implemented using another NN layer whose activation function has an output range of [0, 1], and hence is another configurable hyper-parameter of the computation model for the GBCSS. The value discretization can be achieved by the indicator function $I_{[0.5,1]}\langle\cdot\rangle$ that near-evenly maps the continuous values from [0, 1] to binary values $\{0, 1\}$.

6.3.3 Complexity

As the problem given in (6.10a) is an NP-hard problem, it does not have a deterministic polynomial-time solution. However, since it is a combinatorial optimisation, its optimal solution can be found with an exhaustive search algorithm that iterates through every possible option in the search space. Therefore, it is computationally demanding, and since in the cell switching problem, every SC has two possible states (ON and OFF), the total number of state combinations is $2^{N_{SC}}$ which is the steps required for the exhaustive search to find the optimal ON/OFF switching combination.

In contrast, the presented graph representation modelling and GNN computation model aim to reduce the overall computational complexity. With both graph representation and GNN computation model introduced in previous sections, the forward inference of the GBCSS procedure is summarised in Algorithm 6.1, which is a high-level abstraction of the actual implementation using previously introduced notations to mainly help analyse the algorithm's complexity.

Algorithm 6.1: Feed-forward inference for the proposed GBCSS at time step t

initialize : GNN of $l \in \{1, ..., L\}$ graph convolutional layers and 1 linear layer with weight W_l and bias b_l

input : $P_{tot} = \{P_{BS}^1, P_{BS}^2, ..., P_{BS}^{N_{SC}+1}\}$, $\Lambda_t = \{\lambda_t^1, \lambda_t^2, ..., \lambda_t^{N_{SC}+1}\}$, N_{SC}

output : $\Gamma_t = \{\gamma_t^1, \gamma_t^2, ...\gamma_t^{N_{SC}+1}\}$

1 $(\mathcal{V}_g, \mathcal{E}_g, \mathcal{X}_g, \mathcal{A}_g) =$ `CreateGraphData(`P_{HetNet}, Λ_{HetNet}, N_{SC}`)` ;

2 **for** $l \leftarrow 1$ **to** L **do**

3 **for** $v \in \mathcal{V}_g$ **do**

 `// for every node feature` $x_v \in \mathcal{X}_g$ `do graph convolution as (6.13)`

4 $x_v^l \leftarrow \psi_l\langle\Theta_1^{l-1}x_v^{l-1} + \sum_{(\overrightarrow{u,v})\in\mathcal{E}_g} e_{u,v}(\Theta_2^{l-1}x_v^{l-1} - \Theta_3^{l-1}x_u^{l-1})\rangle$;

5 **end**

6 **end**

7 $\mathcal{Y}_g \leftarrow \Psi\langle W_l^T\mathcal{X}_g + b_l\rangle$ `//` W_l^T `here stands for the transpose of a` W_l

8 $\Gamma_t \leftarrow I_{[0.5,1]}\langle\mathcal{Y}_g\rangle$

Step 1 of the algorithm denotes the graph data creation according to the graph representation design, which can be treated as a linear function that maps the input P_tot and Λ_t to graph-structured data, hence its overall operation steps grow linearly to N_{SC}. Steps 2 to 8 mimic the computational flow of the GNN computation model and step 9 represents the value discretisation introduced in the above section. Step 2 to 7 represents the graph embedding using the LEConv convolution operator and has in total $L \times (N_{SC}+1)$ operations. Step 8 indicates the linear output layer which essentially performs a linear transformation before passing to the activation function $\Psi\langle\cdot\rangle$, while step 9 simply passes the output of step 8 to the indicator function $I_{[0.5,1]}\langle\cdot\rangle$ to produce binary output Γ_t. Both activation functions apply to the input element-wise so their total number operations grow linearly to N_{SC}.

Define $\mathbf{N} = N_{SC} + 1$ being the total number of BSs in a HetNet unit and thus the number of nodes in the graph representation. The complexity of the exhaustive search is then $O(2^{\mathbf{N}-1})$ according to the above discussion. In comparison, most operations occur during steps 2 to 7 for the GBCSS which is $L \times n$, with other operations being linear. Moreover, the number of neurons for all ANN layers will also impact the overall number of operations. However, L will be a constant for a defined GNN model, and each ANN layer's number of neurons will also be a constant upon definition. Therefore, GBCSS' complexity is bound to $O(\mathbf{N})$, which is linear to the total number of BSs in a HetNet unit as introduced in the graph representation. Therefore, this complexity will not lead to a large computational burden compared to the exhaustive search with $O(2^{\mathbf{N}})$ that grows exponentially with respect to the total BS number.

6.3.4 Training and Loss Function Design

The parameters of the GNN computation model can be trained in either supervised or unsupervised learning manner [136]. For the supervised learning approach, it is essential to obtain high-quality labelled samples indicating the optimal cell-switching decisions for each input graph g. However, ES that always generates the optimal solutions has the complexity of $O(2^{\mathbf{N}})$, and hence it is impractical to generate a dataset with sufficient optimal cell-switching samples as the node size increases (e.g. above approximately 20 nodes). In contrast, other methods cannot always guarantee to produce optimal cell-switching decisions for labelled samples, which may hinder the overall learning performances.

Therefore, this research proposed an unsupervised learning approach to train the proposed GBCSS. Assuming a batch of $Batch$ unlabelled samples of a HetNet unit's graph representation g. The designed loss function L is given by

$$L = -\frac{1}{Batch} \sum_{i=1}^{Batch} \zeta_{j,g} \tag{6.14}$$

where $\zeta_{i,g}(\cdot)$ is the objective function for the j-th sample of graph g in the data batch. L aims to directly tune the GNN model to optimise the objective functions in Eq. (6.10a) and Eq. (6.10c). Derived from the calculation of $\frac{P_{tot}(\Gamma)}{\Lambda(\Gamma)}$, $\zeta_{j,g}$ indicates the system-wise power consumption per unit traffic load for the graph representation after cell-switching, following the cell-switching decisions from the GNN outputs. The calculation of $\zeta_{i,g}$ is given by:

$$\zeta_{i,g}(\Psi\langle x_v^L\rangle) = \sum_{v \in \mathcal{V}_g} \frac{\hat{x}_v}{\hat{a}_{v,1}} \tag{6.15}$$

where $\zeta_{j,g}(\Psi\langle x_v^L\rangle)$ denotes the loss $\zeta_{j,g}$ following the cell switching decision represented by the GNN output $\Psi\langle x_v^L\rangle$; $\hat{x}_v$ and $\hat{a}_{v,1}$ are the node and edge features after cell-switching, following the calculation of P_{tot}, $\hat{\lambda}$ and Λ, as described in the problem formation and graph representation. Note that $\hat{a}_{v,1}$ is used instead of a general notation $\hat{a}_{v,u}$ since all edges are defined to connect an SC node to the MC node at index 1, according to the proposed graph representation. The system requires that the MC should always be switched on is also learned by the GNN, as the magnitude of L will become very large when the output label of the MC node is OFF, due to a substantial decrease of Λ.

In this section, a graph representation model and a cell switching scheme were proposed. The design focuses on the BS side (the power consumption and traffic load), which

is independent of the actual communication types from the UE side. As a result, the proposed GBCSS can be applied to both V2N, general cellular networks, or a mixture of communication scenarios as long as cellular BSs are being considered sleep model decision-making and the ANN models being trained with corresponding datasets.

6.4 Evaluation Configurations

The experiments use the EARTH power consumption model [45] and compare the performance of GBCSS with other cell switching benchmarks under various metrics. The power consumption characteristics for each type of BSs are summarised in Table 6.1. For a real-world CDSA HetNet cell-switching scenario, it is natural to consider a set of BSs at fixed geographic locations that experience traffic variances at different time steps of a day and across different days, which is an essential assumption for the experimental configurations in this work.

For the deployment of GBCSS, it is assumed that the algorithm is implemented at the local controller located at the MC for every HetNet unit in a locally centralised manner, along with all other benchmarking algorithms. At each time step t, all SCs in operation send their factorised traffic load and power consumption measurement to the MC via the backhaul for cell switching measurement, while that of the MC will be directly available at the controller due to where it is deployed. For sleeping SCs, the traffic load will naturally be 0, and the power consumption will be the sleeping power for their corresponding BS types, which is known at the MC upon initial deployment. The "measurement" from sleeping SCs can be filled by the MC after receiving all operating SCs' measurements. Note that this work focuses on algorithmic design and evaluation, and thus more detailed real-world deployment setup is beyond the scope of this work. An ultra-dense HetHet may comprise many HetNet units, each consisting of one MC and various numbers of SCs, therefore the obtained results may also be utilised to infer other HetNet units' performances pattern in the network.

The experiments have been implemented via Python 3.9 using scientific and data analysis libraries Numpy [252], Scipy [269], and Pandas [270], with related result visualisations generated via Matplotlib [253].

Table 6.1: Power profiles for each type of BSs according to [45].

BS Type	**Power consumption (W)**			Δ_p
	Operational P_o	**Transmit (max)** P_{max}	**Sleep** P_s	
Macro	130	20	75	4.7
RRH	84	20	56	2.8
Micro	56	6.3	39	2.6
Pico	6.8	0.13	4.3	4.0
Femto	4.8	0.05	2.9	8.0

6.4.1 Dataset and Experimental Setups

6.4.1.1 The original dataset

When calculating power consumption using Eq. (3.6), it is important to obtain the traffic load λ for every BS, and it is also important to evaluate BS-based algorithms using standardised datasets and/or simulation environments [36]. Both aspects considered, an established multi-source dataset [264] is chosen for the performance evaluation, in which the city of Milan is divided into 10,000 square-shaped grids of 235 m $\times$ 235 m. The grid indices are calculated as $ID_{gird} = (x + 1) + 100 \cdot y$, where $x, y \in [0, 99]$ are for indexing purposes only, which is shown in Figure 6.4.

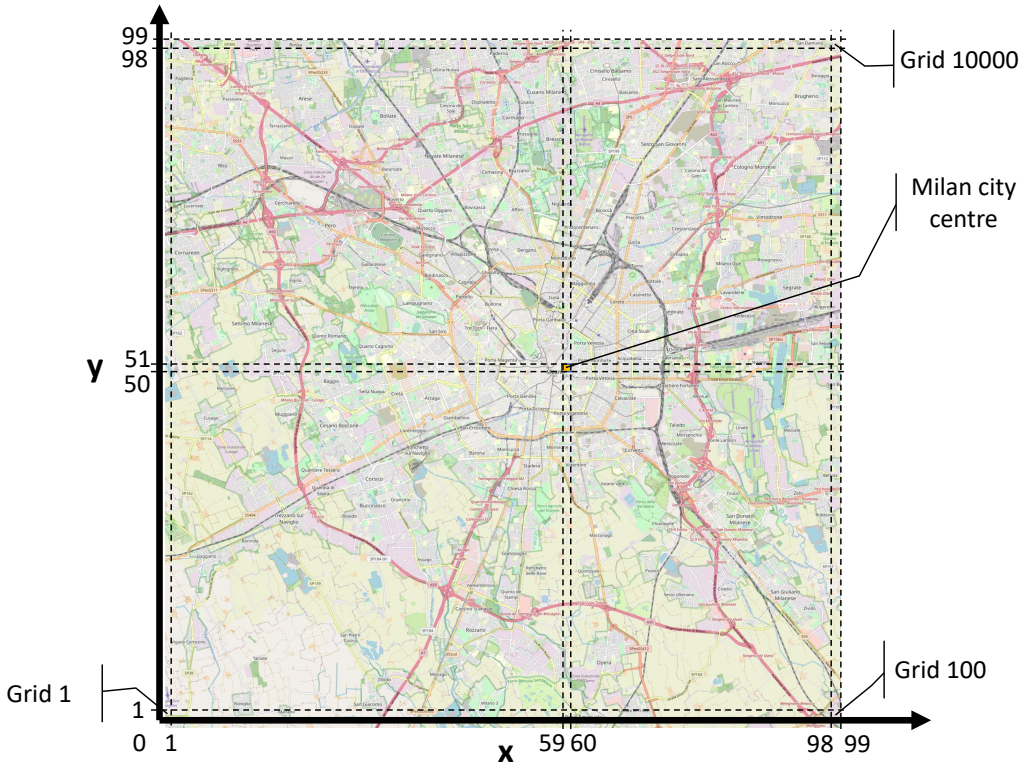


Figure 6.4: The grid representation of the Milan dataset.

In particular, the telecommunication dataset of Milan city based on real-world CDR data provided by Telecom Italia is used for the evaluation experiments [271]. The dataset contains phone calls, text messages, and Internet activities between a user and a BS, which are spatially aggregated into each grid according to the spatial intersections among the grid and nearby BSs' coverage. Additionally, the CDR data was recorded in a 10-minute resolution for a two-month period from November 1st, 2013 to January 1st, 2014. Therefore, the original dataset contains 8 types of features: the grid ID, country code, timestamp (representing date and time), in/out Short Message Service (SMS) activities, in/out call activities, and Internet activities. In total, the dataset contains 62 days' data with 144 time steps per day for 10,000 grids, resulting in 8928 entries of $\{$*Grid ID, datetime, countrycode, in-SMS, out-SMS, in-call, out-call, Internet*$\}$. The names in each data entry stand for the grid index, time interval, phone country code, received SMS, sent SMS, incoming call, outgoing call, and Internet traffic of users.

Although the dataset consists of unitless values (due to undisclosed reasons in the original publication while potentially commercial confidentiality and user privacy) for each type of activity, while no information is provided to reverse the spatial aggregation, these activity levels represent the volume of user-network interaction at each time step and can hence be utilised to calculate and compare traffic loads between grids. Note that since the thesis considers vehicular networks, a dataset that comprises V2N traffic should be considered to better evaluated the performance with vehicular network features. However, due to pragmatic reasons, the chosen dataset may be the most suitable open-source dataset that is also created out of real-world data. Moreover, since the original dataset is composed of various CDR data, it can also be assumed that the original dataset contains traffic generated by passengers using a cell phone on a car or bus, which is similar to V2N regarding aspects such as service types and user mobility. Overall, the insights generated by this evaluation can also be applied to V2N scenarios.

6.4.1.2 Dataset pre-processing and scenario setups

To evaluate the scalability of GBCSS, the experiments consider a scenario of a HetNet unit located in the city centre area, with different numbers of SCs $N_{sc} \in \{4, 8, 12, ..., 32\}$ with BS types assigned uniformly. In the data pre-processing phase of the evaluation process, CDR values of all activity types are first combined into the sum CDR activity data for each grid in the Milan dataset as the cell switching problem considers BSs' overall traffic loads. This operation fuses the original feature set {*in-SMS, out-SMS, in-call, out-call, Internet*} to a new feature type *sum-load* for each time step per grid, which is demonstrated in Figure 6.5. Note that two assumptions have been made to support this pre-processing:

- Unavailable data (marked as "nan"s) is assumed to have a value zero (suggesting not related traffic activity) in order to combine all types of traffic.

- As the original dataset only provides unitless values, all 4 types of data traffic are assumed to have the same unit and the summation of these traffics is treated as the traffic load on a BS.

After the above steps comes a grid selection phase. The 10,000 grids in the Milan dataset cover both urban and suburban areas containing markedly different behavioural signatures regarding traffic variation trends, as shown in Figure 7 in the original work [264] of the Milan dataset. Therefore, choosing grids within the full grid list may violate the above scenario assumptions. Therefore, this work focuses on grids around the city centre of Milan (Grid 5060 with $x = 59$ and $y = 50$, representing the area around the *Duomo di Milano* cathedral, and prominent city square region) as defined in the original article [264]. Grids of $x \pm 4$, $y \pm 4$ from Grid 5060 have been chosen, forming an area

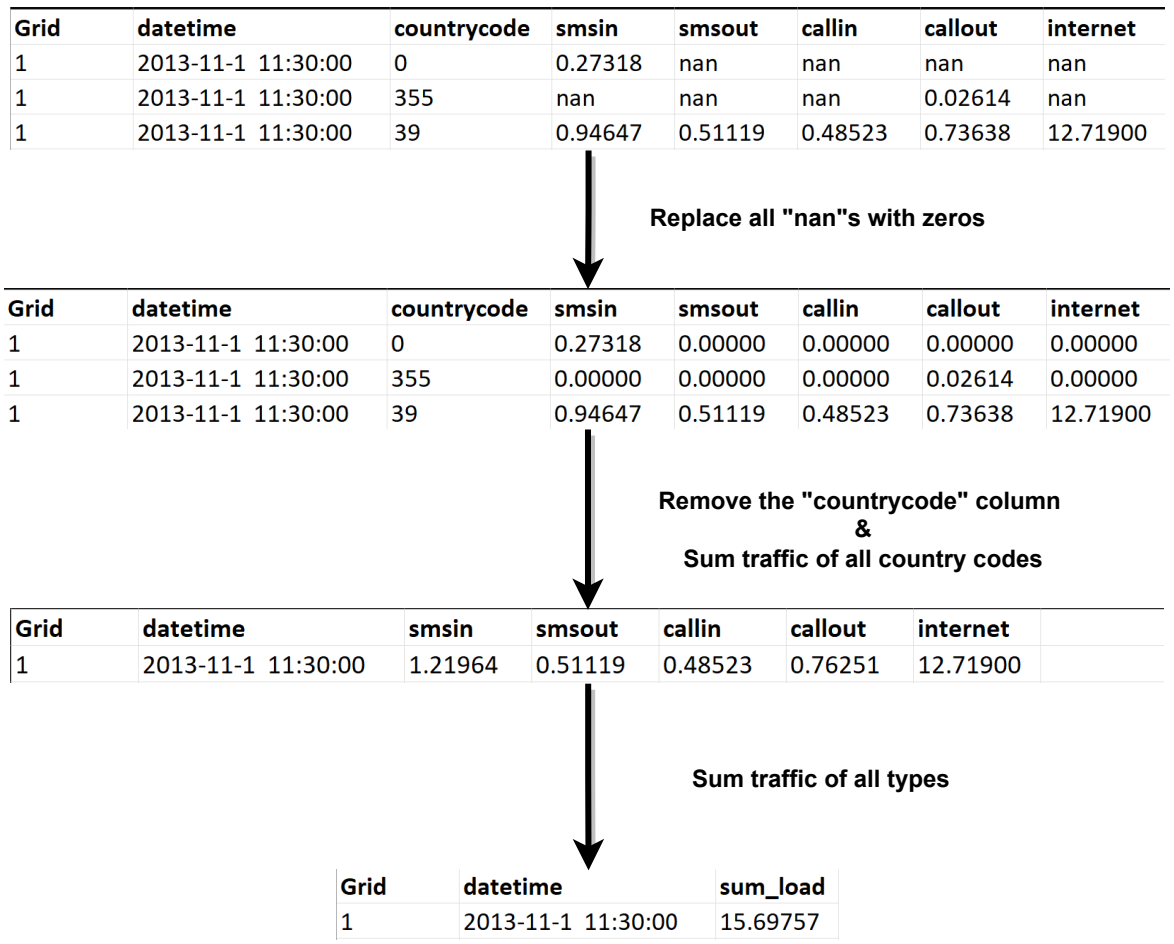


Figure 6.5: The pre-processing workflow of Milan Telecommunications Dataset for performance evaluation. The top is part of the original dataset while the bottom is the pre-processed version of it used in the evaluation. nan: Not A Number, suggesting data unavailable in the original dataset.

of 2115 m × 2115 m that contains 81 grids within the same geographic regions of the Milan city centre. This also suggests that SC offloading is feasible in this area within the coverage of the MC, which is a cornerstone for cell-switching operations.

By non-repetitive random selections, 12 dates were chosen from the business days within the two-month period in the Milan dataset for all N_{sc} cases, and data for 8 of these days were used to form a training dataset [272], while the other 4 days' data were utilised to form a validation dataset [272]. Additionally, two dates have been initially preserved before the above random selections, forming a test dataset [272] that is not used in the training process. Consisting of data from one workday (Nov. 15^{th}, 2013) and one holiday (Jan. 1^{th}, 2014), the test dataset is used for one-day performance evaluation emulating online execution after the algorithm's deployment to test the proposed GBCSS' generalisation ability. As a result, for the processed dataset after grid assignment, the feature set at the time step t is $\{\lambda_t^1, \lambda_t^2, ..., \lambda_t^{N_SC+1}\}$ of size $N_{SC}+1$. The training set contains 1152 entries while the validation and test sets consist of 576 and 288 entries, respectively.

As for the grid assignment, the MC was always assigned with Grid 5060's activities in each N_{sc} case, while one grid for each SC was then selected non-repetitively within the defined region. Only one round of grid selections was carried out for every N_{sc} case (i.e. data samples in the generated datasets were from the same set of grids for each $N_{sc} \in \{4, 8, 12, ...32\}$ case). A fixed random seed is used for all N_{sc} cases to provide consistent and reproducible results. After grid assignment, a BS will experience the traffic variation following that of the assigned grid when no cell switching is applied. For readers interested in the temporal and geographical aspects of traffic variations that lead to the above pre-processing design, Figure 5 and 7 in the original work [264] has provided the related information over a week's time period.

Finally, a sum traffic normalisation phase is executed to produce factorised values that represent λ of BSs. The CDR normalisation scale is determined by ϕ for each type of BSs, after which the *sum_load* feature of a grid becomes λ_n^t, $1 \leq n \leq N_{Sc}+1$ while preserving the traffic variation pattern. This calculation is expressed as follows:

$$\lambda_n^t = \frac{load_n^t}{\max(\bar{load})} \cdot \phi_n \tag{6.16}$$

where t and n denote the time step and BS identifier (BS_n), respectively. $\bar{load}$ represents the set of *sum_load* values of all selected grids while $load_n^t$ stands for BS_n's traffic at time t, and ϕ_n is the relative capacity between BS_n and BS_1 (the MC) as defined in (6.4).

With all experiment setups introduced, the corresponding experimental parameters are summarised in Table 6.2. All BSs are assumed to have the same maximum capacity in terms of radio resources (bandwidth and resource blocks) to simplify the traffic load normalisation during the data pre-processing, and the calculations in Eq. (6.10c) and Eq. (6.19). This is due to the focus of this research being whether the original traffic load is preserved for each cell-switching scheme according to the introduced performance metrics, following the optimisation constraint defined by Eq. (6.10c). Moreover, setting different capacities for each BS type only influences ϕ and thus some numerical results for Λ and P_{tot} after offloading, while such differences do not influence how a cell switching strategy is formulated.

Table 6.2: Experimental configurations.

Parameters		**Values**
Number of time steps per day		144
Number of grids considered for each BS		1
Number of days	Training set	8
	Validation set	4
	Test set	2 (1 workday & 1 holiday)
Bandwidth; number of resource blocks for BSs		20 MHz, 100

6.4.2 GNN Setups

For the experiments, the dataset goes through the graph representation process and the normalised load factors become the node features (X_g) and the calculated power consumption for all nodes becomes the edge features $\mathcal{A}_\}$.

Some configurable hyper-parameters are mentioned. For the evaluation, $L = 3$ hidden layers for node embedding in the GNN computation model are configured, with the neuron size or 128, 128 and 64. The activation function $\mu\langle\cdot\rangle$ is set to the Rectified Linear Unit (ReLU) [273] for all 3 hidden layers. For the output layer setup and binary value translation, $\Psi\langle\cdot\rangle$ is configured as:

$$\Psi\langle x_v^L\rangle = \sigma\langle W^T x_v^L + b\rangle \tag{6.17}$$

where W_l and b_l represent learnable parameters (weights and biases) of a linear transformation, T denotes the matrix transpose, and $\sigma\langle\cdot\rangle$ is the sigmoid activation function [274]. This makes the GNN output continuous values between [0, 1], which can then be used to provide binary via $\gamma_v = I_{[0.5,1]}\langle\Psi\langle W_l^T x_v^L + b_l\rangle\rangle$ as previously discussed.

For other GNN configurations, the batch size is set to 64, and each GNN model (one for each N_{sc} instance) is trained for a maximum epoch of 200 in the experiments.

The learning rate (LR) is initially set to 10^{-3}, with a dynamic LR scheduler [275] configured which reduces the LR by a factor of 2 if no improvement has been made to the loss defined in Eq. (6.15) for the past 10 epochs. The optimisation algorithm is set to the "Adam with decoupled weight decay" (AdamW) optimiser [228]. The above GNN configurations are summarised in Table 6.3. The GNN model and other deep learning-related implementations are fulfilled by Pytorch [204] and Pytorch Geometric [276].

Table 6.3: GNN configurations

Hyperparameters	**Values**
Number of hidden layers; Neuron size	3; 128 × 128 × 64
Hidden layer activation function	ReLU [273]
Output layer activation function	Sigmoid [274]
optimiser	AdamW [228]
Learning rate (LR)	10^{-3}
LR scheduler	Reduce LR on Plateau [275]
Batch size	64
Maximum number of epochs	200

6.4.3 Benchmarks

Benchmarks are necessary to compare the performance of the proposed GBCSS, and the comparison basis was selected following this rationale: 1. The optimal solution (where applicable) maximising energy saving while preserving all original traffic, which stands as the performance upper bound. 2. The bottom line of the performance benchmark is without any cell switching strategy, such that all BSs' traffic and thus user QoS are preserved while sacrificing the energy efficiency optimisation. 3. Another sub-optimal cell switching solution whose performance can be directly compared with GBCSS in terms of the performance metrics defined in the next section. As a result, three different methods are used for benchmarking, introduced as follows:

- **Exhaustive search (ES)**: This method iterates through all possible combinations of binary switching options consisting of the on/off states for all SCs. It also considers the available radio resources at the MC for offloading such that the maximum traffic demand that the network can serve is not exceeded during power consumption optimisation. Therefore, this method checks all possible SC combinations to switch off, and guarantees to produce the optimal cell-switching policy that minimises the total power consumption of the network while preserving the user QoS in the network.

- **Linear function approximation-based SARSA (FA)**: This is a state-of-the-art RL-based cell-switching scheme proposed by Ozturk *et al.* [54]. FA defines every time step t as an episode, and uses a feature vector $\{P_{tot}, \lambda_t^1\, \lambda_t^2, ..., \lambda_t^{N_{SC}+1}\}$ containing all BSs' load factors and system-wise power consumption to train

a parameter set θ that represents the optimal cell switching policy via linear function approximation. For interested readers, more detailed design of the FA algorithm can be found in the original work.

- **All-on**: This approach implements a scheme with no off-loading and cell-switching, and hence all BSs are always left ON. This method ensures the user QoS within a HetNet unit, but no energy saving can be achieved since no SCs will be switched off. It is used as the baseline of optimal throughput with respect to the power consumption bottom line.

6.4.4 Performance Metrics

This subsection introduces the metrics used to evaluate the performance of GBCSS compared with the selected benchmarks. As all metrics are based on the dataset, they are chosen as:

- **Power consumption** P_{tot}: This is the HetNet unit's instantaneous power consumption during a day defined in Eq. (6.1) for each method calculated based on Eq. (3.6). Measured in Watts (W), this metric evaluates the performance of each solution as it reflects the variations in network power consumption in different time steps of the day.

- **Total energy saved** E_{saving}: The total energy saved is another straightforward yet essential metric to assess the performance of GBCSS. Compared to the All-on method, which does not consider energy-saving, it is calculated as $E_{saving} = E_M - E_{ON}$, where E_{ON} and E_M are the total energy consumption with All-on method and with one of the cell-switching solutions: exhaustive search, the FA-based solution and GBCSS, such that $E_M \in \{E_{GNN}, E_{ES}, E_{FA}\}$. The calculation of daily total energy consumption E for each method following the dataset time steps as follows:

$$E = \sum_{t=1}^{N_{slots}} P_{tot}^{t} \times 60 \times 10 \tag{6.18}$$

where P_{tot}^{t} is the power consumption (W) of the HetNet unit at time step t. As t is presented in 10-minute resolution in the Milan dataset, one day (24 hours) leads to $N_{slots} = 144$. Additionally, since the evaluation process may include multiple-day samples, the average values among different day samples are further calculated to represent E in such cases.

- **Normalised network traffic load** $\Lambda_{\%}$: This metric is the HetNet unit's sum traffic load after offloading normalised by that before offloading. As the All-on method does not implement any offloading and cell-switching schemes, thus can always preserve the original traffic loads. This metric can hence be calculated as $\Lambda_{\%} = \frac{\Lambda_M}{\Lambda_{ON}}$, where $\Lambda_M \in \{\Lambda_{GNN}, \Lambda_{ES}, \Lambda_{FA}\}$ is the sum traffic load after offloading using the covered solutions and Λ_{ON} is the sum traffic load using the All-on method. Following Eq. (6.3) - Eq. (6.4) and Eq. (6.10c), the sum traffic load of one day (24 hours) using any of the covered solutions is calculated as:

$$\Lambda = \sum_{t=1}^{N_{slots}} (\hat{\lambda}_t^1 + \sum_{i=2}^{N_{SC}+1} \phi_i \hat{\lambda}_t^i) \tag{6.19}$$

  where $N_{slots} = 144$ in the Milan dataset for 10-minute time steps.

- **Normalised energy efficiency** $\eta_{\%}$: This is the daily energy efficiency of the HetNet unit implemented cell-switching solutions, normalised by that without cell-switching (i.e. All-on). Similar to that of $\Lambda_{\%}$, this relative energy efficiency is calculated as $\eta_{\%} = \frac{\eta_M}{\eta_{ON}}$, where $\eta_M \in \{\eta_{GNN}, \eta_{ES}, \eta_{FA}\}$ is the energy efficiency of the HetNet unit using the corresponding cell switching solution while η_{ON} is that without cell switching. The energy efficiency η using any of the covered solutions is calculated as:

$$\eta = \frac{\Lambda}{E} \tag{6.20}$$

6.5 Results and Discussions

Following the evaluation setups, this section covers the experimental result analysis for the proposed GBCSS, compared with other benchmarks. Qualitative discussions regarding GBCSS with some state-of-the-art solutions are also included in this section.

For learning-based solutions (GNN and FA), an offline training stage was first carried out. The trained GNN and FA's policy were then exported to produce statistical results (i.e. metrics E_{saving} and $\Lambda_{\%}$ with respect to N_{sc}) using the validation dataset. Finally, the two-day samples in the test dataset are used to emulate the online deployment for cell-switching execution that provides results for P_{tot} throughout the day (24 hours). Unless otherwise stated, the results for each N_{sc} case are generated using the GNN trained with the dataset generated for that case. Note that during the online execution phase, it is possible to update the learning models using the latest collected data to

further improve the models' performances. However, such online model updating is beyond the scope of this work.

Before presenting the results regarding each metric, it is also important to analyse the convergence behaviours of the GNN training. Using the configured GNN setups, the loss function value defined in Eq. (6.15) was collected during the training stage. For all considered N_{sc}, the GNN model managed to converge within the first 20 epochs for 7 out of 8 N_{sc} cases, with the minimum epochs for convergence being 5, and the maximum epochs around 55. As the loss records for all 8 N_{sc} cases cannot be summarised clearly in a graphical manner, the essential information has been presented above.

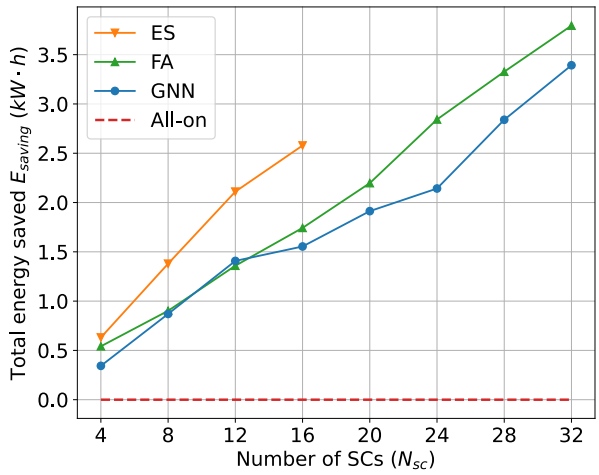
6.5.1 Statistical Results from Validation Set

Figure 6.6 shows the results of metrics E_{saving}, $\Lambda_{\%}$, and $\eta_{\%}$ with respect to N_{sc}. The average values using the 4-day samples in the validation dataset are calculated for the metrics. It is noteworthy that the ES algorithm has only been executed for $N_{sc} \in \{4, 8, 12, 16\}$ due to the time consumption burden as the algorithm is highly computationally demanding with a complexity of $O(2^{\mathbf{N}})$. This means that the processing time for the ES algorithm doubles for every unit N_{sc} increment. In contrast, GBCSS learns to find a sub-optimal solution that approximates the optimality as much as possible while maintaining a much lower computational complexity of $O(\mathbf{N})$.

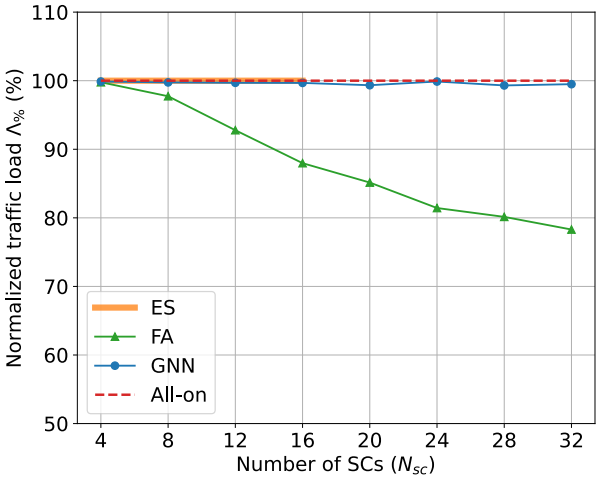
The metric E_{saving} is the optimisation objective for cell switching solutions according to the problem definition in Eq. (6.10a), and is an essential metric to consider. It can be seen in Figure 6.6a that the daily total energy saved increases when N_{sc} is raised for all cell-switching methods, based on the fact that deploying more SCs leads to increased power consumption, while creating more possibilities for offloading and cell switching when the MC has sufficient resource to take over and hence larger energy saving.

For $N_{sc} \in \{4, 8, 12, 16\}$, the saved energy using the ES algorithm is the highest among the considered solutions and can be expected to remain so for larger N_{sc} values if ES was to be executed. For GBCSS, the energy saved is lower than that of ES. For $N_{sc} \in \{4, 8, 12, 16\}$, the GBCSS achieves 53.97%, 63.04%, 66.82%, and 60.08% of ES' E_{saving} performance, resulting in a 62% E_{saving} performance for the 4 N_{sc} cases. Moreover, the GNN is able to further increase the E_{saving} for a large number of deployed SCs as the slope of the E_{saving} curve has clearly increased for $N_{sc} \in \{24, 28, 32\}$. A detailed discussion regarding this aspect is covered in the one-day performance analysis with more supporting results.

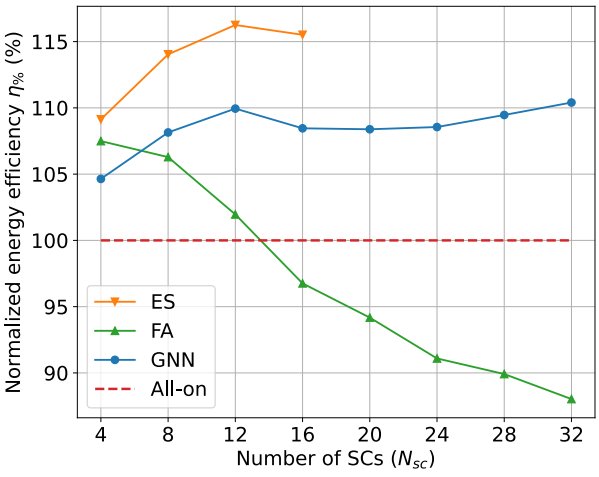
Interestingly, the E_{saving} using the FA benchmark is clearly larger than that of GBCSS for most considered N_{sc} cases except for $N_{sc} = 8$ and 12, in which both solutions result in similar E_{saving}. GBCSS can achieve a maximum of 103.61% and a minimum of



(a)



(b)



(c)

Figure 6.6: Statistical results from the validation set for different N_{sc} (a) Total energy saved E_{saving}. (b) Relative traffic load $\Lambda_{\%}$. (c) Normalised energy efficiency $\eta_{\%}$. ES is not executed for $N_{sc} > 16$ due to huge time consumption. ES has the highest $\eta_{\%}$ (theoretical upper bound) since it achieves the highest energy saving while sacrificing no original traffic; The FA benchmark has the second highest energy saving but compromises too much traffic with the increment of SC number and thus has a declining $\eta_{\%}$. GBCSS has a moderate energy saving in comparison but has almost no data traffic comprise and hence has a satisfactory $\eta_{\%}$.

62.28% E_{saving} performances compared with using the FA, with an average of 86.60% E_{saving} performance compared with using the FA for all N_{sc} cases. This suggests that the FA benchmark outperforms GBCSS in raw energy saving.

However, it is equally important to also consider the metric $\Lambda_{\%}$, which indicates how much of the original traffic load without cell switching (i.e. All-on) can be preserved using different cell-switching solutions and represents the optimisation constraint defined in Eq. (6.10c). According to its definition, the maximum value for $\Lambda_{\%}$ is 100%, which means that all original traffic load is preserved after cell switching execution.

Figure 6.6b shows this metric with a reference red dashed line of the All-on method stands for the upper bound. It can be seen in the figure that ES has $\Lambda_{\%} = 100\%$ for $N_{sc} \in \{4, 8, 12, 16\}$, and is reasonable to assume this trend will be consistent for other N_{sc} cases. In comparison, using the proposed GBCSS results in an average $\Lambda_{\%}$ of 99.63% for all 8 N_{sc} cases, with a maximum of 99.88% and minimum of 99.31%. This suggests that the GNN learns to preserve the user QoS as much as possible when reducing the HetNet unit's energy consumption.

In contrast, it can be seen that the $\Lambda_{\%}$ using FA decreases from 99.77% for $N_{sc} = 4$ to 78.30% for $N_{sc} = 32$. This means that compared to GBCSS, the extra energy saved when using the FA benchmark as shown in Figure 6.6a will cost 21% of the original traffic load and hence the user QoS in the worst case. The reason is that using the offline trained FA algorithm for online decision-making leads to much more frequent decision-making that causes the MC to overload and thus user QoS downgrade, as only the MC can take over the traffic load of an SC according to the problem formation.

Considering both energy consumption and traffic loads, Figure 6.6c shows the normalised daily energy efficiency $\eta_{\%}$ for the considered cell switching solutions with respect to All-on. It is clear that $\eta_{\%}$ of using the ES algorithm is the highest and achieves an average $\eta_{\%}$ of 13.74% among the N_{sc} cases, with a maximum energy efficiency gain of 16.25% compared to that of All-on, while $\eta_{\%}$ using the FA solution drops continuously and becomes even lower than that of All-on due to a large proportion of original traffic load being sacrificed to achieve higher power saving. In comparison, GBCSS achieves an average and maximum $\eta_{\%}$ of 8.50% and 10.41% respectively compared to All-on. The trend of $\eta_{\%}$ using GBCSS is similar to that of ES based on the results for $N_{sc} \in \{4, 8, 12, 16\}$ according to Figure 6.6c, while overall the energy efficiency gain using the GNN is about 62% for these N_{sc} cases. Moreover, assuming the average $\eta_{\%}$ (13.74%) using the ES is preserved for $N_{sc}in\{20, 24, 28, 36\}$, the GNN can achieve a maximum 75.76% of ES' performance regarding energy efficiency gain.

Nevertheless, the FA benchmark still outperforms the proposed GBCSS when $N_{sc} = 4$

with FA's $\eta_{\%}$ being around 2.5% larger as in Figure 6.6c. A potential reason is that the GNN is not able to further approximate the optimal solution when the gradient calculated via the loss function Eq. (6.15) becomes too small, as learning to always switch on the MC leads to a large $\mathcal{L}$ improvement when training the GBCSS. In comparison, the FA benchmark avoids such situations as the action for the MC has been predefined to be always ON. However, the relative underperformance of GNN, in this case, can be regarded as insignificant as the overall energy saved in this case is low due to only 4 SCs being deployed.

6.5.2 Test Set Performance Results

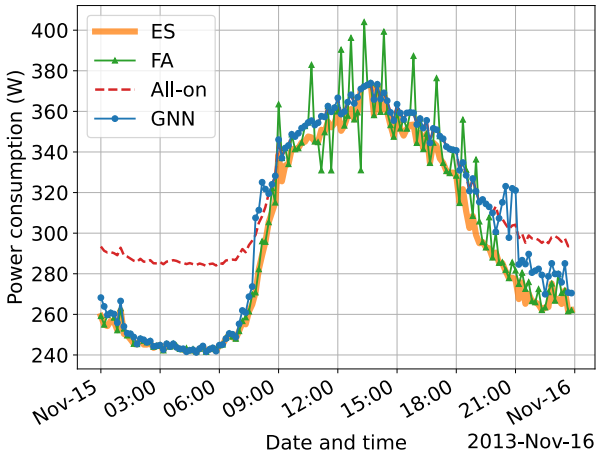
The results generated with the test dataset for one-day power consumption using each solution are presented for 3 N_{sc} cases (i.e. $N_{sc} \in \{4, 12, 32\}$) that represent scenarios of a small, medium and a large number of deployed SCs within the considered N_{sc} cases. The results of the node size generalisation test for the GNN are also covered in this section.

6.5.2.1 Performance comparison on workday samples

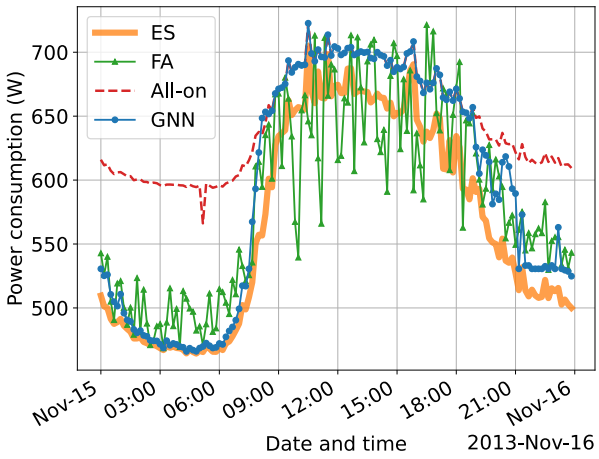
Figure 6.7 shows the power consumption per time step using GBCSS and other benchmarks throughout a workday (from 00:00 a.m. to 11:59 p.m.) for the three N_{sc} cases. Due to the same computational complexity reason as for statistical results analysis, the ES algorithm is not executed to generate results for $N_{sc} = 32$.

According to Eq. (3.6) and Eq. (6.2), the power consumption calculation is a linear transformation of λ when no BS is put into sleep. Therefore, a HetNet unit's traffic load trend throughout a day can be inferred by the power consumption trend of the All-on method. It can be seen in Figure 6.7 that the HetNet unit's power consumption is relatively low before dawn with only a small number of active users, while the traffic load starts to rise around 8 a.m. and peaks before midday, leading to an increased power consumption period with less potential for power saving. Later, the traffic load start to decline more significantly in the late afternoon (4 p.m.), leading to another period for energy efficiency optimisation using cell switching.

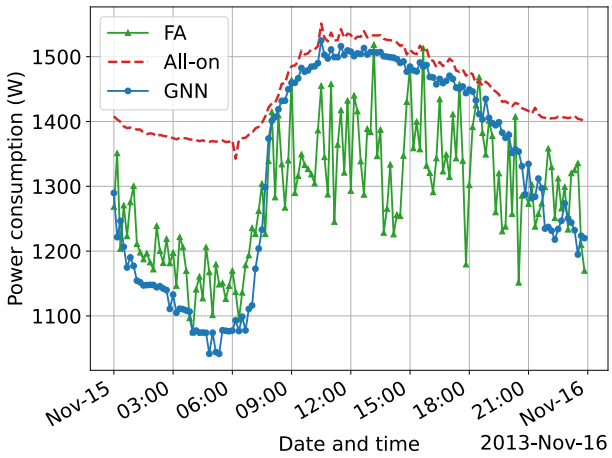
As shown in Figure 6.7a, all 3 cell-switching solutions are able to significantly reduce the power consumption from 0 a.m. to 8 a.m. During this period, the power consumption using GBCSS highly mirrors the behaviour of the ES algorithm. During the high-traffic hours, GBCSS turns to follow the strategy of All-on, which is a suboptimal strategy for this time period. From 4 p.m. until midnight, the GNN also learns to reduce the HetNet unit's power consumption, but the performance is not as significant as it does in the time period before dawn compared to the optimal results computed via ES. In contrast, the FA benchmark also mirrors the behaviour of ES over the day and overall



(a) $N_{sc} = 4$



(b) $N_{sc} = 12$



(c) $N_{sc} = 32$

Figure 6.7: One-day performance results for the workday sample (Nov. 15$^{\text{th}}$, 2013) in the test set with respect to power consumption for different N_{sc}.

outperforms GBCSS, especially after 4 p.m. Moreover, during the busy hours between 9 a.m. and 4 p.m., it can be seen that for some time steps, the power consumption of using the FA benchmark becomes less than that of using ES. Because ES produces the optimal cell switching decisions for power saving while maintaining the original traffic loads in the HetNet unit, it can be inferred that FA's further power-saving comes from sacrificing the user QoS.

For the $N_{sc} = 12$ case in Figure 6.7b, the behaviour of the ES algorithm remains the same as in the $N_{sc} = 4$ case, while a larger gap can be found compared with the power consumption of All-on, suggesting a larger potential for energy efficiency optimisation. Similarly, GBCSS also demonstrates comparable results consistent to those in Figure 6.7a, with the performance after 4 p.m. also improved compared to that in the $N_{sc} = 4$ case. However, the results of the FA benchmark start to have more significant fluctuations in Figure 6.7b, with obviously lower power consumption compared with using the ES during busy hours. Combined with the results in Figure 6.6b, this means that the FA benchmark starts to output more decisions that cause user QoS sacrifices.

As for the $N_{sc} = 32$ case in Figure 6.8c, the fluctuation in the results of the FA benchmark has even worsened with the number of decisions sacrificing the user QoS further rises. An obvious explanation for this trend is that the FA benchmark utilises the linear function approximation technique to represent the value function, which may not have enough expressiveness for scenarios with higher complexity. In contrast, GBCSS shows much more stable results that are consistent with those for $N_{sc} = 4$ and 12. Moreover, GBCSS also starts to switch off SCs during busy hours, and the power consumption during this period becomes smaller than that of All-on for $N_{sc} = 32$ according to Figure 6.7c. This is much more similar to the strategy that ES produces based on results in Figure 6.7a and Figure 6.7b. As discussed in the above section, the main reason for it can be that the loss function cannot be significantly optimised when N_{sc} is small, following the calculation in Eq. (6.15). Moreover, cell switching during a time period with intensive traffic mainly results in marginal power consumption improvement for small N_{sc}, as shown by the results using the ES algorithm. In contrast, a larger N_{sc} leads to more potential for a significant loss reduction during busy hours. This can be regarded as an advantage to exploit because the envisioned ultra-dense HetNet development beyond 5G will result in significantly large numbers of SCs to be deployed, where the GNN may find great potential in approximating the optimal cell switching decision. All the results presented in this section so far correspond to the discoveries in Figure 6.6.

Additionally, it can be seen in Figure 6.7 that sometimes using GBCSS and the FA benchmark results in more power consumption than using the All-on method during

the busy hours for $N_{sc} = 4$ and 8. This raises another question as it is counter-intuitive to have such observations that switching off some BSs causes more power consumption than always keeping all the SCs on. However, considering Eq. (3.6) together with the parameters in Table 6.1, it is possible for certain cell switching decisions to cause an overall larger power consumption by offloading to the MC. For example, switching off a half-loaded Femto BS results in a 2.1W power consumption reduction under the experiment configuration, but the MC taking over the offloaded traffic (assuming sufficient resources) will have its power consumption raised by 47W, which leads to a -44.9 W power consumption gain. A formal mathematical proof can be found in [54] that uses the same power model and BS power profiles.

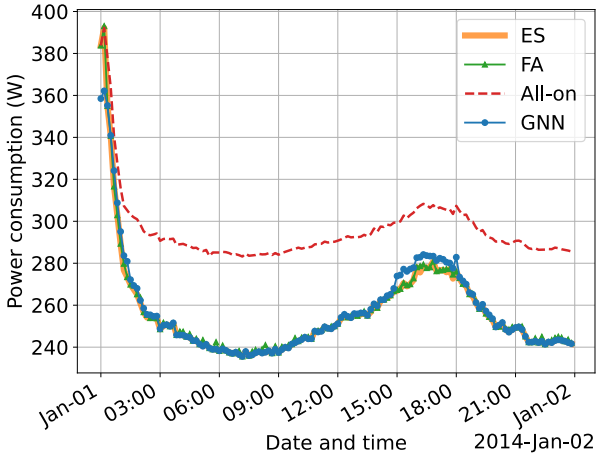
In summary, the proposed GBCSS is able to closely approximate the optimal cell switching decisions computed by the ES algorithm when the total traffic load on the HetNet unit is low while tending to generate suboptimal strategies during the time period with intensive traffic. Nevertheless, such suboptimal strategy during busy hours can be improved when N_{sc} becomes larger (Figure 6.7c) when the GNN starts to mirror the behaviours of ES as in Figure 6.7a and 6.7b. The one-day performance evaluation on a workday produces results that closely correspond to the statistical results generated from the validation dataset.

6.5.2.2 Performance Comparison on Holiday Samples

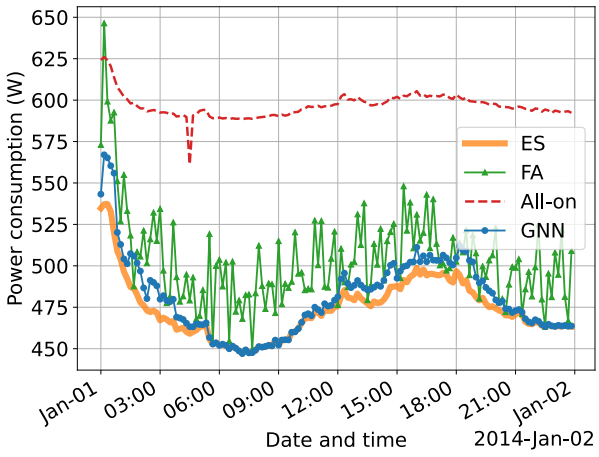
Under the same setup, Figure 6.8 shows the power consumption using different cell switching solutions on the New Year's Day holiday (2014/01/01). The trending in the figures corresponds with the event of people celebrating New Year's Eve, leading to a large number of active users and hence high power consumption throughout the early hours after midnight. In comparison, the overall power consumption during daytime is more stable compared with that during the workday sample in Figure 6.7.

Furthermore, it is clear that using cell-switching solutions results in significant power savings during the daytime. This is similar to the two power-saving time periods in Figure 6.7, suggesting that during such a holiday, mobile service requests during normal busy hours are not as intensive compared to that in a workday. Moreover, in Figure 6.8a, the power consumption using both GNN and FA is nearly identical to the optimal results using the ES benchmark. In addition, the GNN makes no decisions that cause the power consumption to be higher than All-on, and FA also performs significantly better in this regard. The reasoning behind this phenomenon is that learning-based solutions learn to capture the power-saving potential during low-activity time periods better than during the high-activity periods, combined with the results in Figure 6.7.

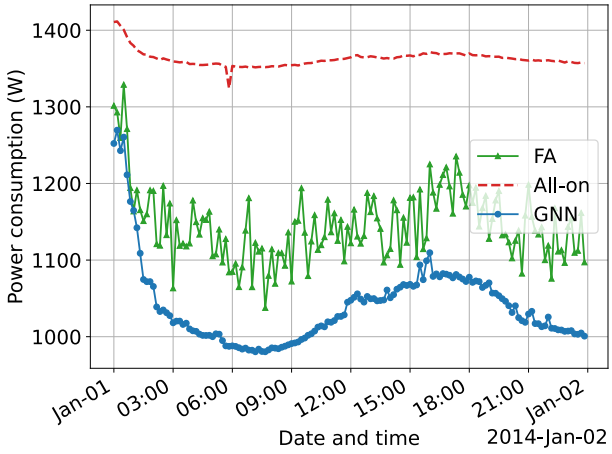
Other results found in Figure 6.8 are highly comparable to the findings in Figure 6.7, such as the results using the FA benchmark have fluctuations with the magnitude



(a) $N_{sc} = 4$



(b) $N_{sc} = 12$



(c) $N_{sc} = 32$

Figure 6.8: One-day performance results for the holiday sample (Jan. 1st, 2014) in the test set with respect to power consumption for different N_{sc}.

increases for a larger N_{sc}, while the GNN is more stable in this regard. As these aspects are already discussed in the workday case, this section includes no further elaborations.

6.5.2.3 Generalisation Capability on Node Size

A remarkable feature of GNN models is their node size invariance, indicating that as long as the data with a similar underlying topology can be expressed using the same graph representation, a GNN model trained using data of node size i can be directly used to produce results for node size j ($i \neq j$). This feature greatly boosts the generalisation capability of GNN models compared with other ML models, leading to a significant cost reduction when deploying GNN models to different scenarios for a defined task.

Therefore, this section presents the node size generalisation test for the proposed GBCSS. The workday data samples in the test dataset are used. Two GNN models trained with training data of $N_{sc} = 4$ and 32 are applied in this test, while the node size for the test case is $N_{sc} = 12$ for both models to give a clearer comparison. Because RL-based solutions need to confirm the feature space and/or action space that cannot be naturally extended by the model itself without reapplication, the FA benchmark is hence not applicable in this evaluation.

The one-day power consumption results of this test are shown in Figure 6.9. These results show that both models trained with different node sizes (both larger and smaller node sizes during the training stage) can be directly utilised in the $N_{sc} = 12$ scenario. For the two lower-traffic periods, 0 a.m. to 8 a.m. and after 4 p.m., both models generate comparable results to that in the same node size scenarios in Figure 6.8b. Furthermore, it can be seen that the models follow some detail from what was learned in the original node size scenario. For example, the GNN model trained with $N_{sc} = 4$ produces some sub-optimal decisions that lead to higher power consumption around 9 a.m., similar to that in Figure 6.7a, while the GNN model trained with $N_{sc} = 32$ tends to result in large power consumption around 0 a.m., which corresponds to the behaviour in Figure 6.7c. Unfortunately, the model trained with $N_{sc} = 32$ does not maintain the strategy to switch off some SCs for power saving as in Figure 6.7c for $N_{sc} = 12$, while keeps mirroring All-on during busy hours, similar to that in Figure 6.7b. The reason for this may still be the learned loss function characteristics, that a smaller N_{sc} leads to insignificant loss improvement for cell switching during busy hours, as discussed for the workday case.

The node size generalisation test results suggest that models trained with one node size can be directly applied to a similar scenario with another node size. Although the performance may not be optimal, this feature can greatly reduce the cost of the model transfer, as the whole GNN model can be directly utilised without any preparatory

steps. After the transfer, the model can be updated using data collected in the new scenario to learn the underlying patterns to improve performance.

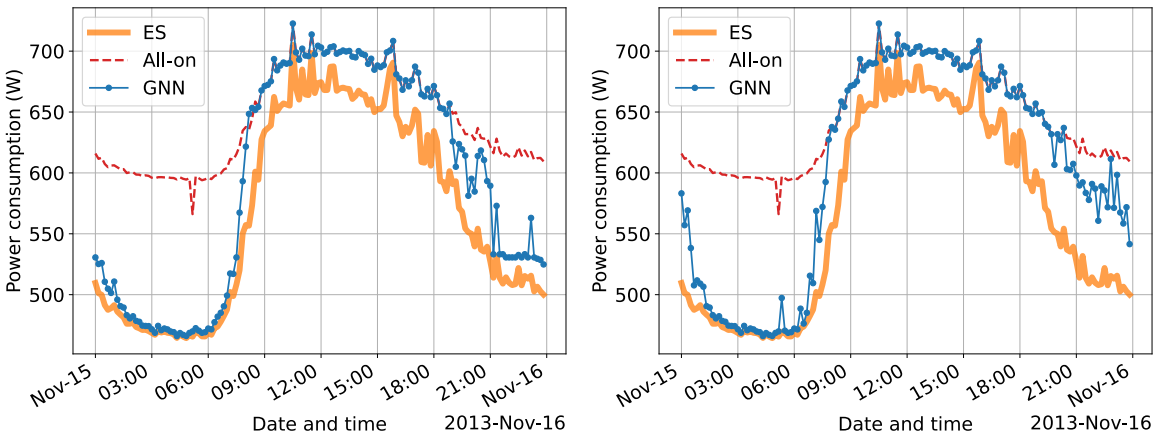


Figure 6.9: One-day power consumption results for the GNN's node size generalisation test, with models trained using two different node sizes tested with $N_{sc} = 12$. (a) $N_{sc} = 4$ for training. (b) $N_{sc} = 32$ for training.

6.6 Summary

The development of cellular networks has led to the proliferation of network deployment with BSs being the major energy consumers in cellular networks. This has resulted in calls for greater energy efficiency to meet green and sustainable cellular network demands when applied to real-life network deployments and architectures. This is also true for the V2N communication case of C-V2X as cellular VUEs use the same interface as general UEs such as mobile phones. As GNN has the significant features of learning graph-structured data to improve training robustness and node size invariance that largely reduce the computational cost for redeployment, this work initially explores a GNN-based cell switching solution (GBCSS) for a CDSA HetNet which can be deployed at each macro BS and is capable of learning the optimal policy in a dense HetNet environment to save energy and while maintaining the user QoS. The performance of GBCSS approach was then evaluated against other benchmarks including the baseline without any cell switching, the theoretical upper bound performance calculated by the exhaustive search algorithm, and a state-of-the-art RL algorithm. Overall, this work covers a theoretical derivation of the problem, an algorithmic design, and simulations using real-world data from the Milan telecommunication dataset.

Experimental results showed that the GBCSS approach can attain 10.41% energy efficiency gains compared to the baseline power with no cell switching while maintaining an average of 99.63% of the original traffic loads for differing numbers of BSs, suggesting that virtually no user QoS is sacrificed while reducing energy consumption. This performance is 75.76% of the optimum results computed by the ES algorithm.

Additionally, the GNN model trained using data from only workdays generalises well to both workday and holiday test cases and is capable of learning the pattern for cell switching during busy hours in a larger node size (number of SCs deployed) setup for further performance improvement. Node size generalisation tests were also performed, with the results supporting the notable feature of GNN's node size invariance that models trained using data of one node size can be directly utilised in scenarios with different node sizes. Furthermore, GBCSS has a computational complexity of $O(\mathbf{N})$ for online execution and is thus much more scalable compared to the ES and similar algorithms of $O(2^{\mathbf{N}})$ as discussed in the complexity analysis.

The proposed GBCSS produces satisfactory energy saving in the network with almost no impact on the user QoS while showing great potential for a large number of deployed SCs. Besides, the proposed solution has a very good generalisation ability and scalability. All these results make GBCSS a promising candidate for practical cell switching applications to future ultra-dense HetNets. With the development of beyond 5G comes new energy efficiency and network intelligence demand, and the world is also witnessing the rise of energy prices, implementing GBCSS will result in significant energy cost savings, while also relieving the deployment cost of the learning-based algorithm. This will significantly relieve the operational cost for both developed and developing markets. Although the utilised real-world dataset comprises of general CDR data, the generated insights can also be applied to V2N use cases.

Chapter 7

Conclusions, Future Trends, and Open Issues

Vehicular networks are a key enabler to improve road safety and traffic efficiency and to support a variety of infotainment services for future intelligent transportation development. The fundamental characteristics of vehicular networks are the high mobility of road vehicles and heterogeneous communication and service types. With the evolution of vehicular networks, a new requirement for cost-effective and sustainable network development has also been raised. Three research topics have been studied: Handover (HO) management, resource allocation, and energy efficiency at the network side for cellular vehicular networks centred around the core idea of adaptive network optimisation. Specifically, three detailed research questions have been developed from the literature review and answered by the research work.

In this chapter, conclusions are drawn for each detailed research question developed from these research topics, including the summary of contributions, limitations, and future work proposal. Open issues and future trends are also provided to each research topic and ML applications to vehicular networks as a whole.

7.1 Contribution Statement

Vehicle's high mobility and network densification lead to frequent HOs that can greatly influence the quality of V2N services, leading to the first research question: **If deployed within the current cellular architecture, does an ML-based HO decision-making scheme improve HO performances for seamless V2N communications?** A well-establish DRL algorithm, Double Deep Q-network (DDQN), was utilised to develop a HO algorithm for V2N communications in Chapter 4. The proposed DDQN HO algorithm used the same input parameter as conventional HO solutions, i.e., the

Reference Signal Received Power (RSRP), and its performance was evaluated using the full-stack network simulator ns-3 emulating a cellular network protocol stack against the A3 HO algorithm. The proposed DRL algorithm can achieve near-optimal HO decision-making through exhaustive case analysis, with a 42.62% reduction in accumulated packet loss compared to the A3 RSRP HO algorithm baseline. This DRL HO algorithm can be theoretically implemented by software patching to the current network with little network architecture modification, answering the research question with clear positivity, and extending the state-of-the-art research from proof-of-concept solution designs towards the deployment of high-performing ML-based solutions.

The second research question considers effective resource management to improve network performance: **Can the latest decentralised learning architecture be utilised to improve the decentralised radio resource allocation and reduce communication overheads for V2N and general cellular communication networks**? A cellular V2X with the sidelink setup of underlay, Mode 2 was considered in Chapter 5, with the decentralised learning framework, Federated Learning (FL), explored for the development of a fully decentralised, DRL-based resource allocation scheme for joint spectrum and power allocation in for spectrum-sharing V2V and V2N communications. Evaluated against other state-of-the-art DRL solutions, the proposed algorithm outperformed the benchmarks by having a smaller communication overhead of 40.8% with much less frequent signalling for training, while having a matched performance in V2V payload delivery ratio and V2N transmission rate. Furthermore, the proposed quantisation-based ANN model compression could reduce the ANN model size by 73.9% with only a 3% performance sacrifice. This research also examined two different types of DRL algorithms with the FL structure and proved the viability of both algorithms. The main advantage of the proposed solution is the simple global knowledge sharing through model aggregation compared with the state-of-the-art solutions without FL while still having small communication overheads by requiring no direct information-sharing in the algorithmic design.

Aiming at green and cost-effective network development, the third research question asks: **Can the latest ML techniques be leveraged to improve cell switching decision-making performance for energy efficiency optimisation for V2N and general cellular communication networks**? This question was answered by Chapter 6, where the GNN-based Cell-switching Solution (GBCSS) was proposed for a Heterogeneous Network (HetNet) under the Control Data Separated Architecture (CDSA) configuration. The proposed GBCSS contained a graph representation of the BSs in a HetNet unit and a GNN-based cell switching algorithm and was evaluated against the optimal solution calculated by the Exhaustive Search (ES) algorithm, an RL-based state-of-the-art solution, and the baseline without any cell-switching. The evaluation was based on the Milan telecommunication dataset comprising real-world

call detail record data to be more realistic. The experimental results showed that the GBCSS could obtain 94.98% of ES's performance regarding the normalised energy efficiency, which is 11.55% higher compared to the RL benchmark, with almost all (over 99%) user QoS (original traffic) preserved. Although the performance is still suboptimal, GBCSS has a much smaller computational complexity (linear to the number of considered BSs) compared to ES (exponential to the number of BSs). Exploiting the feature of a GNN model, the proposed GBCSS had a very strong generalisation capability, which could be directly deployed without any model update to operate with satisfactory performance in a similar scenario with a different network configuration. As BSs are the main energy consumers in cellular network systems, the GBCSS with a close performance to the theoretical upper bound could greatly reduce the energy consumption in cellular networks to help achieve the sustainable development goal and reduce the network operators' OpEx. Its generalisation ability also has the great potential to reduce the redeployment cost for real-world applications. Overall, the proposed GBCSS greatly extends the cell-switching research by exploring the GNN model and the demonstration of high performance and generalisation ability through effective design of the graph model and cell switching algorithm.

In summary, the key contributions of research work for all three research questions are:

- Effective solution proposal which considers real-world deployment aspects
- Filling the gap of the state-of-the-art with respect to algorithmic development: exploring different RL design including the overall setting, the state space, and reward design; exploring the GNN models receiving less attention in the research topics.
- Proof-of-concept and experimental demonstration of the deployment potentials for all proposed solutions.

7.2 Future Trends and Open Issues

7.2.1 Future Trends for Research Topics

7.2.1.1 Handover management

As 5G is being rolled out globally, future research extending the work of the proposed DRL HO algorithm presented in Chapter 4 is to first update the implementation of the proposed DDQN HO algorithm using ns-3's latest standardised cellular module, 5G-LENA [217]. The 5G-LENA module became publically available at the time the proposed algorithm was being implemented but is still under active development requiring updating with the 5G HO interfaces [277]. The aim is to validate the

performance of the proposed algorithm in a 5G network setup while also examining how many neighbouring BSs are sufficient for the state space design, especially with the inclusion of mmWave that have distinct features in the radio environment. Other research direction aiming at learning-based algorithm optimisations, such as state space reduction, is also relevant to this research to reduce the foreseeable communication overheads in the control plane.

Beyond the research question studied in this thesis, HOs can be extended to different scenarios following different system setups, such as intra-frequency (within the same frequency range) and inter-frequency (between different frequency ranges), as well as inter-RAT HOs (between different RATs such as from a cellular BS to a DSRC access point) [120, 154, 210]. As 5G already has two frequency bands (Frequency Range 1 and 2) while also aiming to accomplish the internetworking among different RATs to form a unified wireless network [18, 278, 279], all the above scenarios are bringing new challenges to HO management with additional variables to consider for future vehicular networks. Moreover, it is essential for future HO algorithms to consider three-dimensional scenarios, such as vehicular networks on/under viaducts or tunnels and future 3D networks or ground, air, and space networks. To be specific, 3D networks are envisioned to integrate road vehicles, Unmanned Aerial Vehicles (UAVs) and satellites, which have high complexity on the decision-making of when and which entity to connect to [280]. In these scenarios, communication links can be on different layers and cross layers, where ML is envisioned to play a key role in HO decision-making for complex environments.

7.2.1.2 Resource Allocation

For future work on the considered resource allocation problem presented in Chapter 5, the FL deployment aspects are an important research direction by considering how communication-related factors influence the FL training process. For example, how unreliable communication channels influence the federated model aggregation and its effect on the overall performance [281]. Furthermore, encompassing a hierarchical DRL algorithm as in [138] to the current design remains another promising future direction to enable automatic adaptation to different communication environments such as different network congestion levels influenced by the pattern of road traffic.

MEC and NFV are extensively studied and key technologies for adoption in vehicular networks as well [282]. In vehicular networks, ML must deal with high network dynamics, which is further enhanced by scenarios of viaducts, tunnels, and future 3D networks, leading to an even more complex network environment. In addition to the HO management requirement as introduced above, the vision of 3D networks also includes differentiated on-demand services such as Intelligent Transportation Systems (ITS) services and remote surgery, further requiring effective resource allocation to

fulfil the stringent service requirement [280]. MEC enables distributed solutions by moving less computationally demanding tasks from cloud computing centres to network edges, which helps in latency improvement and gives the possibility to exploit data stored in different network nodes. In comparison, NFV introduces more flexibility to the network through hardware abstraction, where ML methods can be applied and form a more general solution independent of hardware. Applying ML methods with these technologies in vehicular networks faces significant challenges from the highly dynamic vehicular networks and needs specific attention. With the deployment of MEC, computation resources including computation power and data storage become another type of valuable resource and the joint resource allocation optimisation of different resource types should be further investigated beyond the existing work such as [242, 283]. An interesting direction is to develop a joint solution for computation load-balancing and data storage management (what data is worth storing) among MEC nodes to improve MEC efficiency and fault detection for network slices in NFV for the network operation maintenance and safety measures.

7.2.1.3 Energy Efficiency via Cell Switching

Future research in the cell switching aspect extending the proposed GNN-based cell switching solution (GBCSS) includes combing RL algorithms and GNN to further improve the GBCSS' convergence to the optimality and thus further enhance the performance. Another viable future work direction is exploring the heterogeneous graph representation for a cell switching problem. Incorporating date and time information in the heterogeneous graph representation in addition to power and throughput information to improve robustness for GBCSS remains of high importance. Extending the binary cell switching decision-making into multi-level sleep mode is also a viable future direction. This is because deep sleep mode usually requires more time for a BS to reactivate, which may lead to slowness in face of rapid response requirements (e.g., another BS going offline and the network requiring service reinforcement). Multi-level sleep mode hence gives the flexibility for more generalised cases for energy-saving. Moreover, as the work of GBCSS mainly focused on algorithmic design, investigating how GBCSS and learning-based algorithms, in general, may be deployed in a real-world scenario considering detailed protocol stacks is also a valuable direction. One possible approach is to implement the GBCSS and benchmarks in a network simulator such as ns-3 and evaluate using real-time generated data in a realistic network simulation.

For ML-based cell switching solutions in general, the real-world deployment issues such as data processing and storage, and redeployment for online model update remain an outstanding issue for future research and development. Besides, ML-based joint energy efficiency scheme development remains a possible direction for future resources, such as joint decision-making of cell switching and link adaptation, following the different energy efficiency schemes introduced in Chapter 3.

7.2.2 Machine Learning for Vehicular Networks

Beyond the research topics, several major issues exist in the state-of-the-art ML applications to vehicular networks in general.

7.2.2.1 Data Collection and Processing

Data is the fundamental ingredient to support ML algorithmic developments. Vehicular networks are rich in data as various data sources exist in the network including onboard sensors of vehicles such as cameras and communication components. However, with gigantic data being produced in vehicular networks, what types of data should be collected becomes the fundamental and foremost question to answer to reduce data redundancy. Moreover, as data are distributed on different network nodes, while the overall data volume is enormous, managing the data storage becomes an important challenge to address regarding how much and what type of data to store. This is highly important, especially for centralised ML solutions that require data to be managed and trained by a central controller. Finally, the data quality in vehicular networks is heterogeneous, e.g., the image quality of photos taken from different cameras can have significant differences due to e.g., out-of-focus and blur caused by motion. How to process the data to guarantee the overall data quality to train high-performing ML algorithms remains an outstanding topic for the research community.

7.2.2.2 Distributed learning and Multi-agent Cooperation for ML in Vehicular Networks

In contrast to traditional ML application scenarios, data in vehicular networks is generated and stored in different network units, such as vehicles and BSs. Therefore, individual vehicles would not have access to all the rich data sources demanded by certain types of learning tasks. These expectations lead to learning on local partially-observed data while exploiting values from data stored in other devices. A multi-agent setup for ML methods can thus promise better performance at a system level through cooperation in vehicular networks to avoid the aforementioned systematic communication delay and overhead of a purely centralised setup for data transfer. The setup can also utilise the computational power of vehicles to make decisions locally. Research has investigated in that direction such as the research that inspired the work in Chapter 5 with a cooperative MARL setup [25, 138, 243]. However, these existing designs have the disadvantage of individual agents having access to only local data, and thus lacking knowledge of the global state or data distribution.

Federated Learning (FL) is another decentralised ML technique to train models via distributed data and learner cooperation, which can also exploit value through MEC as shown by some recent work [246, 284]. It has also been identified as a key enabler for the development of future ML-based vehicular networking solution [30]. The main

advantage of FL is that it enables knowledge sharing among individual learners through local model sharing to produce a global model with values extracted from distributed data sources. Attempts have also been made in this thesis to explore the algorithmic design using the FL framework. However, given its potential importance to vehicular networks with identified future trends introduced above, key challenges still exist for FL with respect to its deployment. Specifically, how to cope with the extra stages of global model aggregation in real-time applications, how to deploy FL aggregators and the heterogeneous data issues as discussed above, which need further exploration.

7.2.2.3 ML Method Complexity Issues

ANN-based learning methods have achieved continuous performance breakthroughs in various areas. Many current ML solutions in vehicular networks have ANN-based methods or combined ANNs with different ML techniques [138, 148, 174, 285], e.g., ANN-based prediction with SVM classification, with top performances reaching greater than 99% accuracy in tasks such as mobility prediction [285]. Future vehicular networks will need to extend current two-dimensional scenarios for handover and resource allocation to three-dimensional ones with possible UAV integration, causing overall task complexity to continue to rise. ANN-based DL is currently the best solution for its descriptive power. However, the training process of ANNs often takes up a significant amount of computation resources because of the deep ANN structure. Unlike traditional ML methods, ANNs can learn directly from raw data while not requiring dedicated feature design or causing large data batches, either of which increase overall computation complexity.

Sophisticated DL methods have significantly improved performance but can lead to greater latency and reduced energy efficiency. While devices such as self-powered RSUs have limited computation power, the requirement of regulating complexity arises for computation alleviation while coping with performance requirements. One practical solution lies in model compression that reduces the ANN model size while maintaining much original performance, which has also been adopted in this thesis. However, more detailed implementation aspects need to be explored in this realm, such as where to place the compression, on user devices or a network controller? Computation offloading is also a promising solution in this topic. Leveraging mobile cloud and edge computing, computation offloading avoids the direct reduction on a DL model complexity but transfers the training to a remote cloud server to complete or divided into less computationally intensive subtasks, which are then distributed among network edge nodes to share training results. However, the accompanying communication overhead issues need to be investigated for offloading approaches and DL models need to be transmitted, while the task dividing and reconstruction quality should also be considered for the edge computing approach. To conclude, different designs for model compression should be assessed for real-world deployment, while the execution design

of computation offloading for ANN training in vehicular networks needs more research in the future.

7.2.2.4 Privacy and Security Issues

It can be seen from the above aspects that either data or ML models need to be transmitted among different entities within the network, e.g., data transmission for centralised processing and model transmission for federated learning and computation offloading. These raise the need to guarantee the privacy and security of ML applications in the network. Specifically, collected data used for ML applications may contain sensitive data such as user identity, positioning data, etc. Since a potential data leakage of such data will cause serious consequences, privacy protection becomes vitally important for real-world ML applications in the network. Moreover, model transmission for the above situations can be in danger of cyber attacks, causing damage to the network. For example, if a transmitted ML model is hacked and cannot provide desired outputs, the corresponding functionality will be heavily damaged and may even lead to casualties when safety-critical applications are concerned. As a result, the privacy and security aspects of ML applications in vehicular networks may need to be further explored in this realm.

7.2.2.5 Dataset and Simulation Environment Standardisation for ML Adoption

Regarding ML methodology adoption in the field, a remaining issue is the standardisation of research datasets and simulation environments. To assist researchers to focus on learning algorithm design and to simplify performance comparison, common problems should be identified with related datasets, while simulation environments should also be standardised in common with other ML areas. A good example is the Modified National Institute of Standards and Technology (MNIST) database [286] used for image recognition tasks and the Open AI Gym environment [287] employed for RL algorithm development. Some work has been done in these aspects of vehicular networks, such as Klautau et al.'s [288] presentation of a dataset for mmWave beam selection and a simulator for vehicle traffic and raytracing. Another work developed the first framework for RL research in networking, combining the ns-3 network simulator and Open AI gym [289], with an inspired study also developed ns-3 interconnections with popular ML frameworks [229]. However, more research resources are necessary to fill this gap, which requires enriched datasets able to perform more networking tasks and standardised networking simulation environments that support ML integration.

7.2.2.6 Interpretability and Trust for ML methods

Another major challenge for ML adoption is the "BlackBox" problem for complex methods, such as DL techniques. As decision-making procedures in DL algorithms

presently cannot produce output that can be interpreted directly by humans, it creates risks for safety- and security-related applications that cause trust issues involving verification and legal liability confusion when accidents occur. As ML methods have attracted more research interests in vehicular networks, adding interpretability for such systems becomes crucial to enable drivers and network operators to understand system behaviour, introducing the possibility of user-based control, justification, and improved performance [290]. Interpretable DL-based systems for vehicular networks should also be developed to assist the legal system with traceable interpretability-founded responsibility made available for legal judgement. A recent advance in CNN hidden-layer neuron activity visualisation tools has shown promising means of supporting non-experts in understanding the DL process of CNN training [291]. This can be a valuable example for related research to improve the interpretability of other ANN-based methods and help address potential liability issues.

The research questions studied in this thesis have only addressed a small part of the challenges demanded by the industry to deploy ML-based solutions in a real network, with other outstanding issues for ML solution development. Nevertheless, through working closely with the industry to address the identified open issues, it should be possible for ML solutions to greatly improve the user experience beyond 5G.

Bibliography

[1] L. Huawei Technologies Co. "Smart transportation: Maximize mobile network's value beyond connectivity." (2022), [Online]. Available: https://www-file.huawei.com/~/media/CORPORATE/PDF/white%5C%20paper/smart-transportation.pdf.

[2] IEA. "Global car sales by key markets, 2005-2020." License: CC BY 4.0. (), [Online]. Available: https://www.iea.org/data-and-statistics/charts/global-car-sales-by-key-markets-2005-2020.

[3] L. Figueiredo *et al.*, "Towards the development of intelligent transportation systems," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, 2001, pp. 1206–1211. DOI: 10.1109/ITSC.2001.948835.

[4] J. Zhang *et al.*, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011. DOI: 10.1109/TITS.2011.2158001.

[5] W. H. Organization, *Road traffic injuries*, Accessed 30 Sep., 2022. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[6] W. H. Organization. "Global status report on road safety 2018." (Jun. 2018), [Online]. Available: https://apps.who.int/iris/rest/bitstreams/1164010/retrieve.

[7] D. Flore. "5g v2x: The automotive use-case for 5g." (2017), [Online]. Available: https://www.3gpp.org/ftp/information/presentations/Presentations_2017/A4Conf010_Dino%5C%20Flore_5GAA_v1.pdf.

[8] T. I. BV., *Tomtom traffic index: Ranking 2021*, Accessed 30 Sep., 2022. https://www.tomtom.com/traffic-index/ranking/.

[9] M. Sweet, "Does traffic congestion slow the economy?" *Journal of Planning Literature*, vol. 26, no. 4, pp. 391–404, 2011.

[10] E. Struyf, C. Sys, E. Van de Voorde, and T. Vanelslander, "Calculating the cost of congestion to society: A case study application to flanders," *Research in Transportation Business & Management*, vol. 44, p. 100 573, 2022, Challenges and solutions for current freight transport and logistics, ISSN: 2210-5395. DOI: https://doi.org/10.1016/j.rtbm.2020.100573. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210539520301115.

[11] L. Chapman, "Transport and climate change: A review," *Journal of Transport Geography*, vol. 15, no. 5, pp. 354–367, 2007, ISSN: 0966-6923. DOI: https://doi.org/10.1016/j.jtrangeo.2006.11.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0966692306001207.

[12] M. Barth and K. Boriboonsomsin, "Traffic congestion and greenhouse gases," *Access Magazine*, vol. 1, no. 35, pp. 2–9, 2009.

[13] H. T. Cheng, H. Shan, and W. Zhuang, "Infotainment and road safety service support in vehicular networking: From a communication perspective," *Mechanical Systems and Signal Processing*, vol. 25, no. 6, pp. 2020–2038, 2011, Interdisciplinary Aspects of Vehicle Dynamics, ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2010.11.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327010004127.

[14] A. Sumalee and H. W. Ho, "Smarter and more connected: Future intelligent transportation system," *IATSS Research*, vol. 42, no. 2, pp. 67–71, 2018, ISSN: 0386-1112. DOI: https://doi.org/10.1016/j.iatssr.2018.05.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0386111218300396.

[15] J. Misener. "Smart transportation." (Aug. 2020), [Online]. Available: https://www.qualcomm.com/media/documents/files/smart-transportation-presentation.pdf.

[16] P. K. Singh, S. K. Nandi, and S. Nandi, "A tutorial survey on vehicular communication state of the art, and future research directions," *Vehicular Communications*, vol. 18, p. 100 164, 2019.

[17] N. Aljeri and A. Boukerche, "Mobility management in 5g-enabled vehicular networks: Models, protocols, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–35, 2020.

[18] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1064–1078, 2018.

[19] L. Liang, H. Peng, G. Y. Li, and X. Shen, "Vehicular communications: A physical layer perspective," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10 647–10 659, 2017. DOI: 10.1109/TVT.2017.2750903.

[20] S. Chen *et al.*, "A vision of c-v2x: Technologies, field testing, and challenges with chinese development," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3872–3881, 2020. DOI: 10.1109/JIOT.2020.2974823.

[21] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5g for vehicular communications," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 111–117, 2018. DOI: 10.1109/MCOM.2018.1700467.

[22] R. Calabrò. "C-v2x deployment: Continuing the momentum in 2021." (May 2021), [Online]. Available: https://5g-mobix.com/assets/files/IEEE-C-V2X-Deployment-Riccardo-Calabro-FINAL.pdf.

[23] 5GAA, *Automotive c-v2x ready to roll out globally, says 5gaa at this year's mwc barcelona*, Accessed 2020. https://5gaa.org/news/automotive-c-v2x-ready-to-roll-out-globally-says-5gaa-at-this-years-mwc-barcelona/.

[24] C. Shin *et al.*, "Vehicle-to-everything (v2x) evolution from 4g to 5g in 3gpp: Focusing on resource allocation aspects," *IEEE Access*, vol. 11, pp. 18 689–18 703, 2023. DOI: 10.1109/ACCESS.2023.3247127.

[25] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2282–2292, 2019.

[26] M. Placek. "Connected cars worldwide - statistics & facts." Accessed 1 Oct., 2022. (Dec. 2021), [Online]. Available: https://www.statista.com/topics/1918/connected-cars/#dossierContents__outerWrapper.

[27] C. Freitag *et al.*, "The real climate and transformative impact of ict: A critique of estimates, trends, and regulations," *Patterns*, vol. 2, no. 9, p. 100 340, 2021, ISSN: 2666-3899. DOI: https://doi.org/10.1016/j.patter.2021.100340. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666389921001884.

[28] F. Salahdine *et al.*, "A survey on sleep mode techniques for ultra-dense networks in 5g and beyond," *Computer Networks*, vol. 201, p. 108 567, 2021, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2021.108567. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621004801.

[29] A. Srivastava, M. S. Gupta, and G. Kaur, "Energy efficient transmission trends towards future green cognitive radio networks (5g): Progress, taxonomy and open challenges," *Journal of Network and Computer Applications*, vol. 168, p. 102 760, 2020, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2020.102760. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520302344.

[30] Z. Zhang *et al.*, "6g wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, 2019. DOI: 10.1109/MVT.2019.2921208.

[31] 5GAA. "A visionary roadmap for advanced driving use cases, connectivity technologies, and radio spectrum needs." (Sep. 2020), [Online]. Available: https://5gaa.org/wp-content/uploads/2020/09/A-Visionary-Roadmap-for-Advanced-Driving-Use-Cases-Connectivity-Technologies-and-Radio-Spectrum-Needs.pdf.

[32] I. Qualcomm Technologies, *5g nr based c-v2x*, Accessed 2020. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/powerpoint_messaging_-_5g_nr_based_c-v2x.pdf.

[33] 3GPP, "Service requirements for enhanced V2X scenarios," 3rd Generation Partnership Project (3GPP), Technical specification (TS) 22.886, Sep. 2018,

Version 16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3180.

[34] N. Bhushan *et al.*, "Network densification: The dominant theme for wireless evolution into 5g," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 82–89, 2014. DOI: 10.1109/MCOM.2014.6736747.

[35] 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, 2017. DOI: 10.1109/MCOM.2017.1700487.

[36] K. Tan *et al.*, "Machine learning in vehicular networking: An overview," *Digital Communications and Networks*, vol. 8, no. 1, pp. 18–24, 2022, ISSN: 2352-8648. DOI: https://doi.org/10.1016/j.dcan.2021.10.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864821000870.

[37] 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, 2019. DOI: 10.1109/COMST.2019.2904897.

[38] N. H. Hussein *et al.*, "A comprehensive survey on vehicular networking: Communications, applications, challenges, and upcoming research directions," *IEEE Access*, vol. 10, pp. 86 127–86 180, 2022. DOI: 10.1109/ACCESS.2022.3198656.

[39] G. Miao, J. Zander, K. W. Sung, and S. B. Slimane, *Fundamentals of mobile data networks*. Cambridge University Press, 2016.

[40] W. Viriyasitavat, M. Boban, H.-M. Tsai, and A. Vasilakos, "Vehicular communications: Survey and challenges of channel and propagation models," *IEEE Vehicular Technology Magazine*, vol. 10, no. 2, pp. 55–66, 2015.

[41] S. K. Bhoi and P. M. Khilar, "Vehicular communication: A survey," *IET Networks*, vol. 3, no. 3, pp. 204–217, 2014. DOI: https://doi.org/10.1049/iet-net.2013.0065. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-net.2013.0065. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-net.2013.0065.

[42] M. Feng, S. Mao, and T. Jiang, "Base station on-off switching in 5g wireless networks: Approaches and challenges," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 46–54, 2017. DOI: 10.1109/MWC.2017.1600353.

[43] D. Chen. "5g power: Creating a green grid that slashes costs, emissions & energy use." (Jul. 2020), [Online]. Available: https://www.huawei.com/en/technology-insights/publications/huawei-tech/89/5g-power-green-grid-slashes-costs-emissions-energy-use.

[44] L. hardesty. "5g base stations use a lot more energy than 4g base stations: Mtn." (Apr. 2020), [Online]. Available: https://www.fiercewireless.com/tech/5g-base-stations-use-a-lot-more-energy-than-4g-base-stations-says-mtn.

[45] G. Auer *et al.*, "How much energy is needed to run a wireless network?" *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40–49, 2011. DOI: 10.1109/MWC.2011.6056691.

[46] T. M. Mitchell and T. M. Mitchell, *Machine learning*. McGraw-hill New York, 1997, vol. 1.

[47] D. L. Poole and A. K. Mackworth, *Artificial Intelligence: foundations of computational agents*. Cambridge University Press, 2010.

[48] J. Wang *et al.*, "Thirty years of machine learning: The road to pareto-optimal wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1472–1514, 2020. DOI: 10.1109/COMST.2020.2965856.

[49] H. Ye *et al.*, "Machine learning for vehicular networks: Recent advances and application examples," *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 94–101, 2018. DOI: 10.1109/MVT.2018.2811185.

[50] 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, 2017. DOI: 10.1109/COMST.2017.2727878.

[51] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and iot networks: Potentials, current solutions, and open challenges," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1251–1275, 2020. DOI: 10.1109/COMST.2020.2964534.

[52] Y. Sun *et al.*, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019. DOI: 10.1109/COMST.2019.2924243.

[53] V. Yajnanarayana, H. Rydén, and L. Hévizi, "5g handover using reinforcement learning," in *2020 IEEE 3rd 5G World Forum (5GWF)*, IEEE, 2020, pp. 349–354.

[54] M. Ozturk *et al.*, "Energy optimization in ultra-dense radio access networks via traffic-aware cell switching," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 832–845, 2021. DOI: 10.1109/TGCN.2021.3056235.

[55] K. Tan, J. L. Kernec, M. Imran, and D. Bremner, "Clustering algorithm in vehicular ad-hoc networks: A brief summary," in *2019 UK/ China Emerging Technologies (UCET)*, 2019, pp. 1–5. DOI: 10.1109/UCET.2019.8881833.

[56] K. Tan, D. Bremner, J. L. Kernec, and M. Imran, "Federated machine learning in vehicular networks: A summary of recent applications," in *2020 International Conference on UK-China Emerging Technologies (UCET)*, 2020, pp. 1–4. DOI: 10.1109/UCET51115.2020.9205482.

[57] K. Tan *et al.*, "Intelligent handover algorithm for vehicle-to-network communications with double-deep q-learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7848–7862, 2022. DOI: 10.1109/TVT.2022.3169804.

[58] K. Tan *et al.*, "Graph neural network-based cell switching for energy optimization in ultra-dense heterogeneous networks," *Scientific Reports*, vol. 12, no. 1, p. 21 581, Dec. 2022, ISSN: 2045-2322. DOI: 10.1038/s41598-022-25800-3. [Online]. Available: https://doi.org/10.1038/s41598-022-25800-3.

[59] M. L. Sichitiu and M. Kihl, "Inter-vehicle communication systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 2, pp. 88–105, 2008. DOI: 10.1109/COMST.2008.4564481.

[60] A. Festag, "Standards for vehicular communication—from ieee 802.11 p to 5g," *e & i Elektrotechnik und Informationstechnik*, vol. 132, no. 7, pp. 409–416, 2015.

[61] H. Wei, G. Zheng, V. V. Gayah, and Z. Li, "A survey on traffic signal control methods," *CoRR*, vol. abs/1904.08117, 2019. arXiv: 1904.08117. [Online]. Available: http://arxiv.org/abs/1904.08117.

[62] D. Garcia-Roger, E. E. González, D. Martín-Sacristán, and J. F. Monserrat, "V2x support in 3gpp specifications: From 4g to 5g and beyond," *IEEE Access*, vol. 8, pp. 190 946–190 963, 2020. DOI: 10.1109/ACCESS.2020.3028621.

[63] Y. Mao *et al.*, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017. DOI: 10.1109/COMST.2017.2745201.

[64] M. Giordani, A. Zanella, and M. Zorzi, "Millimeter wave communication in vehicular networks: Challenges and opportunities," in *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2017, pp. 1–6. DOI: 10.1109/MOCAST.2017.7937682.

[65] T. Bai and R. W. Heath, "Coverage and rate analysis for millimeter-wave cellular networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 1100–1114, 2015. DOI: 10.1109/TWC.2014.2364267.

[66] P. P. Shinde and S. Shah, "A review of machine learning and deep learning applications," in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–6. DOI: 10.1109/ICCUBEA.2018.8697857.

[67] C. Gambella, B. Ghaddar, and J. Naoum-Sawaya, "Optimization problems for machine learning: A survey," *European Journal of Operational Research*, vol. 290, no. 3, pp. 807–828, 2021.

[68] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[69] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[70] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[71] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[72] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, pp. 1–27, 2011.

[73] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[74] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020.

[75] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: An overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 1, pp. 86–97, 2012.

[76] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[77] 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)*, IEEE, 2018, pp. 135–139.

[78] M. Ayyub *et al.*, "A comprehensive survey on clustering in vehicular networks: Current solutions and future challenges," *Ad Hoc Networks*, vol. 124, p. 102 729, 2022, ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2021.102729. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870521002183.

[79] M. Usama *et al.*, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE Access*, vol. 7, pp. 65 579–65 615, 2019. DOI: 10.1109/ACCESS.2019.2916648.

[80] C. O'Reilly, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Anomaly detection in wireless sensor networks in a non-stationary environment," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1413–1432, 2014. DOI: 10.1109/SURV.2013.112813.00168.

[81] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[82] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[83] OpenAI. "Openai spinning up, part 2: Kinds of rl algorithms." (2021), [Online]. Available: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html.

[84] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[85] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994*, Elsevier, 1994, pp. 157–163.

[86] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*, K. G. Vamvoudakis, Y. Wan, F. L. Lewis, and D. Cansever, Eds. Cham: Springer International Publishing, 2021, pp. 321–384, ISBN: 978-3-030-60990-0. DOI: 10.1007/978-3-030-60990-0_12. [Online]. Available: https://doi.org/10.1007/978-3-030-60990-0_12.

[87] G. Owen, "Game theory," in Emerald Group Publishing, 2013, pp. 13–31, ISBN: 9780470978504.

[88] G. Owen, "LTE-the UMTS long term evolution: from theory to practice," in Emerald Group Publishing, 2013, pp. 159–183, ISBN: 9781781905074.

[89] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth international conference on machine learning*, 1993, pp. 330–337.

[90] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[91] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[92] D. Silver *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[93] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015. DOI: 10.1016/j.neunet.2014.09.003. [Online]. Available: https://doi.org/10.1016%5C%2Fj.neunet.2014.09.003.

[94] M. F. Abbod, J. W. Catto, D. A. Linkens, and F. C. Hamdy, "Application of artificial intelligence to the management of urological cancer," *The Journal of Urology*, vol. 178, no. 4, pp. 1150–1156, 2007, ISSN: 0022-5347. DOI: https://doi.org/10.1016/j.juro.2007.05.122. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022534707013936.

[95] O. I. Abiodun *et al.*, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, e00938, 2018, ISSN: 2405-8440. DOI: https://doi.org/10.1016/j.heliyon.2018.e00938. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405844018332067.

[96] T. Mikolov *et al.*, "Recurrent neural network based language model.," in *Interspeech*, Makuhari, vol. 2, 2010, pp. 1045–1048.

[97] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[98] R. HECHT-NIELSEN, "Iii.3 - theory of the backpropagation neural network**based on "nonindent" by robert hecht-nielsen, which appeared in proceedings of the international joint conference on neural networks 1, 593–611, june 1989. © 1989 ieee.," in *Neural Networks for Perception*, H. Wechsler, Ed., Academic Press, 1992, pp. 65–93, ISBN: 978-0-12-741252-8. DOI: https://

doi.org/10.1016/B978-0-12-741252-8.50010-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780127412528500108.

[99] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Springer, 2010, pp. 177–186.

[100] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006, Neural Networks, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2005.12.126. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231206000385.

[101] B. Widrow, A. Greenblatt, Y. Kim, and D. Park, "The no-prop algorithm: A new learning algorithm for multilayer neural networks," *Neural Networks*, vol. 37, pp. 182–188, 2013, Twenty-fifth Anniversay Commemorative Issue, ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2012.09.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608012002614.

[102] Y. Ollivier, C. Tallec, and G. Charpiat, *Training recurrent networks online without backtracking*, 2015. DOI: 10.48550/ARXIV.1507.07680. [Online]. Available: https://arxiv.org/abs/1507.07680.

[103] Y. Li, "Deep reinforcement learning: An overview," *CoRR*, vol. abs/1701.07274, 2017. arXiv: 1701.07274. [Online]. Available: http://arxiv.org/abs/1701.07274.

[104] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," in MIT press, 2018, pp. 195–341.

[105] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[106] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," in MIT press, 2018, pp. 131–133.

[107] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[108] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine learning*, vol. 8, no. 3, pp. 293–321, 1992.

[109] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," in MIT press, 2018, 62-64 and 140–141.

[110] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," *CoRR*, vol. abs/1602.01783, 2016. arXiv: 1602.01783. [Online]. Available: http://arxiv.org/abs/1602.01783.

[111] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12,

MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

[112] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," in MIT press, 2018, pp. 209–232.

[113] F. Monti *et al.*, "Geometric deep learning on graphs and manifolds using mixture model cnns," *CoRR*, vol. abs/1611.08402, 2016. arXiv: 1611.08402. [Online]. Available: http://arxiv.org/abs/1611.08402.

[114] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020, ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.01.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000012.

[115] F. Scarselli *et al.*, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009. DOI: 10.1109/TNN.2008.2005605.

[116] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. DOI: 10.1109/ICEngTechnol.2017.8308186.

[117] D. Bacciu, F. Errica, A. Micheli, and M. Podda, "A gentle introduction to deep learning for graphs," en, *Neural Networks*, vol. 129, pp. 203–221, Sep. 2020, ISSN: 0893-6080. DOI: 10.1016/j.neunet.2020.06.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608020302197 (visited on 03/28/2022).

[118] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, vol. abs/1602.05629, 2016. arXiv: 1602.05629. [Online]. Available: http://arxiv.org/abs/1602.05629.

[119] J. Konečný *et al.*, *Federated learning: Strategies for improving communication efficiency*, 2016. arXiv: 1610.05492 [cs.LG].

[120] 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. DOI: 10.1109/ACCESS.2019.2937405.

[121] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 341–356, 2020. DOI: 10.1109/JPROC.2019.2957798.

[122] S. Khan Tayyaba *et al.*, "5g vehicular network resource management for improving radio access through machine learning," *IEEE Access*, vol. 8, pp. 6792–6800, 2020. DOI: 10.1109/ACCESS.2020.2964697.

[123] A. Talpur and M. Gurusamy, "Machine learning for security in vehicular networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 346–379, 2022. DOI: 10.1109/COMST.2021.3129079.

[124] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive survey on machine learning in vehicular network: Technology, applications and challenges," *IEEE*

Communications Surveys & Tutorials, vol. 23, no. 3, pp. 2027–2057, 2021. DOI: 10.1109/COMST.2021.3089688.

[125] A. Abdelmohsen *et al.*, "Lte handover parameters optimization using q-learning technique," in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2018, pp. 194–197. DOI: 10.1109/MWSCAS.2018.8623826.

[126] T. Goyal and S. Kaushal, "Handover optimization scheme for lte-advance networks based on ahp-topsis and q-learning," *Computer Communications*, vol. 133, pp. 67–76, 2019.

[127] N. AlJeri and A. Boukerche, "An efficient movement-based handover prediction scheme for hierarchical mobile IPv6 in VANETs," in *Proceedings of the 15th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, ACM Press, 2018, pp. 47–54, ISBN: 9781450359610. DOI: 10.1145/3243046.3243053. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3243046.3243053.

[128] L. Rabiner and B. Juang, "An introduction to hidden markov models," *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[129] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[130] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with gaussian mixture models," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 141–146.

[131] Y. Xing, C. Lv, and D. Cao, "Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1341–1352, 2020. DOI: 10.1109/TVT.2019.2960110.

[132] K. Qi, T. Liu, and C. Yang, "Federated learning based proactive handover in millimeter-wave vehicular networks," in *2020 15th IEEE International Conference on Signal Processing (ICSP)*, vol. 1, 2020, pp. 401–406. DOI: 10.1109/ICSP48669.2020.9320974.

[133] H. Sun *et al.*, "Learning to optimize: Training deep neural networks for wireless resource management," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017, pp. 1–6. DOI: 10.1109/SPAWC.2017.8227766.

[134] M. Lee, G. Yu, and G. Y. Li, "Graph embedding-based wireless link scheduling with few training samples," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2282–2294, 2021. DOI: 10.1109/TWC.2020.3040983.

[135] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1760–1776, 2020. DOI: 10.1109/TCOMM.2019.2957482.

[136] N. Naderializadeh, "Wireless Link Scheduling via Graph Representation Learning: A Comparative Study of Different Supervision Levels," *arXiv e-prints*, arXiv:2110.01722, arXiv:2110.01722, Oct. 2021. arXiv: 2110.01722 [eess.SP].

[137] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for v2v communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3163–3173, 2019. DOI: 10.1109/TVT.2019.2897134.

[138] Y. Yuan, G. Zheng, K.-K. Wong, and K. B. Letaief, "Meta-reinforcement learning based resource allocation for dynamic v2x communications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 8964–8977, 2021. DOI: 10.1109/TVT.2021.3098854.

[139] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[140] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*, PMLR, 2017, pp. 1126–1135.

[141] G. Yu, Q. Chen, and R. Yin, "Dual-threshold sleep mode control scheme for small cells," *IET communications*, vol. 8, no. 11, pp. 2008–2016, 2014.

[142] S. M. Asad *et al.*, "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.

[143] Q. Zhang *et al.*, "Dynamic load adjustments for small cells in heterogeneous ultra-dense networks," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6. DOI: 10.1109/WCNC45663.2020.9120688.

[144] A. E. Amine, P. Dini, and L. Nuaymi, "Reinforcement learning for delay-constrained energy-aware small cells with multi-sleeping control," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6. DOI: 10.1109/ICCWorkshops49005.2020.9145431.

[145] K. Zhang, X. Wen, Y. Chen, and Z. Lu, "Deep reinforcement learning for energy saving in radio access network," in *2020 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, 2020, pp. 35–40. DOI: 10.1109/ICCCWorkshops49972.2020.9209916.

[146] A. I. Abubakar *et al.*, "A lightweight cell switching and traffic offloading scheme for energy optimization in ultra-dense heterogeneous networks," *Physical Communication*, vol. 52, p. 101 643, 2022, ISSN: 1874-4907. DOI: 10.1016/j.phycom.2022.101643. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874490722000283.

[147] G. Jang *et al.*, "Base station switching and sleep mode optimization with lstm-based user prediction," *IEEE Access*, vol. 8, pp. 222 711–222 723, 2020. DOI: 10.1109/ACCESS.2020.3044242.

[148] Q. Wu *et al.*, "Deep reinforcement learning with spatio-temporal traffic forecasting for data-driven base station sleep control," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 935–948, 2021. DOI: 10.1109/TNET.2021.3053771.

[149] 3GPP, "Vocabulary for 3GPP Specifications," 3rd Generation Partnership Project (3GPP), Technical report (TR) 21.905, Dec. 2021, Version 17.1.0. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/21_series/21.905/.

[150] Q. Ye *et al.*, "User association for load balancing in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2706–2716, 2013. DOI: 10.1109/TWC.2013.040413.120676.

[151] D. Liu *et al.*, "User association in 5g networks: A survey and an outlook," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1018–1044, 2016. DOI: 10.1109/COMST.2016.2516538.

[152] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, pp. 1–32, 2013.

[153] 3GPP, "NR; Physical layer measurements," 3rd Generation Partnership Project (3GPP), Technical specification (TS) 38.215, Jan. 2020, Version 16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3217.

[154] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management," 3rd Generation Partnership Project (3GPP), Technical specification (TS) 36.133, Jul. 2021, Version 17.2.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2420.

[155] S. Sesia, I. Toufik, and M. Baker, "LTE-the UMTS long term evolution: from theory to practice," in John Wiley & Sons, Jul. 2011, pp. 63–77, ISBN: 9780470978504.

[156] 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. DOI: 10.1109/ACCESS.2018.2885821.

[157] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), Technical specification (TS) 36.331, Jul. 2021, Version 16.5.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2440.

[158] 3GPP, "NR; Radio Resource Control (RRC); Protocol specification," 3rd Generation Partnership Project (3GPP), Technical specification (TS) 38.331, Apr. 2020, Version 16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3197.

[159] S. Sesia, I. Toufik, and M. Baker, "LTE-the UMTS long term evolution: from theory to practice," in John Wiley & Sons, Jul. 2011, pp. 513–518, ISBN: 9780470978504.

[160] H. Hendrawan, A. R. Zain, and S. Lestari, "Performance evaluation of A2-A4-RSRQ and A3-RSRP handover algorithms in LTE network," *Jurnal Elektronika dan Telekomunikasi*, vol. 19, no. 2, pp. 64–74, Dec. 2019, ISSN: 2527-9955. DOI: 10.14203/jet.v19.64-74. [Online]. Available: https://www.jurnalet.com/jet/article/view/272.

[161] S. Sesia, I. Toufik, and M. Baker, "LTE-the UMTS long term evolution: from theory to practice," in John Wiley & Sons, Jul. 2011, pp. 49–53, ISBN: 9780470978504.

[162] 3GPP, "NR; NR and NG-RAN Overall description; Stage-2," 3rd Generation Partnership Project (3GPP), Technical specification (TS) 38.300, Jan. 2020, Version 16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3191.

[163] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan, "Coordinating handover parameter optimization and load balancing in lte self-optimizing networks," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, 2011, pp. 1–5. DOI: 10.1109/VETECS.2011.5956561.

[164] P. Muñoz, R. Barco, and I. de la Bandera, "On the potential of handover parameter optimization for self-organizing networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1895–1905, 2013. DOI: 10.1109/TVT.2013.2247778.

[165] K. Ghanem, H. Alradwan, A. Motermawy, and A. Ahmad, "Reducing ping-pong handover effects in intra eutra networks," in *2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, 2012, pp. 1–5. DOI: 10.1109/CSNDSP.2012.6292642.

[166] M. S. Mollel *et al.*, "A survey of machine learning applications to handover management in 5g and beyond," *IEEE Access*, vol. 9, pp. 45 770–45 802, 2021. DOI: 10.1109/ACCESS.2021.3067503.

[167] C. Shen, C. Tekin, and M. van der Schaar, "A non-stochastic learning approach to energy efficient mobility management," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3854–3868, 2016.

[168] G. Neu, "Explore no more: Improved high-probability regret bounds for non-stochastic bandits," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[169] S. H. Srikantamurthy and A. Baumgartner, "A novel unified handover algorithm for lte-a," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 407–411. DOI: 10.23919/CNSM52442.2021.9615604.

[170] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[171] M. S. Mollel *et al.*, "Intelligent handover decision scheme using double deep reinforcement learning," *Physical Communication*, vol. 42, p. 101 133, 2020.

[172] Z. Wang *et al.*, "Handover control in wireless systems via asynchronous multiuser deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4296–4307, 2018. DOI: 10.1109/JIOT.2018.2848295.

[173] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13 124–13 138, 2020. DOI: 10.1109/TVT.2020.3020400.

[174] N. Aljeri and A. Boukerche, "A two-tier machine learning-based handover management scheme for intelligent vehicular networks," *Ad Hoc Networks*, vol. 94, p. 101 930, 2019.

[175] M. Noor-A-Rahim *et al.*, "A survey on resource allocation in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 701–721, 2022. DOI: 10.1109/TITS.2020.3019322.

[176] Z. Han and K. R. Liu, *Resource allocation for wireless networks: basics, techniques, and applications*. Cambridge university press, 2008.

[177] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[178] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, 2018. DOI: 10.1109/TCCN.2018.2809722.

[179] I. Qualcomm Technologies, *Introduction to cellular v2x*, Accessed 2020. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/c-v2x_intro.pdf.

[180] 3GPP, "Study on evaluation methodology of new Vehicle-to-Everything (V2X) use cases for LTE and NR," 3rd Generation Partnership Project (3GPP), Technical report (TR) 37.885, Jun. 2017, Version 15.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3209.

[181] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014. DOI: 10.1109/COMST.2014.2319555.

[182] J. C. Haartsen, "The bluetooth radio system," *IEEE personal communications*, vol. 7, no. 1, pp. 28–36, 2000.

[183] F. Loussaief, H. Marouane, H. Koubaa, and F. Zarai, "Radio resource management for vehicular communication via cellular device to device links: Review and

challenges," en, *Telecommunication Systems*, vol. 73, no. 4, pp. 607–635, Apr. 2020, ISSN: 1018-4864, 1572-9451. DOI: 10.1007/s11235-019-00644-x. [Online]. Available: http://link.springer.com/10.1007/s11235-019-00644-x (visited on 04/20/2022).

[184] R. Molina-Masegosa and J. Gozalvez, "Lte-v for sidelink 5g v2x vehicular communications: A new 5g technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017. DOI: 10.1109/MVT.2017.2752798.

[185] 3GPP, "Study on NR Vehicle-to-Everything (V2X)," 3rd Generation Partnership Project (3GPP), Technical report (TR) 38.885, Mar. 2019, Version 16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3497.

[186] F. Romeo, C. Campolo, A. Molinaro, and A. O. Berthet, "Denm repetitions to enhance reliability of the autonomous mode in nr v2x sidelink," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5. DOI: 10.1109/VTC2020-Spring48590.2020.9129367.

[187] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011. DOI: 10.1109/TSP.2011.2147784.

[188] K. Shen and W. Yu, "Fplinq: A cooperative spectrum sharing strategy for device-to-device communications," in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 2323–2327. DOI: 10.1109/ISIT.2017.8006944.

[189] L. Xiao *et al.*, "Mobile crowdsensing games in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1535–1545, 2018. DOI: 10.1109/TVT.2016.2647624.

[190] M. Ozturk, M. Jaber, and M. A. Imran, "Life-span extension for sensor networks in the industry," *Wireless Automation as an Enabler for the Next Industrial Revolution*, pp. 19–45, 2020.

[191] M.-L. Ku, W. Li, Y. Chen, and K. J. Ray Liu, "Advances in energy harvesting communications: Past, present, and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1384–1412, 2016. DOI: 10.1109/COMST.2015.2497324.

[192] R. F. Atallah, C. M. Assi, and J. Y. Yu, "A reinforcement learning technique for optimizing downlink scheduling in an energy-limited vehicular network," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 4592–4601, 2017.

[193] Z. Ning *et al.*, "Mobile edge computing-enabled internet of vehicles: Toward energy-efficient scheduling," *IEEE Network*, vol. 33, no. 5, pp. 198–205, 2019.

[194] X. Wang *et al.*, "Future communications and energy management in the internet of vehicles: Toward intelligent energy-harvesting," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 87–93, 2019.

[195] S. Zhang, S. Xu, G. Y. Li, and E. Ayanoglu, "First 20 years of green radios," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 1, pp. 1–15, 2020. DOI: 10.1109/TGCN.2019.2934531.

[196] T. Chen *et al.*, "Network energy saving technologies for green wireless access networks," *IEEE Wireless Communications*, vol. 18, no. 5, pp. 30–38, 2011. DOI: 10.1109/MWC.2011.6056690.

[197] F. Wang *et al.*, "Design of wide-bandwidth envelope-tracking power amplifiers for ofdm applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 4, pp. 1244–1255, 2005. DOI: 10.1109/TMTT.2005.845716.

[198] Q. Wang *et al.*, "Application of bbu+rru based comp system to lte-advanced," in *2009 IEEE International Conference on Communications Workshops*, 2009, pp. 1–5. DOI: 10.1109/ICCW.2009.5208027.

[199] P. H. Pednekar, W. Hallberg, C. Fager, and T. W. Barton, "Analysis and design of a doherty-like rf-input load modulated balanced amplifier," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 12, pp. 5322–5335, 2018. DOI: 10.1109/TMTT.2018.2869571.

[200] J. Wu, Y. Zhang, M. Zukerman, and E. K.-N. 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. DOI: 10.1109/COMST.2015.2403395.

[201] 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)*, 2015, pp. 1–7. DOI: 10.1109/VTCSpring.2015.7145603.

[202] S. Mertens, "Exhaustive search for low-autocorrelation binary sequences," *Journal of Physics A: Mathematical and General*, vol. 29, no. 18, p. L473, 1996.

[203] B. Polyak and P. Shcherbakov, "Lyapunov functions: An optimization theory perspective**this work was supported by the russian scientific foundation, project no. 16-11-10015," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7456–7461, 2017, 20th IFAC World Congress, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.1513. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896317320955.

[204] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach *et al.*, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[205] A. Orsino, G. Araniti, A. Molinaro, and A. Iera, "Effective rat selection approach for 5g dense wireless networks," in *2015 IEEE 81st vehicular technology conference (VTC Spring)*, IEEE, 2015, pp. 1–5.

[206] K. Ghanem, H. Alradwan, A. Motermawy, and A. Ahmad, "Reducing ping-pong Handover effects in intra EUTRA networks," in *2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, IEEE, 2012, pp. 1–5.

[207] J. Sultan, M. S. Mohsen, N. S. G. Al-Thobhani, and W. A. Jabbar, "Performance of hard handover in 5g heterogeneous networks," in *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, 2021, pp. 1–7. DOI: 10.1109/eSmarTA52612.2021.9515745.

[208] R. Arshad *et al.*, "Handover management in dense cellular networks: A stochastic geometry approach," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7. DOI: 10.1109/ICC.2016.7510709.

[209] H. Tabassum, M. Salehi, and E. Hossain, "Mobility-aware analysis of 5g and B5G cellular networks: A tutorial," *CoRR*, vol. abs/1805.02719, 2018. arXiv: 1805.02719. [Online]. Available: http://arxiv.org/abs/1805.02719.

[210] S. Sesia, I. Toufik, and M. Baker, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[211] G. Çelik, H. Çelebi, and G. Tuna, "A novel RSRP-based E-CID positioning for LTE networks," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2017, pp. 1689–1692. DOI: 10.1109/IWCMC.2017.7986538.

[212] E. Rastorgueva-Foi *et al.*, "User Positioning in mmW 5G Networks Using Beam-RSRP Measurements and Kalman Filtering," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 1–7. DOI: 10.23919/ICIF.2018.8455289.

[213] D. Harris and S. Harris, *Digital design and computer architecture*. Morgan Kaufmann, 2010.

[214] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[215] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*, Springer, 2010, pp. 15–34.

[216] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, "An open source product-oriented lte network simulator based on ns-3," in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, 2011, pp. 293–298.

[217] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, "An e2e simulator for 5g nr networks," *Simulation Modelling Practice and Theory*, vol. 96, p. 101 933, 2019.

[218] T. Cerqueira and M. Albano, "Routesmobilitymodel: Easy realistic mobility simulation using external information services," in *Proceedings of the 2015 Workshop on ns-3*, 2015, pp. 40–46.

[219] G. Developers. "Getting directions through the directions api." (2021), [Online]. Available: `https://developers.google.com/maps/documentation/directions` (visited on 03/2021).

[220] G. Developers. "Getting directions through the directions api." (2021), [Online]. Available: `https://developers.google.com/maps/documentation/directions/get-directions` (visited on 12/2022).

[221] C. mapper community. "Vodafone uk (united kingdom) - cellular coverage and tower map." (2021), [Online]. Available: `https://www.cellmapper.net/map?MCC=234&MNC=15&type=LTE&latitude=0&longitude=0&zoom=11&showTowers=true&showTowerLabels=true&clusterEnabled=true&tilesEnabled=true&showOrphans=false&showNoFrequencyOnly=false&showFrequencyOnly=false&showBandwidthOnly=false&DateFilterType=None&showHex=false&showVerifiedOnly=false&showUnverifiedOnly=false&showLTECAOnly=false&showENDCOnly=false&showBand=0&showSectorColours=true&mapType=roadmap` (visited on 03/2021).

[222] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz," 3rd Generation Partnership Project (3GPP), Technical report (TR) 38.901, Mar. 2017, Version 14.0.0. [Online]. Available: `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173`.

[223] ns-3. "Design documentation of the lte module, fading model." (2021), [Online]. Available: `https://www.nsnam.org/docs/models/html/lte-design.html#fading-model`.

[224] W. C. Jakes and D. C. Cox, *Microwave mobile communications*. Wiley-IEEE press, 1994.

[225] G. Piro, N. Baldo, and M. Miozzo, "An LTE module for the ns-3 network simulator.," in *SimuTools*, 2011, pp. 415–422.

[226] G. C. Council. "Glasgow city centre 20 mph zone - frequently asked questions." (2021), [Online]. Available: `https://www.glasgow.gov.uk/index.aspx?articleid=19202` (visited on 12/2022).

[227] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with gaussian error linear units," *CoRR*, vol. abs/1606.08415, 2016. arXiv: `1606.08415`. [Online]. Available: `http://arxiv.org/abs/1606.08415`.

[228] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017. arXiv: `1711.05101`. [Online]. Available: `http://arxiv.org/abs/1711.05101`.

[229] H. Yin *et al.*, "ns3-ai: Fostering artificial intelligence algorithms for networking research," in *Proceedings of the 2020 Workshop on ns-3*, ser. WNS3 2020,

Gaithersburg, MD, USA: Association for Computing Machinery, Jun. 2020, pp. 57–64, ISBN: 9781450375375. DOI: 10.1145/3389400.3389404. [Online]. Available: https://doi.org/10.1145/3389400.3389404.

[230] ns-3. "User documentation of the lte module, simulation outputs." (2021), [Online]. Available: https://www.nsnam.org/docs/models/html/lte-user.html#simulation-output.

[231] ns-3. "Design documentation of the lte module, pdcp." (2021), [Online]. Available: https://www.nsnam.org/docs/models/html/lte-design.html#pdcp.

[232] ns-3. "Design documentation of the lte module, data phy error model." (2021), [Online]. Available: https://www.nsnam.org/docs/models/html/lte-design.html#data-phy-error-model.

[233] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[234] L. Wang, H. Ye, L. Liang, and G. Y. Li, "Learn to compress CSI and allocate resources in vehicular networks," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3640–3653, Jun. 2020, ISSN: 0090-6778. DOI: 10.1109/TCOMM.2020.2979124. [Online]. Available: https://ieeexplore.ieee.org/document/9026965/.

[235] K. Dimou *et al.*, "Handover within 3GPP LTE: Design Principles and Performance," in *2009 IEEE 70th Vehicular Technology Conference Fall*, 2009, pp. 1–5. DOI: 10.1109/VETECF.2009.5378909.

[236] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep-reinforcement-learning-based mode selection and resource allocation for cellular v2x communications," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6380–6391, 2020. DOI: 10.1109/JIOT.2019.2962715.

[237] T. T. Thanh Le and S. Moh, "Comprehensive survey of radio resource allocation schemes for 5g v2x communications," *IEEE Access*, vol. 9, pp. 123 117–123 133, 2021. DOI: 10.1109/ACCESS.2021.3109894.

[238] F. L. Da Silva and A. H. R. Costa, "A survey on transfer learning for multiagent reinforcement learning systems," *Journal of Artificial Intelligence Research*, vol. 64, pp. 645–703, 2019.

[239] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[240] C. Tan *et al.*, "A survey on deep transfer learning," in *Artificial Neural Networks and Machine Learning – ICANN 2018*, V. Kůrková *et al.*, Eds., Cham: Springer International Publishing, 2018, pp. 270–279, ISBN: 978-3-030-01424-7.

[241] Z. Zhu, K. Lin, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *CoRR*, vol. abs/2009.07888, 2020. arXiv: 2009.07888. [Online]. Available: https://arxiv.org/abs/2009.07888.

[242] D. Li, S. Xu, and P. Li, "Deep reinforcement learning-empowered resource allocation for mobile edge computing in cellular v2x networks," *Sensors*, vol. 21,

no. 2, 2021, ISSN: 1424-8220. DOI: 10.3390/s21020372. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/372.

[243] B. Gu *et al.*, "Multiagent reinforcement learning-based semi-persistent scheduling scheme in c-v2x mode 4," *IEEE Transactions on Vehicular Technology*, pp. 1–13, 2022. DOI: 10.1109/TVT.2022.3189019.

[244] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Jan. 2019, ISSN: 2157-6904. DOI: 10.1145/3298981. [Online]. Available: https://doi.org/10.1145/3298981.

[245] L. De Haan and A. Ferreira, *Extreme value theory: an introduction*. Springer Science & Business Media, 2007.

[246] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, 2020.

[247] 3GPP, "Study on enhancement of 3GPP support for 5G V2X services," 3rd Generation Partnership Project (3GPP), Technical report (TS) 22.886, Jun. 2018, Version 16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3108.

[248] H.-H. Nguyen and W.-J. Hwang, "Distributed scheduling and discrete power control for energy efficiency in multi-cell networks," *IEEE Communications Letters*, vol. 19, no. 12, pp. 2198–2201, 2015. DOI: 10.1109/LCOMM.2015.2418771.

[249] B. McMahan *et al.*, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, A. Singh and J. Zhu, Eds., ser. Proceedings of Machine Learning Research, vol. 54, PMLR, Apr. 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html.

[250] T. Choudhary, V. Mishra, A. Goswami, and J. Sarangapani, "A comprehensive survey on model compression and acceleration," *Artificial Intelligence Review*, vol. 53, no. 7, pp. 5113–5155, Oct. 2020, ISSN: 1573-7462. DOI: 10.1007/s10462-020-09816-7. [Online]. Available: https://doi.org/10.1007/s10462-020-09816-7.

[251] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *CoRR*, vol. abs/1710.09282, 2017. arXiv: 1710.09282. [Online]. Available: http://arxiv.org/abs/1710.09282.

[252] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. DOI: 10.1038/s41586-020-2649-2. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2.

[253] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

[254] 3GPP, "Study on LTE-Based V2X Services," 3rd Generation Partnership Project (3GPP), Technical report (TR) 36.885, Jun. 2016, Version 14.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2934.

[255] T. P. Foundation. "Quantization." Accessed 2022. (), [Online]. Available: https://pytorch.org/docs/stable/quantization.html.

[256] A. Mohamed *et al.*, "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. DOI: 10.1109/COMST.2015.2451514.

[257] 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)*, 2016, pp. 1–5. DOI: 10.1109/WCSP.2016.7752551.

[258] Y. Luo, Z. Shi, F. Bu, and J. Xiong, "Joint optimization of area spectral efficiency and energy efficiency for two-tier heterogeneous ultra-dense networks," *IEEE Access*, vol. 7, pp. 12 073–12 086, 2019. DOI: 10.1109/ACCESS.2019.2891551.

[259] M. J. Daas, M. Jubran, and M. Hussein, "Energy management framework for 5g ultra-dense networks using graph theory," *IEEE Access*, vol. 7, pp. 175 313–175 323, 2019. DOI: 10.1109/ACCESS.2019.2957378.

[260] D. Bacciu, F. Errica, A. Micheli, and M. Podda, "A gentle introduction to deep learning for graphs," *Neural Networks*, vol. 129, pp. 203–221, 2020, ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2020.06.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608020302197.

[261] Z. Wu *et al.*, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021. DOI: 10.1109/TNNLS.2020.2978386.

[262] W. Jiang, "Graph-based deep learning for communication networks: A survey," *Computer Communications*, vol. 185, pp. 40–54, 2022, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2021.12.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421004874.

[263] S. He *et al.*, "An overview on the application of graph neural networks in wireless networks," *CoRR*, vol. abs/2107.03029, 2021. arXiv: 2107.03029. [Online]. Available: https://arxiv.org/abs/2107.03029.

[264] G. Barlacchi *et al.*, "A multi-source dataset of urban life in the city of Milan and the Province of Trentino," en, *Scientific Data*, vol. 2, no. 1, p. 150 055, 2015, Number: 1 Publisher: Nature Publishing Group, ISSN: 2052-4463. DOI: 10/gc4nzj. [Online]. Available: https://www.nature.com/articles/sdata201555 (visited on 02/03/2022).

[265] 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), 2017, pp. 1–6. DOI: 10.1109/WCNC.2017.7925662.

[266] P. J. Kolesar, "A branch and bound algorithm for the knapsack problem," *Management science*, vol. 13, no. 9, pp. 723–735, 1967.

[267] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[268] E. Ranjan, S. Sanyal, and P. P. Talukdar, "ASAP: adaptive structure aware pooling for learning hierarchical graph representations," *CoRR*, vol. abs/1911.07979, 2019. arXiv: 1911.07979. [Online]. Available: http://arxiv.org/abs/1911.07979.

[269] P. Virtanen *et al.*, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

[270] T. pandas development team, *Pandas-dev/pandas: Pandas*, version latest, Feb. 2020. DOI: 10.5281/zenodo.3509134. [Online]. Available: https://doi.org/10.5281/zenodo.3509134.

[271] T. Italia, *Telecommunications - SMS, Call, Internet - MI*, version V1, 2015. DOI: 10.7910/DVN/EGZHFV. [Online]. Available: https://doi.org/10.7910/DVN/EGZHFV.

[272] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[273] A. F. Agarap, *Deep learning using rectified linear units (relu)*, 2018. DOI: 10.48550/ARXIV.1803.08375. [Online]. Available: https://arxiv.org/abs/1803.08375.

[274] P. Sibi, S. A. Jones, and P. Siddarth, "Analysis of different activation functions using back propagation neural networks," *Journal of theoretical and applied information technology*, vol. 47, no. 3, pp. 1264–1268, 2013.

[275] Pytorch Documentation, *Reduce LR on Plateau*, 2022. [Online]. Available: https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html.

[276] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *CoRR*, vol. abs/1903.02428, 2019. arXiv: 1903.02428. [Online]. Available: http://arxiv.org/abs/1903.02428.

[277] K. Koutlia, B. Bojovic, Z. Ali, and S. Lagén, "Calibration of the 5g-lena system level simulator in 3gpp reference scenarios," *Simulation Modelling Practice and Theory*, vol. 119, p. 102 580, 2022, ISSN: 1569-190X. DOI: https://doi.org/10.1016/j.simpat.2022.102580. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X22000697.

[278] 3GPP, "NR; User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone," 3rd Generation Partnership Project (3GPP), Technical specification (TS) 38.101-1, Jan. 2021, Version 17.0.0. [Online]. Available:

https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3283.

[279] 3GPP, "NR; User Equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone," 3rd Generation Partnership Project (3GPP), Technical specification (TS) 38.101-2, Jan. 2021, Version 17.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3284.

[280] E. C. Strinati *et al.*, "6g in the sky: On-demand intelligence at the edge of 3d networks," *CoRR*, vol. abs/2010.09463, 2020. arXiv: 2010.09463. [Online]. Available: https://arxiv.org/abs/2010.09463.

[281] H. Ye, L. Liang, and G. Y. Li, "Decentralized federated learning with unreliable communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 487–500, 2022. DOI: 10.1109/JSTSP.2022.3152445.

[282] M. A. Imran, Y. A. Sambo, and Q. H. Abbasi, "Evolution of vehicular communications within the context of 5g systems," in *Enabling 5G Communication Systems to Support Vertical Industries*. 2019, pp. 103–126.

[283] D. Li, S. Xu, and P. Li, "Deep reinforcement learning-empowered resource allocation for mobile edge computing in cellular v2x networks," *Sensors*, vol. 21, no. 2, 2021, ISSN: 1424-8220. DOI: 10.3390/s21020372. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/372.

[284] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[285] S. Khan Tayyaba *et al.*, "5g vehicular network resource management for improving radio access through machine learning," *IEEE Access*, vol. 8, pp. 6792–6800, 2020.

[286] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[287] G. Brockman *et al.*, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[288] A. Klautau *et al.*, "5g mimo data for machine learning: Application to beam-selection using deep learning," in *2018 Information Theory and Applications Workshop (ITA)*, 2018, pp. 1–9.

[289] P. Gawłowicz and A. Zubow, "ns-3 meets OpenAI Gym: The Playground for Machine Learning in Networking Research," in *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Miami Beach, USA, Nov. 2019. [Online]. Available: http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/2019/gawlowicz19_mswim.pdf.

[290] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[291] Z. J. Wang *et al.*, *Cnn explainer: Learning convolutional neural networks with interactive visualization*, 2020. arXiv: 2004.15004 [cs.HC].