UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DeepWiN: Deep Graph Reinforcement Learning for User-Centric Radio Access Networks Automation

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

BY

Maria Shaukat
Norman, Oklahoma
2023

DeepWIN: Deep Graph Reinforcement Learning for User-Centric Radio Access Networks Automation

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Ali Imran, Chair

Dr. Hazem Refai

Dr. Samuel Cheng

Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Ali Imran, for his continuous support during my M.S. study and research. His expertise, guidance, support, and patience added considerably to my graduate research experience.

I would also like to thank my fellow colleagues in AI4Networks Research Center for their support and providing a research conducive work environment.

I am extremely grateful to my mentors; Dr. Hassam Sheikh, Dr. Zeenat Hameed, and my family; parents and brothers for their prayers. Finally, thank you to Mrs. Denise Davis for her continued support during the graduate program.

Table of Contents

List of Figures

List of Tables

Abstract

The future cellular networks are expected to support an increasing number of users with heterogeneous applications, requiring varying network resources. Therefore, the 6G and beyond cellular networks need to be elastic, and user-centric. User-centric Radio Access Networks (UCRAN), with virtual cells (S-zones), can provide on-demand connectivity, coverage and quality of service to different user applications while optimizing the network for energy efficiency, area spectral efficiency, reliability and user service rate. However, with high variability in the network, due to user mobility and fading, the selection of S-zone sizes which optimize the network performance for multiple types of users simultaneously becomes a challenge. Therefore, to automate the selection of S-zone sizes dynamically, we propose deep graph reinforcement learning (DGRL), a Soft actor-critic model integrated with Graph neural network. DGRL infers from DeepWiN, a graphical representation of UCRAN that encodes the non-euclidean topology of the network along with its euclidean features, effectively encapsulating the wireless domain knowledge of the network configuration. Our experiments show that the deep graph reinforcement learning can learn to optimize S-zone sizes with 15% fewer training episodes in comparison to the legacy neural-network-based reinforcement learning, hence demonstrating the advantage of network topology-awareness for artificial intelligence.

CHAPTER 1

Introduction

1.1 Motivation

The evolution of next-generation wireless networks anticipates serving an increasing number of users per geographic area, leading the wireless research towards ultra-dense networks with a higher area density of base stations [1, 2, 3, 4, 5]. However, in ultra-dense networks, the overlapping area of coverage among closely located cells exasperates the problem of inter-cell interference. Additionally, as future user applications become increasingly diverse, encompassing new-age technologies such as virtual reality, online surgery, and autonomous driving alongside low-resource intensive activities like internet browsing and streaming, the conventional one-size-fits-all configuration may lead to either a degradation of service quality or inefficient resource utilization in the face of widely varying user requirements.

User-centric Radio Access Networks (UCRAN) [3, 6, 7] address these challenges by introducing user-centric virtual cells which are non-overlapping; thus mitigating inter-cell interference, and allow dynamic resizing; hence accommodating the varying quality of service needs of different users. UCRAN consists of a low density of high coverage Control Base Stations (CBS) with a high density of Data Base Stations (DBS). User applications are divided into Quality of service (QoS) categories based on the user service requirements such as throughput, latency and reliability. For each UE, a circular virtual cell centered on the UE, defines the area of service (S-zone) in which, for each Transmission time interval (TTI), 1) only the highest priority UE is scheduled 2) only one DBS is active for the UE while other DBSs within the S-zone are powered off. This scheduling scheme mitigates the problem of down-link interference among the cells in close proximity. In addition, the S-zone radii per UE category is elastically varied allowing the system to optimize for heterogeneous UE requirements resulting in increase in energy and area spectral efficiency of the network.

In realistic cellular networks with a large number of user QoS categories, the optimal choice of S-zone size for all user categories becomes a challenging problem. This in conjunction with the joint optimization of multiple key performance indicators and variability of the cellular environment, such as user mobility and fading, further complicates the optimization of S-zone for all user categories. To that end, reinforcement learning-assisted solutions are proposed to find the Pareto-optimal front; i.e., the S-zone radii per UE category which jointly optimize the multiple KPIs of network while remaining robust to user mobility and environmental changes [3, 6].

Even though the proposed reinforcement learning-assisted solutions in current literature provide a potential solution to optimize the sizes of S-zones, these solutions do not take into account a key element of wireless networks, that is, the network topology. A wireless network topology defines the spatial arrangement and interconnection of network devices, outlining how they communicate or interfere with each other, thus embodying essential wireless domain knowledge within the network structure and relationships among its nodes. However, unlike Euclidean spaces with fixed dimensions, the wireless network topology lacks a specific order, can change dynamically with variable distances and adjacency criteria among communication devices. This non-Euclidean characteristic [8] is crucial for accurately modeling the complex and ever-changing nature of wireless networks. Notably, this non-Euclidean nature challenges the applicability of multi-layer perceptron (MLP) or convolutional neural network (CNN) based reinforcement learning (RL) agents, which are tailored for Euclidean data and struggle to fully capture the complexity of wireless network topology. Therefore, the prior works [3, 6] on UCRAN, which employ RL with classical neural networks agents are limited to only learning from the network features/parameters, while not taking advantage of the domain knowledge latent in UCRAN topology.

Recently, Graph Neural Networks (GNN) have emerged as a tool to learn from non-Euclidean data. By representing data as graphs, GNNs adeptly capture the inherent characteristics of non-Euclidean data, such as non-fixed adjacency and dynamic order. GNNs apply convolution over graph nodes, utilizing back-propagation to learn through gradient descent—mirroring the processes seen in MLPs and CNNs designed for Euclidean data such as images. This progress paves the way for augmenting reinforcement learning by seamlessly integrating GNNs to enable RL to learn with a topology-aware graphical representation of wireless networks.

Building on this context, we present a method for conceptualizing UCRAN architecture as a **D**eep **Wi**reless **N**etwork graph (DeepWiN), where network entities, parameters, and interactions are respectively represented as nodes, node features, and the graph structure (edges and edge weights). This *deep* network representation encodes the wireless domain knowledge, in the form of network topology along with network parameters, as opposed to *shallow* representations in prior studies [3, 6], which only consisted of the network parameters. Employing graph convolutions, we embed the DeepWiN graph state for Reinforcement learning, introducing a variation of the Soft actor-critic (SAC) algorithm designed with graph convolution layers in the actor, critic, and value networks. This unified algorithm, **D**eep **G**raph **R**einforcement **L**earning (DGRL), amalgamating recent advancements in graph neural networks and reinforcement learning, demonstrates significantly faster learning compared to RL with conventional neural networks. Specifically, DGRL achieves convergence with 15% fewer training episodes than LegacyRL, underscoring that the DeepWiN graph imparts a more profound understanding of the UCRAN network to the RL agent than a shallow network state can.

1.2 Related Work

The contemporary literature on this study can be divided into three categories: RL applications for wireless optimization [3, 6, 9, 10], GNN applications of graphical models of wireless networks [11, 12, 13], RL and GNN combined frameworks for distributed and user-centric networks such as UCRAN [14].

Application of reinforcement learning for wireless network optimization is an active area of research. In [9] various single-agent and multi-agent DRL methods are applied to optimize the cell edge user performance. However, the state is limited to a Euclidean representation of the wireless network which does not consider the non-Euclidean network topology. Furthermore, the action space, although continuous, is limited to a discretized form such that the agent can only choose a fixed increase or decrease in the network parameters instead of selecting from continuous parameter ranges. Allowing the agent to only take constant length steps within the action space, increases the number of steps it can take for an agent to reach the optimal state. [10] further emphasizes the need of data driven RL to optimize wireless network parameters. Deep deterministic policy gradient is used to maximize coverage and minimize interference while jointly optimizing the transmit power and down-tilt. However, the study does not address the joint optimization of various coverage and capacity parameters in ultra-dense networks expected in 6G and beyond. [3, 6, 7] lay down the framework of UCRAN for ultra-dense user-centric network and employ Q Learning to optimize S-zone sizes for multiple user categories based on criticality of their application. Though, like [9] and [10], these studies do not explore graphical topology of the network and therefore fail to incorporate the UCRAN's configuration and domain knowledge into the network state at the loss of state depth and generalization.

[11, 12] give insight into considerations for modeling various wireless network scenarios as graphs and the design of graph neural network for specific research problems in wireless domain. However, the studies do not address user-centric networks particularly.

More recently, [14] laid down a framework of a GNN-RL model for connection management in Open-RAN (O-RAN). O-RAN is modelled as a heterogeneous graph with users and radio units (RUs) as nodes and the wireless links among them defining the edges. In this study, the number of cells is limited to 6 with only 50 users. While there is a discussion on scaling to a more realistic network with hundreds of cells, the study only focuses on the aforementioned 6 cell network.

1.3 Research Objectives

Based on the discussed motivation and the corresponding literature study, the objectives of this research are outlined as follows:

1. **User-Centric Radio Access Networks**: Explore and develop techniques within UCRAN to mitigate interference issues by introducing non-overlapping, dynamically resizable virtual cells, considering diverse user applications and service requirements.

2. **Reinforcement Learning for Network Optimization**: Develop and evaluate reinforcement learning-assisted solutions to optimize the size of S-zones in realistic cellular networks, considering multiple key performance indicators (KPIs), diverse user QoS categories and accounting for factors like user mobility and environmental changes.

3. **Wireless Domain Knowledge-Aware AI**: Investigate the impact of wireless network topology on the optimization process, emphasizing the spatial arrangement and interconnection of network devices, and evaluate the limitations of existing reinforcement learning approaches in capturing this non-Euclidean characteristic.

4. **Integration of Graph Convolution with Reinforcement Learning**: Identify the approach and challenges in enabling Reinforcement learning to learn from a graphical state.

1.4 Contributions

The contributions of this thesis can be summarized as follows:

1. Unveiled an innovative approach for modeling the UCRAN topology alongside network parameters, encapsulating this insight into a Deep Wireless Network (DeepWiN) graph. We highlight the distinctive rationale guiding our methodology, to be used as a roadmap for modeling various wireless networks as graphs allowing researchers to take advantage of graph neural networks.

2. Modified a LegacyRL algorithm, Soft Actor Critic, to incorporate graph convolution such that the amalgamated graph neural network and SAC framework can learn to embed and interact with the UCRAN graphical state. As a result, we develop an Deep Graph Reinforcement Learning framework (DGRL) which opens research opportunities for multitude of domains where graphical representation is crucial such as social media, traffic forecasting, networking, drug discovery etc. [8, 15].

3. Conducted a comparative analysis between LegacyRL and DGRL on training for S-zone optimization, showcasing that with a topology-aware graph representation, RL learns with a remarkable 15% reduction in episodes compared to a shallow representation lacking topology awareness.

1.5 Articles Currently Under Review for Publication

1. M.S. Riaz, **M. Shaukat**, T. Saeed, A. Ijaz, H.N. Qureshi, I. Posokhova, I. Sadiq, A. Rizwan, A. Imran, " iPREDICT: AI Enabled Proactive Pandemic Prediction using Biosensing Wearable Devices," in Computers in Biology and Medicine Journal (submitted Oct 25, 2023)

2. **M. Shaukat**, S. K. Kasi, A. Imran, "DeepWiN: Deep Graph Reinforcement Learning for User-Centric Radio Access Networks Automation," (Under Co-authors Review)

Academic Awards

1. Generation Google Scholarship (North America) 2023

1.6 Organization

The rest of this thesis is organized as follows: Chapter 2 presents the system model for UCRAN along with the algorithm used for scheduling users within the network. We then outline the mathematical framework of major network parameters and key performance indicators considered in this study, followed by the multi-objective optimization problem formulation with the user category related as well as overall networks related performance indicators. In Chapter 3, the components of the proposed solution are discussed including an explanation of non-Euclidean data, modeling of UCRAN as graph, graph neural networks, soft actor critic and finally the deep graph reinforcement learning framework. Chapter 4 discussed the details of the experimentation and results of DGRL and LegacyRL training and evaluation. Lastly, Chapter 5 concludes this study with the summarizing of our findings while Chapter 6 discusses future research directions.

CHAPTER 2

System Model

2.1 UCRAN Architecture

UCRAN architecture is suited for user-focused ultra dense wireless networks. The architecture, shown in Fig. 2.1, comprises of low density of Control base stations (CBS) and a high density of Data base stations (DBS). The DBSs are connected to the pool of Base band units (BBUs) at CBS through an optical fiber network. The users in the network are classified into categories based on criticality of their applications depending on the latency, throughput and reliability requirements. The pivotal feature of this network is the S-zone, a virtual circular cell centered on each user (UE) such that the UE is connected to only one of the DBSs within its S-zone. The scheduled UEs and corresponding connected DBSs are selected based on a scheduling algorithm as follows. In each TTI, a UE is scheduled if and only if it belongs to the highest priority category within its neighborhood where the neighborhood is defined by the respective S-zone of each UE. Similarly, in each S-zone only one DBS providing the maximum single strength to the UE is activated, while all others are powered off. Each UE category can have a unique S-zone radius within a defined range.

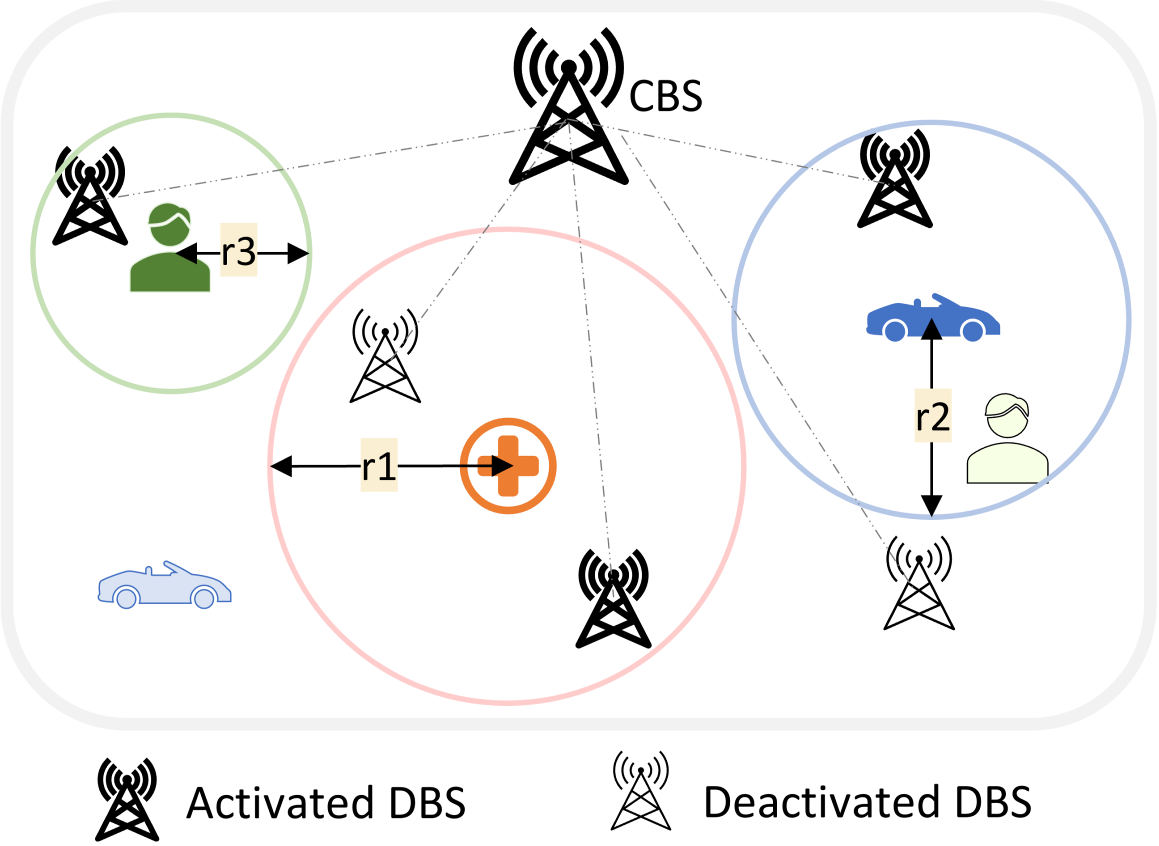

Fig. 2.1: UCRAN architecture.
Three UE quality of service categories are shown with their unique S-zone radii. The priority of each category is based on its application in order of i) medical devices (red), ii) autonomous vehicles (blue), iii) internet browsing users (green). In each S-zone, only the highest priority UE is scheduled. Moreover, in each S-zone, only the DBU providing the highest signal strength to the scheduled UE is powered on.

2.2 Simulation Model

We designed a comprehensive simulation model of UCRAN for this study. The details of which are as follows: The UEs and DBSs are randomly distributed through two homogeneous Poisson point processes Π_{DBS} and Π_{UE} with densities λ_{UE} and λ_{DBS} respectively. All UEs are divided randomly into C QoS categories with uniform probability such that each category has a specific throughput, latency and reliability requirement as listed in Table 2.1. Each category is then assigned a priority-level based on these requirements. The communication channel between a between an arbitrary UE $x \in \Pi_{UE}$ and activated DBS $i \in \Pi'_{DBS}$ experiences both large-scale and small-scale fading given by

hl^{-PLE}, where h is an exponential random distribution with a unit mean, and l_{xi}. represents the propagation distance between x and i, $PLE < 2$ is the pathloss exponent. UE and DBS have a single antenna and the transmission power of all DBSs is equal. Each scheduled user is served by a DBS providing the highest channel gain within an S-zone of radius r_c whose SINR (Γ_x) is given as:

$$\Gamma_x = \frac{h_{xi} l_{xi}^{-PLE}}{\sum\limits_{j \in \Pi'_{DBS}} h_{xj} l_{xj}^{-PLE} + n_o}, \tag{2.1}$$

where $i \neq j$ and n_o is additive Gaussian white noise.

2.3 Network Key Performance Indicators

In this study we jointly optimize multiple key performance indicators (KPI) of UCRAN namely area spectral efficiency, network energy efficiency, user service rate, and reliability satisfaction.

2.3.1 Area Spectral Efficiency

The area spectral efficiency refers to the amount of information that can be transmitted from a DBS per unit bandwidth channel per unit area to a UE, which can be defined as follows for each QoS category c:

$$\mathsf{A}_c = \frac{\sum\limits_{x \in N_c} \log_2(1 + \Gamma_x)}{\mathring{A}}, \tag{2.2}$$

where N_c is the set of UEs belonging to QoS category c, and $\mathring{A}$ is the target area considered in the simulations model.

The size of the S-zone in the QoS category has a pronounced impact on area spectral efficiency [7, 16]. In essence, enlarging the S-zone diminishes the scheduling ratio of UEs. Conversely, reducing the S-zone size enhances SINR which is caused by high the number of neighboring interfering DBSs. Striking a balance between these conflicting influences is essential to determine the optimal S-zone size, thereby maximizing achievable area spectral efficiency. Achieving this optimization necessitates intelligent real-time adjustments to simultaneously fine-tune the S-zone sizes for multiple QoS categories.

2.3.2 Energy Efficiency

According to [7, 17, 18], the network-wide energy efficiency is defined as the ratio of area spectral efficiency and total power consumed for all scheduled UE's. The power consumption model, drawing inspiration from project Earth [19], characterizes the power consumption of CBS and DBSs through a linear combination of fixed and load-dependent power components. Given that energy efficiency is gauged on a network scale, the cumulative power consumption values are summed across all scheduled users. Mathematically, the total power consumption is calculated as follows:

$$P = \lambda_{DBS} P_f + \lambda'_{DBS} \Delta_{DBS} P_{DBS} + \lambda'_{UE} (\Delta_{UE} P_{UE} + P_{disc}), \tag{2.3}$$

where λ_{DBS} is the density of all deployed DBSs, λ'_{DBS} is the density of activated DBSs, λ'_{UE} is the density of scheduled UEs, P_f is the fixed DBS power consumption required for DBS to operate in listening mode, P_{DBS} is the DBS transmission power, Δ_{DBS} is the radio frequency component power at DBS, P_{UE} is the UE transmission power, Δ_{UE} is the radio frequency component power at UE, P_{disc} is the power required at UE for discovery of the DBS with the highest channel gain.

The energy efficiency therefore can be given as:

$$\mathsf{E} = \frac{\text{Å} \times \sum\limits_{c \in C} \mathsf{A}_c}{P}, \tag{2.4}$$

In a cellular DBS, radio frequency components and data transmission account for the majority of total power consumption [20]. Significant energy savings can be achieved by dynamically activating DBSs, especially in dense deployments. The direct connection between energy efficiency and area spectral efficiency emphasizes that the S-zone size of QoS categories plays a crucial role in impacting network energy efficiency. On an intuitive level, enlarging the S-zone size results in a reduction of activated DBSs, leading to a decrease in average power consumption. The diverging trends between area spectral efficiency and power consumption pose a key design question: What S-zone size should be chosen for QoS categories to optimize network-wide energy efficiency?

2.3.3 UE Service Rate

The UEs' heterogeneous latency requirements necessitate scheduling more UEs within each TTI while meeting UE quality of experience requirements. The mean UE service rate (user service rate) for any QoS category c can be calculated as:

$$\mathsf{U}_c = \frac{\lambda_{UE_c}^{service}}{\lambda_{UE_c}}, \tag{2.5}$$

where, λ_{UE_c} is the density of all UEs belonging to QoS category c and $\lambda_{UE_c}^{service}$ is the density of UEs belonging to QoS category c whose minimum throughput requirement is met.

The S-zone size of QoS categories influences the user service rate in two different ways. A decrease in the S-zone size leads to the scheduling of more users. However, decreasing the S-zone size also increases the average distance between UE and DBS, thus, affecting the average SINR. Due to these conflicting results with the change in S-zone size, optimizing user service rate requires intelligent optimization of S-zone sizes for different QoS categories.

2.3.4 Reliability Satisfaction

The concept of reliability, as per 3GPP, refers to the ability to transmit a specified amount of traffic from the application server to UE within the required time constraints with a high probability of success [21]. To quantify this reliability, we assess reliability satisfaction by calculating the weighted sum of the average probability of correctly received packets for UEs in each vertical. The Block Error Ratio (BLER) serves as a metric, representing the ratio of erroneously received blocks to the total number of blocks sent at each TTI. UEs report a Channel Quality Indicator (CQI), which, in conjunction with a Signal-to-Interference-plus-Noise Ratio (SINR) defined in Eq.2.1, is utilized to map the CQI to a corresponding BLER value [22]. Mathematically,

$$S_c = \mathbb{E}_{\tau}\left[\frac{\sum_{i=1}^{N'_c}(1-\beta_{ci\tau})}{|N'_c|}\right], \tag{2.6}$$

where $|N'_c|$ represent the number of scheduled UEs belonging to category c, $\beta_{cj\tau}$ represents the BLER at UE i belonging to category c, $\mathbb{E}_{\tau}[.]$ represents averaging over several TTIs.

2.3.5 Multi-objective Optimization

Hitherto, the above definition of KPIs demonstrate the need for optimizing S-zone size of QoS categories to maximize area spectral efficiency (ASE), energy efficiency (EE), user service rate (SR) and reliability satisfaction rate (RS) individually. The challenge from a network operator's perspective is that all these KPIs should be optimized simultaneously, leading to a Pareto-optimal tradeoff between them. To account for this tradeoff, this study defines the multi-objective optimization problem as follows:

$$\max_{r_c \in [r_{\min}, r_{\max}]} \left(a \sum_{c \in C} w_c A_c + b \sum_{c \in C} \dot{w}_c U_c + y \sum_{c \in C} \ddot{w}_c S_c + zE \right) \text{s.t.} \quad a + b + y + z = 1 \tag{2.7}$$

$w_c \geq 0$, $\forall i$ and $\sum_{c=1}^{C} w_c \leq 1$ are network operator-defined weights assigned to prioritize the area spectral efficiency of specific QoS categories. Similarly, $\dot{w}_c$ and $\ddot{w}_c$ are weights for user service rate and reliability satisfaction. a, b, y, z specify the weights of each cumulative KPI in the multi-objective optimization.

Note that all KPIs are normalized between 0 and 1 by using a pseudo min and max KPI value which are drawn by a brute force sweep of the search space. The search space is however significantly large: with $[r_{min}, r_{max}] = [10, 40]$, there are $30^3 = 27,000$ combinations of S-zone radii. So for a pseudo-brute force search we randomly sample 10 value from S-zone range of each QoS category, reducing the search space to 1000 states.

Table 2.1: UE QoS categories: requirements and priority

Throughput	***Latency***	***Reliability***	***Priority***
1000	50	50	I
10	2	5	II
99.9	99.999	99.99	III

Table 2.2: UCRAN simulation and DeepWiN graph parameters

Parameter Name	***Parameter Value***
UE density (λ_{UE}), DBS Density (λ_{DBS})	$10^3/km^2$,
Range of min. and max. S-zone radius $[r_{min}, r_{max}]$	[10, 40]
Path-loss exponent	3
Number of UE categories (C)	3
Simulation region	1 km^2
Nominal* RSRP	-150
Nominal SINR	-20
Nominal Throughput	0
Edge weight factor (χ)	10

*Minimum/Null value for this simulation

CHAPTER 3

Proposed Solution

This chapter explains the building blocks of the proposed solution.

3.1 Euclidean vs. Non-Euclidean Data

The premise or our study is to recognize that wireless networks are non-Euclidean in nature. Fig. 3.1 shows the difference between Euclidean and non-Euclidean data structures. Euclidean data has fixed and uniform dimensions. For example, in an image which is a prime example of euclidean data, each pixel is adjacent to a fixed number of neighbouring pixels which are located at a pre-determined euclidean distance. In contrast, non-euclidean data can change its dimensions and does not follow any specific order. For instance, in a wireless network a mobile user can be adjacent to a variable number of other users. In fact, the definition of adjacency is variable as well, as it can be based on their geographic distance, interference or even similarity of characteristics. In addition, as the users move around the network, the number of adjacent users can change every second. Hence, the topology of a wireless system is more effectively modelled in a non-Euclidean graph in comparison to vector or matrix which approximates the network to a euclidean structure, hence ignoring its topological information.

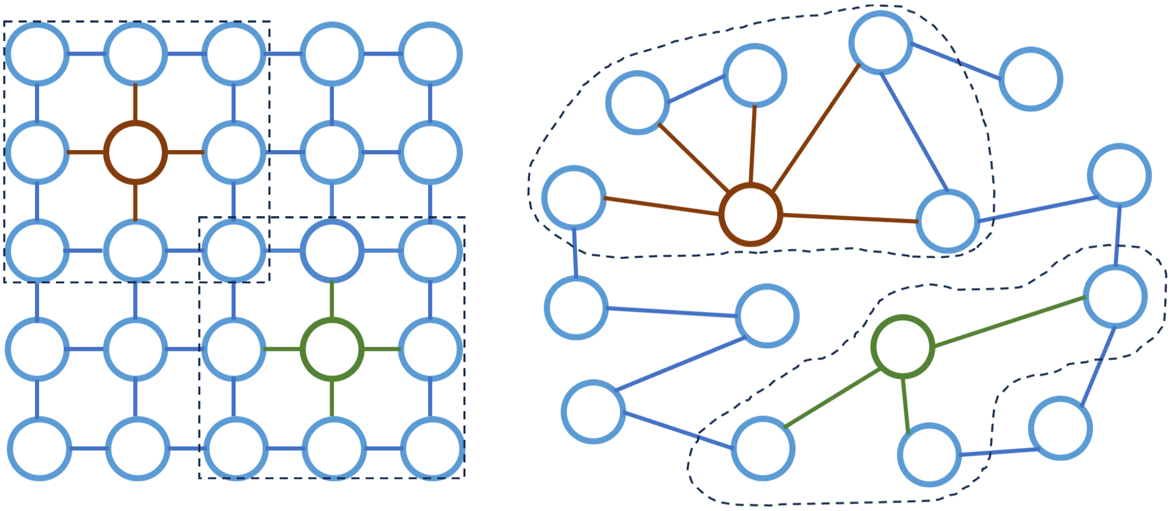

Fig. 3.1: Euclidean vs. non-euclidean data. In a euclidean structure, the number of neighbors of each data-point are uniform across the data. In non-euclidean data, the definition of what comprises as the neighborhood is highly flexible.

3.2 Deep Wireless Network Graphical Representation: DeepWiN

Wireless networks are inherently non-euclidean graph-structured. Graphs provide a natural and efficient representation of network topology by capturing the complex connectivity relationships among network elements. In a graph, nodes can represent wireless devices, access points, and other network components, while edges represent the wireless communication links or interference links between them. This graphical representation simplifies the analysis of network topology, enables the application of graph algorithms for tasks like routing and resource allocation, facilitates network design and optimization, and aids in visualizing and understanding the structure of wireless networks, making it an essential tool for network planning, management, and troubleshooting [11, 12].

Therefore, for UCRAN, we design a Deep wireless network graphical representation (DeepWiN) Fig. 3.2 of the system using the following approach. Each UE serves as one node in the graph while DBSs are not explicitly included as nodes. This is because UCRAN is a user-centric system where the behaviour of each DBS is dependent on the scheduling of UEs in its proximity. Secondly, while DBSs are geographically static, UEs move in each time interval varying the spatial topology of the system. Thirdly, DBSs and UEs consist of different features and therefore a graph involving both the entities is a heterogeneous graph containing different types of nodes and edges as opposed to a homogeneous graph. Currently, most GNNs are designed for homogeneous graphs. Therefore, an exploration of heterogeneous DeepWiN graph is aimed for future studies with the progress on heterogeneous GNN research [8, 23].

For each node, a set of UE parameters is associated as node feature vector. The UE parameters quantify the status of the UE in the network. In DeepWiN, we include the following network parameters: RSRP, SINR, throughput, Probability of coverage, spectral efficiency, BLER. In addition, we add two binary node features indicating if a UE is scheduled and/or served. Lastly, the UE category is indicated as $c < C$ and the current S-zone radius of the respective UE category is also appended as a node feature. So, in total each node feature vector consists of the above mentioned 11 parameters. It is important to note that while GNNs are independent to the change in number of nodes in a graph, the dimensions of feature vector in each node must remain consistent across the graph data set. Therefore, even for UEs which are non-scheduled and therefore their parameters such as RSRP, SINR are not measured, a null value of these parameters is still required in the UE feature vector. However, the current graph convolution models available do not support missing, null or undefined values. To circumvent this challenge, we impute the feature vector with the nominal value of each parameter as a pseudo-null. The pseudo-null values adopted for our UCRAN simulation are as listed in Table 2.2

Finally, in DeepWiN graph, all the edges are non-directional and are defined as: an edge exists between two UEs x_i and x_j, if both the UEs are scheduled and the euclidean distance between them is $d_{i,j} < 2 * r_{max}$ where r_{max} is the maximum possible S-zone radius in the network Table 2.2. The edge signifies the interference effect among UEs and therefore only exist between scheduled UEs which are in close proximity. The edge weight is a real numbered value inversely proportional to the UE distance. Hence, for each edge there is a weight $e_{i,j} = \chi / d_{i,j}$, where χ is a factor that controls the effect of edges in the DeepWiN graph. Higher the χ, higher will be effect of the edges, and resultant the network topology, on the RL agent training. If χ is 0, the effect of edges is diminished and the DeepWiN becomes equivalent to the shallow state with no network topology.

In summary, DeepWiN is a graph G_E^N with N nodes and E edges which represents the UCRAN parameters and topology of UCRAN environment such that each UE is a node $x \in N$ consisting for k features $f_1, f_2, \ldots, f_k$ of each UE while each edge $e \in E$ is defined by the distance between two nodes with the edge weight inversely proportional to the distance among nodes.

The graph considered in this study, based on the simulation environment listed in Table 2.2, consists of 284 nodes, each with 11 features while the number of edges vary from graph to graph due to UE mobility and scheduling changes.

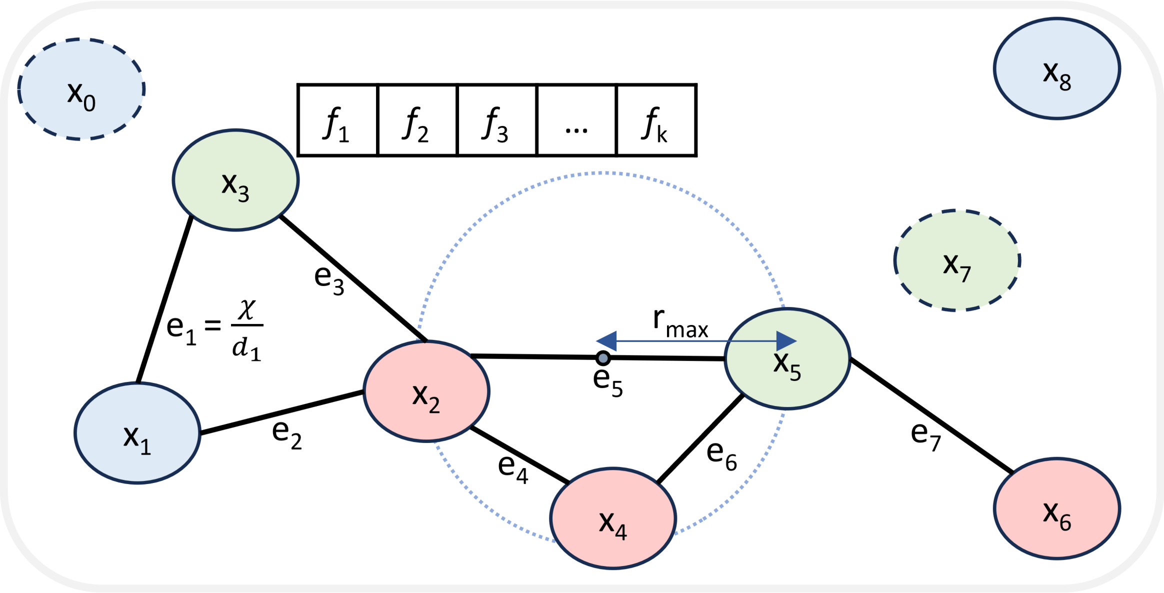

Fig. 3.2: UCRAN architecture as DeepWiN graph.
Nodes 1-6 are connected with each other with edges 1-7 as these nodes are scheduled and lie withing $2 * r_{max}$ distance from each other. Node 0 and 7 are not scheduled, so, no edge extends from them while node 9, although scheduled, is outside the distance threshold for an edge. Each node 1-8 consists of a feature vector of length k and each edge has an edge weight e, inversely proportional to the distance between the nodes.

3.3 Graph Neural Network

Graph Neural Networks (GNNs) [24] are a class of neural network models designed for processing and learning from non-euclidean raph-structured data. They leverage the connections between nodes in a graph to perform convolutions and capture node relationships. GNNs iteratively update node representations by aggregating information from their neighbors, allowing them to learn expressive node embeddings that incorporate both local and global graph information. These embeddings can be used for various tasks, such as node classification, link prediction, and graph classification, making GNNs a powerful framework for analyzing and understanding graph data.

The building block of GNNs is graph convolution as explained here:

Graph Convolution Layer:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{3.1}$$

Where:

$H^{(l)}$ - Node features at layer l

$\hat{A}$ - Normalized adjacency matrix

$\hat{D}$ - Degree matrix of $\hat{A}$

$W^{(l)}$ - Learnable weight matrix at layer l

$\sigma(\cdot)$ - Activation function (e.g., ReLU)

In essence, the information from neighboring nodes in the graph (represented by $\hat{A}$ is propagated to update the node features $H^{(l)}$ at the next layer $l+1$ The learnable weights $W^{(l)}$ control how this information is combined, and the activation function $\sigma(.)$ introduces non-linearity into the transformation. This process is central to how GNNs operate, allowing them to capture graph-based relationships and perform tasks like node classification or graph-level classification.

GNNs are scalable to different graph sizes and configurations. Therefore, in case of wireless networks, where the network topology and the resultant graph changes with the user mobility, environmental factors, and network configuration, GNNs are able to perform convolutions on such dynamic data when UEs of DBSs in a network deployment vary in number and location.

3.4 Soft Actor Critic

Soft Actor-Critic (SAC) [25] is a state-of-the-art off-policy reinforcement learning algorithm designed for training agents in environments with continuous action spaces. It combines the actor-critic architecture with entropy regularization, aiming to maximize not only expected rewards but also the entropy of the policy, promoting more effective exploration. SAC employs target networks, deterministic policy improvement, and a soft Bellman backup to ensure stability during training. This algorithm has proven to be highly effective in a wide range of continuous control tasks, making it a popular choice for developing autonomous systems and robotics applications due to its capacity for efficient exploration and policy optimization.

In the proposed solution, SAC algorithm has been implemented as following:

- **Environment State**: The Graphical representation, G_E^N of the UCRAN network environment (DeepWiN) with N nodes and E edges.
- **Action**: A continuous action space comprising of S-zone radius for each UE QoS category c such that $r_c \in [r_{\min}, r_{\max}]$.
- **Reward**: Multi-objective linear combination of UCRAN KPIs in Eq. 2.7

3.5 Deep Graph Reinforcement Learning

In DGRL, we implement the actor, critic and value networks of SAC as GNNs. Each network consists of two graph convolution layers with 256 channels each, followed by a global mean pooling layer which aggregates the embedding of all graph nodes into one vector of length 256. The embedding is then input to two linear layers with 256 neurons. each. After every graph convolution and linear layer, a Rectified linear unit activation is applied. In actor network, the final two linear layers output the mean μ and standard deviation σ for a Guassian distribution from which the actions are sampled. In value and critic networks, the final linear layers outputs q-value and state-value respectively.

An important consideration in the case of the critic network is to append the action along with state as input. The state in case of DGRL is a graph structure DeepWin, which is to be appended with actions which are in a vector. For this design question, we considered two approaches:

1. Append the action vector to the feature vector of each node.

2. Append the action to the aggregated output of the global mean pooling layer.

The later approach was taken in DGRL as in this approach the action vector is appended only once to the cumulative embedding of the graph after convolution. Whereas the prior introduces redundancy in graph data because the same action is appended n times, i.e., to each node.

Table 3.1: DGRL training parameters

Parameter Name	***Parameter Value***
Actor learning rate (α)	0.0003
Critic, Value network learning rate (β)	0.0003
Discount factor (γ)	0.99
Target smoothing coefficient (τ)	0.005
Maximum memory buffer size	10^6
Mini-batch size	256
Maximum episode length (T_{max})	256
Number of actions	3
Reward scale (λ)	30
Multi-KPI reward coefficients (a, b, c, d)	0.25,0.25,0.25,0.25
UE category KPI weights (w_1, w_2, w_3)*	0.34,0.33,0.33

*Same KPI weights for $\dot{w}$ and $\ddot{w}$

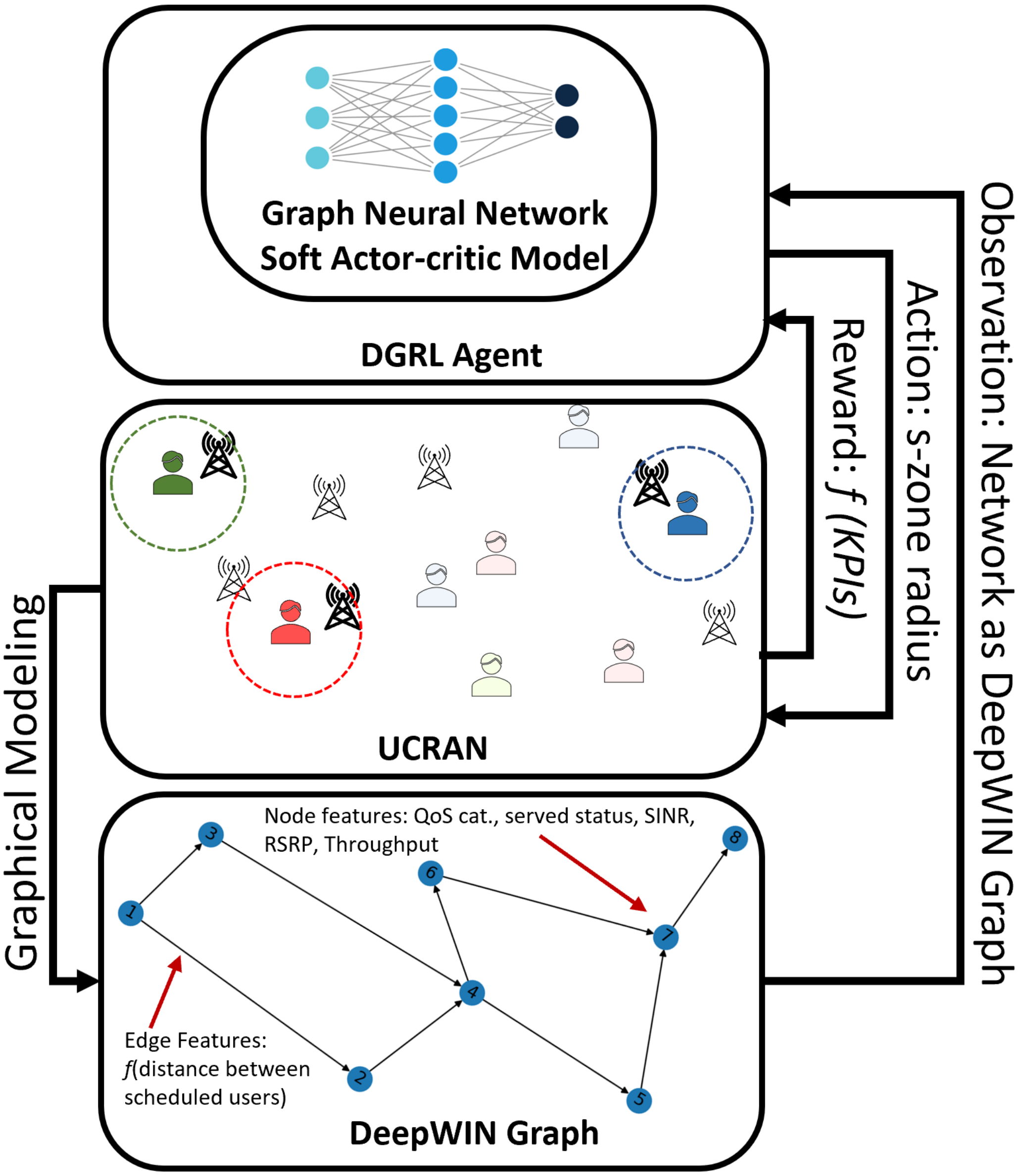
Graph Neural Network
Soft Actor-critic Model
DGRL Agent
UCRAN
Node features: QoS cat., served status, SINR, RSRP, Throughput
Edge Features: *f*(distance between scheduled users)
DeepWIN Graph
1
2
3
4
5
6
7
8
Graphical Modeling
Reward: *f (KPIs)*
Action: s-zone radius
Observation: Network as DeepWIN Graph

Fig. 3.3: Deep graph reinforcement learning framework

Algorithm 1 Soft Actor-Critic with Graph Neural Network (DGRL)

```
Initialize GNN-based actor network, critic network, and target networks
Initialize replay buffer $\mathcal{D}$
Initialize hyperparameters $\alpha$, $\beta$, $\gamma$, $\tau$, batch size, etc
for episode = 1 to $N_{\text{episodes}}$ do
    Sample initial state $s$ (UCRAN graph)
    for step = 1 to $T_{\max}$ do
        Select action $a$ (S-zone radii) using the GNN-based actor network: $a \sim \pi(a|s)$
        Execute action $a$ in the environment, observe next state $s'$ (update S-zone radii) and reward $r$ (eq x)
        Store transition $(s, a, r, s')$ in $\mathcal{D}$
        if length of $\mathcal{D} >$ batch size then
            Sample a minibatch of transitions from $\mathcal{D}$
            Update the critic network parameters using the Temporal Difference error: $\phi \leftarrow \phi + \beta \cdot \nabla_{\phi}\text{MSE}(\phi)$
            Update the actor network parameters using the deterministic policy gradient: $\psi \leftarrow \psi + \alpha \cdot \nabla_{\psi} J(\psi)$
            Update the soft Q-network and target networks
        end if
    end for
end for
```

CHAPTER 4

Experiment Evaluation and Results

DGRL is trained with the SAC and GNN hyper-parameters listed in Table 3.1. In each episode, the radius r_c of the 3 S-zone categories is initialized randomly between r_{min} and r_{max}. UE locations are also initialized randomly by changing the seed for each episode while DBS locations remain static. The network environment state is converted into the DeepWiN graph, as explained in Section 3.2 which is input to the GNN actor, critic and value networks. The actor outputs a probability distribution with mean μ and standard deviation σ for a Gaussian distribution of each of the three actions i.e., selection of the S-zone radius from $[r_{min}, r_{max}]$ for each of the three UE categories. The agent applies the action sampled from this probability distribution to vary the S-zone radii of the UCRAN environment. Reward is calculated as Eq. 2.7 to optimize multiple UCRAN KPIs. The reward range lies between $[0, 1]$ as all the KPIs are normalized in this range. This cycle is repeated at each step in the episode, and the DeepWiN state, action, reward etc., are saved in the replay buffer to be used for training the GNN agent in SAC.

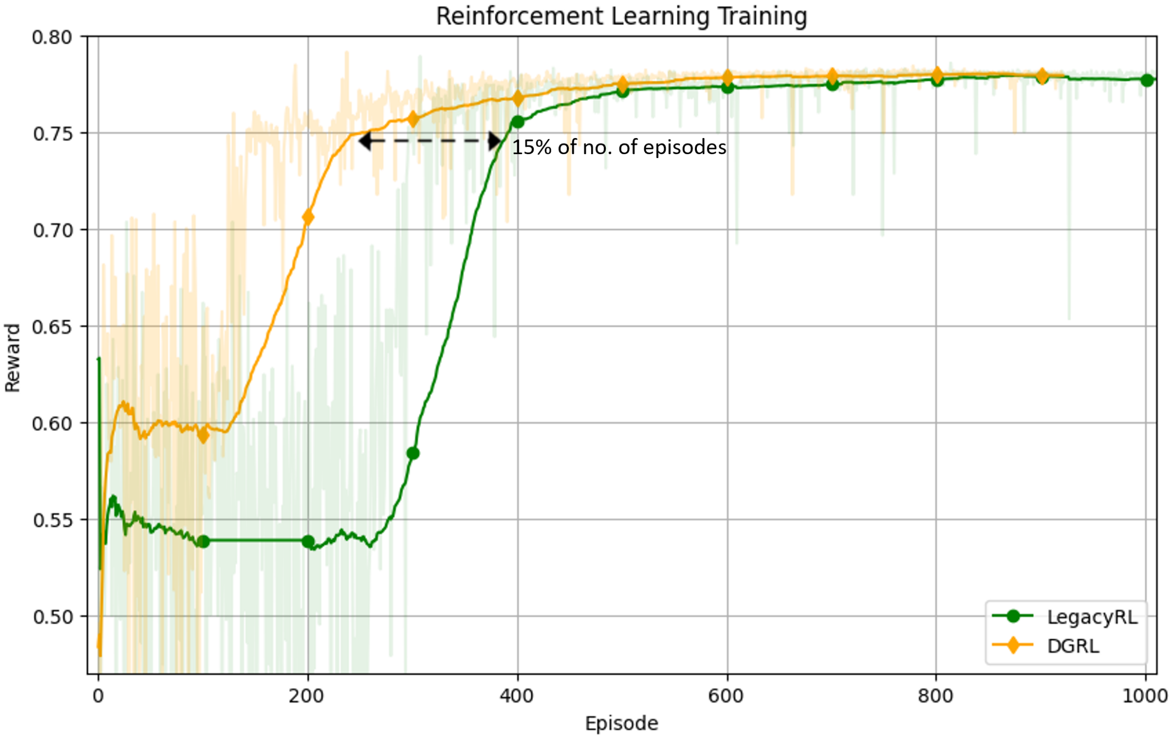

Fig. 4.1: Soft actor-critic training with DeepWiN graph state vs. shallow state. Former converges with 15% less number of episodes i.e., less number of training samples.

To compare the performance of the DeepWiN state and DGRL we train a baseline model with shallow network state and LegacyRL. All training hyper-parameters are kept the same, except that the graph convolution layers in the SAC actor, critic and value networks are replaced by linear neural network layers with the same number of channels. The state in the baseline model is a vectorized (Euclidean) approximation of the network. Shallow state consists of a vector of all the 11 UE parameters but the topology of the network is ignored as the edges and edge weights are not included. Training comparison of DGRL and LegacyRL in Fig. 4.1 shows that DGRL with deepWIN state converges with 15% less number of episodes that LegacyRL with shallow state.

In Fig. 4.2, we can see the trained RL models in action.

The UCRAN KPIs are jointly optimized to a Pareto optimal front during an episode. As the episode starts, the RS, EE, and ASE of the network is low but the trained DGRL and LegacyRL agents select the S-zone radii to optimize the KPIs. Then, during the episode, as fading and scheduling changes the network, the agent manages to dynamically select the S-zone radii to keep the KPIs at the Pareto optimal value.

As visible in Fig. 4.2, even though the DGRL is trained on 15% lesser episodes then the LegacyRL, after the training, they behave the same, i.e., they optimize the KPIs of the UCRAN network to upto 90% of their maximum capacity in the start of the episode, and then maintain the KPIs at this level despite changes in UCRAN environment.

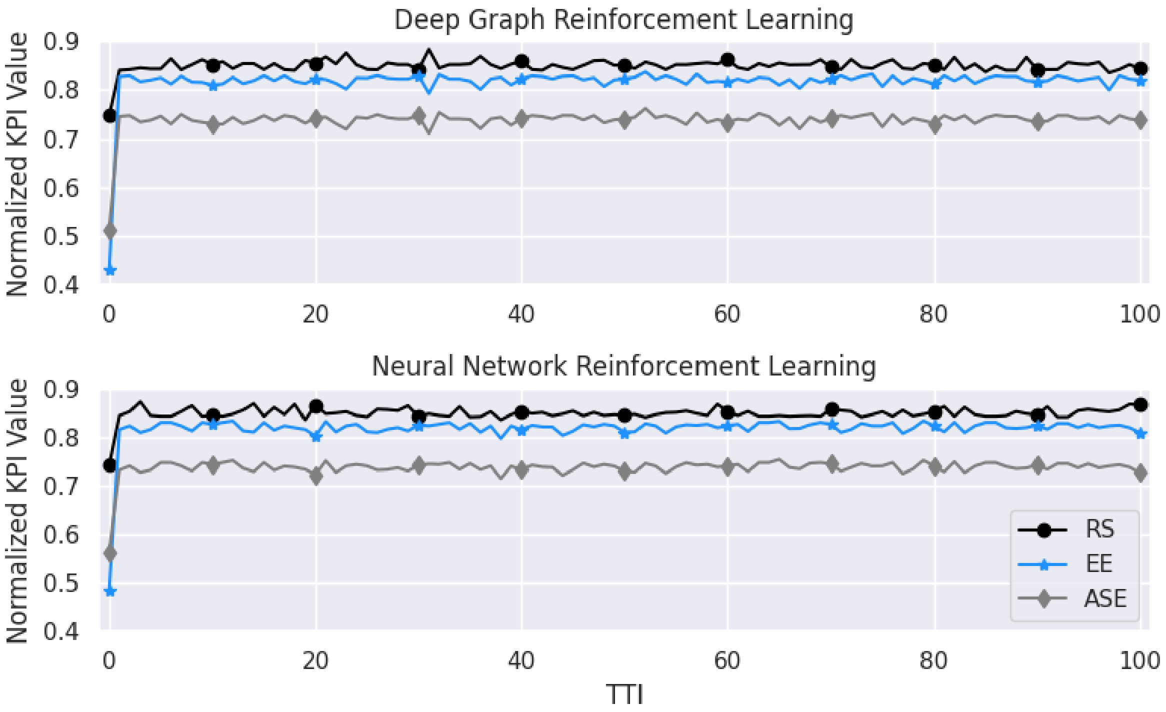

Fig. 4.2: UCRAN KPIs varying over an episode as the trained RL agents select the S-zone radii
In both DGRL and LegacyRL, the RL agent has learned to optimize and then maintain the network S-zone radii to the Pareto-optimal throughout the episode

In Fig. 4.3, the changes in the s-zone radii of three QoS categories can be observed to change over the episode. The trained DGRL agent is selecting s-zone radii between the range of r_{min} and r_{max} such that the system KPIs are jointly optimized over each TTI.

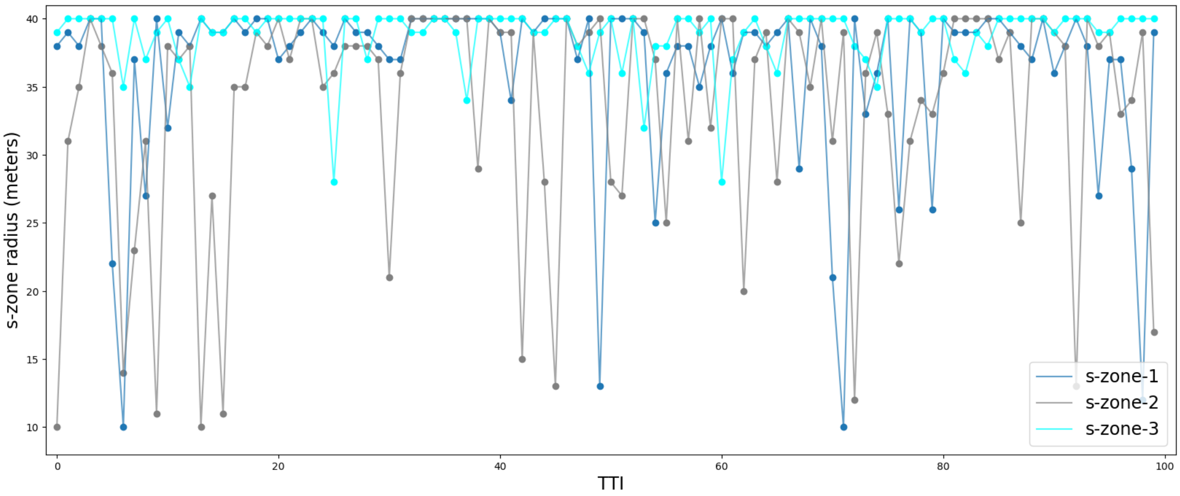

Fig. 4.3: S-zone radii of 3 UE categories selected by the trained DGRL agent over an episode

CHAPTER 5

Conclusion

Over the past few decades, AI has emerged as a powerful tool with diverse applications. Yet, deploying AI effectively in complex domains like wireless technology requires more than generic machine learning algorithms. The valuable domain knowledge amassed by experts and network operators through extensive experimentation and experience is a crucial resource and AI needs to adapt to incorporate this knowledge to efficiently solve the problems in wireless networks. Our study addresses this gap by enhancing one aspect of AI, Reinforcement Learning, through the incorporation of one facet of domain knowledge, network topology. With this adaptation, we show a visible improvement in the learning capability of AI to more speedily solve the challenge under consideration.

Our approach involves modeling wireless networks as graph structures, specifically in the context of UCRAN. We explain the design decisions taken in multiple challenges and show the effectiveness of the selected DeepWiN graph design through experiments. The approach, however focused on a wireless network, specifically UCRAN, can be translated to other domains with non-euclidean and graph data.

We further modify components of RL, particularly in Soft actor-critic (SAC), to facilitate learning from graphical input. This involves introducing convolution layers in the actor, critic, and value networks, adjusting state and action appending in the critic, and enabling the sampling of graph batches from a replay buffer. Our cross-domain framework, deep graph reinforcement learning, jointly trains the graph neural network and SAC.

DGRL is employed in this study to learn the behavior of UCRAN environment with users in three QoS categories prioritized based on their resources needs. DGRL learns to modify the key network parameter of S-zone radius for each QoS category to jointly optimize for network area spectral efficiency, energy efficiency, user service rate and reliability satisfaction rate.

In conclusion, our results demonstrate that DGRL, trained on the DeepWiN graph state, requires 15% fewer training episodes compared to LegacyRL trained on a shallow state of the UCRAN network. This study underscores the significance of incorporating wireless network domain knowledge, such as topology, into AI models for enhanced learning compared to generic and domain-oblivious network features.

CHAPTER 6

Future Work

6.0.1 Heterogeneous Graph Neural Networks

DeepWiN design proposed in this thesis is homogeneous; containing only one type of node and one type of edge. All UEs in UCRAN environment are modeled as nodes and their inter-distance is used to define the edges. However, the DBUs are not included in DeepWiN graph on the grounds of DBUs being immobile and UCRAN being user-centric as explained in Section 3.2. More importantly, incorporating DBUs requires the DeepWiN graph to be heterogeneous as the the number and type of features for DBUs are different from that of UEs. In addition, the different types of edges are needed to be including, such as UE-UE, UE-DBU, and DBU-DBU edges differing in their features and edge weights as well. This complex heterogeneous DeepWiN graph can provide a more holistic representation of the network and can facilitate reinforcement learning to learn faster.

A heterogeneous DeepWiN design will require significant modification and research to adapt DGRL as well. The current graph convolution layers implemented in the actor, critic and value network in DGRL are compatible to work with only homogeneous graphs. Therefore, the recent development in heterogeneous [23, 26, 27, 28] graph neural networks is a promising direction to explore for future.

6.0.2 Unsupervised and Semi-supervised GNN Training

In DGRL, we jointly train the graph neural networks in soft actor-critic. However, a number of studies show that pre-training the graph convolution layers before incorporating them in reinforcement learning models, can significantly improve training. In addition to supervised training [14, 29], GNNs can be pre-trained in an unsupervised and semi-supervised manner [30, 31, 32, 33, 34].

Many different techniques, including edge perturbation, node dropping, node feature masking, random walk sampling and diffusion are used to generate synthetic data from the input graphs. The GNN models are then trained with a contrastive learning loss using both the input and synthetic data. Another un-supervised training technique is used in GNN transformers and auto-encoders where the input graphs are reconstructed at the output enabling the models to learn embedding of the graphs. These generalizable embeddings are then used in downstream tasks.

Bibliography

[1] B. Shubyn, N. Lutsiv, O. Syrotynskyi, and R. Kolodii, "Deep learning based adaptive handover optimization for ultra-dense 5g mobile networks," *Proceedings - 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020*, pp. 869–872, 2 2020.

[2] O. Alamu, A. Gbenga-Ilori, M. Adelabu, A. Imoize, and O. Ladipo, "Energy efficiency techniques in ultra-dense wireless heterogeneous networks: An overview and outlook," *Engineering Science and Technology, an International Journal*, vol. 23, pp. 1308–1326, 12 2020.

[3] S. K. Kasi, U. S. Hashmi, S. Ekin, and A. Imran, "Learning-aided demand-driven elastic architecture for 6g beyond," *IEEE Vehicular Technology Conference*, vol. 2023-June, 2023.

[4] S. Andreev, V. Petrov, M. Dohler, and H. Yanikomeroglu, "Future of ultra-dense networks beyond 5g: Harnessing heterogeneous moving cells," *IEEE Communications Magazine*, vol. 57, pp. 66–92, 6 2019.

[5] V. Stoynov, V. Poulkov, Z. Valkova-Jarvis, G. Iliev, and P. Koleva, "Ultra-dense networks: Taxonomy and key performance indicators," *Symmetry 2023, Vol. 15, Page 2*, vol. 15, p. 2, 12 2022.

[6] S. K. Kasi, U. S. Hashmi, S. Ekin, A. Abu-Dayya, and A. Imran, "D-ran: A drl-based demand-driven elastic user-centric ran optimization for 6g beyond," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, pp. 130–145, 2 2023.

[7] U. S. Hashmi, S. A. R. Zaidi, and A. Imran, "User-centric cloud ran: An analytical framework for optimizing area spectral and energy efficiency," *IEEE Access*, vol. 6, pp. 19 859–19 875, 4 2018.

[8] F. Liang, C. Qian, W. Yu, D. Griffith, and N. Golmie, "Survey of graph neural networks and applications," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[9] K. Yang, C. Shen, and T. Liu, "Deep reinforcement learning based wireless network optimization: A comparative study," *IEEE INFOCOM 2020 - IEEE Conference*

on Computer Communications Workshops, INFOCOM WKSHPS 2020, pp. 1248–1253, 7 2020.

[10] R. M. Dreifuerst, S. Daulton, Y. Qian, P. Varkey, M. Balandat, S. Kasturia, A. Tomar, A. Yazdan, V. Ponnampalam, and R. W. Heath, "Optimizing coverage and capacity in cellular networks using machine learning," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2021-June, pp. 8138–8142, 10 2020.

[11] Y. Shen, J. Zhang, S. H. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Transactions on Wireless Communications*, vol. 22, pp. 3554–3569, 5 2023.

[12] S. He, S. Xiong, Y. Ou, J. Zhang, J. Wang, Y. Huang, and Y. Zhang, "An overview on the application of graph neural networks in wireless networks," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2547–2565, 7 2021.

[13] S. Zhang, B. Yin, W. Zhang, and Y. Cheng, "Topology aware deep learning for wireless network optimization," *IEEE Transactions on Wireless Communications*, vol. 21, pp. 9791–9805, 12 2019.

[14] O. Orhan, V. N. Swamy, T. Tetzlaff, M. Nassar, H. Nikopour, and S. Talwar, "Connection management xapp for o-ran ric: A graph neural network and reinforcement learning approach," *Proceedings - 20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021*, pp. 936–941, 10 2021.

[15] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 1 2020.

[16] S. K. Kasi, U. S. Hashmi, M. Nabeel, S. Ekin, and A. Imran, "Analysis of area spectral energy efficiency in a comp-enabled user-centric cloud ran," *IEEE Transactions on Green Communications and Networking*, vol. 5, pp. 1999–2015, 12 2021.

[17] U. S. Hashmi, A. Rudrapatna, Z. Zhao, M. Rozwadowski, J. Kang, R. Wuppalapati, and A. Imran, "Towards real-time user qoe assessment via machine learning on lte network data," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–7.

[18] L. Sboui, Z. Rezki, A. Sultan, and M.-S. Alouini, "A new relation between en-

ergy efficiency and spectral efficiency in wireless communications systems," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 168–174, 2019.

[19] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume *et al.*, "How much energy is needed to run a wireless network?" *IEEE Wireless Communications*, vol. 18, no. 5, pp. 40–49, 2011.

[20] H. Khaled, I. Ahmad, D. Habibi, and Q. V. Phung, "A green traffic steering solution for next generation communication networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 222–238, 2021.

[21] 3GPP TS 22.261, "Service Requirements for the 5G system." [Online]. Available: http://www.3gpp.org/DynaReport/22261.htm

[22] Forsk, "Forsk Atoll." [Online]. Available: https://www.forsk.com/atoll-overview/

[23] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," *The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020*, pp. 2704–2710, 3 2020.

[24] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, pp. 61–80, 1 2009.

[25] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *35th International Conference on Machine Learning, ICML 2018*, vol. 5, pp. 2976–2989, 1 2018.

[26] X. Wang, D. Bo, C. Shi, S. Fan, Y. Ye, and P. S. Yu, "A survey on heterogeneous graph embedding: Methods, techniques, applications and sources," *IEEE Transactions on Big Data*, vol. 9, pp. 415–436, 4 2023.

[27] X. Mo, Y. Xing, and C. Lv, "Heterogeneous edge-enhanced graph attention network for multi-agent trajectory prediction," *arXiv preprint arXiv:2106.07161*, 2021.

[28] Q. Li, Y. Shang, X. Qiao, and W. Dai, "Heterogeneous dynamic graph attention network," *Proceedings - 11th IEEE International Conference on Knowledge Graph,*

ICKG 2020, pp. 404–411, 8 2020.

[29] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. Songhori, S. Wang, Y. J. Lee, E. Johnson, O. Pathak, A. Nazi, J. Pak, A. Tong, K. Srinivasa, W. Hang, E. Tuncer, Q. V. Le, J. Laudon, R. Ho, R. Carpenter, and J. Dean, "A graph placement methodology for fast chip design," *Nature 2021 594:7862*, vol. 594, pp. 207–212, 6 2021.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 9 2016.

[31] L. Wu, H. Lin, C. Tan, Z. Gao, and S. Z. Li, "Self-supervised learning on graphs: Contrastive, generative, or predictive," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[32] K. Ding, Z. Xu, H. Tong, and H. Liu, "Data augmentation for deep graph learning: A survey," *ACM SIGKDD Explorations Newsletter*, vol. 24, no. 2, pp. 61–77, 2022.

[33] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in neural information processing systems*, vol. 33, pp. 5812–5823, 2020.

[34] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 1548–1554, 9 2020.