IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 February 2025, accepted 5 March 2025, date of publication 12 March 2025, date of current version 21 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3550518

RESEARCH ARTICLE

Intelligent Resource Allocation via Hybrid Reinforcement Learning in 5G Network Slicing

ZAHRAA ZAKARIYA SALEH, MAYSAM F. ABBOD, (Senior Member, IEEE), AND RAJAGOPAL NILAVALAN, (Senior Member, IEEE)

Department of Electronic and Electrical Engineering (EEE), College of Engineering, Design and Physical Sciences (CEDPS), Brunel University of London, UB8 3PH Uxbridge, U.K.

Corresponding author: Rajagopal Nilavalan (nila.nilavalan@brunel.ac.uk)

This work was supported by the Department of Electronic and Electrical Engineering, Brunel University London, U.K.

ABSTRACT Manufacturers are focusing on reconfigurable, resilient environments for Industry 5.0 paradigms. Applications like digital twins and mobile robots require communication networks to meet latency, bandwidth, and reliability requirements. Beyond 5G (B5G) networks provide unprecedented communications performance and flexibility through virtualization and network slicing, which generates various logical partitions for particular applications with specific requirements. RAN slicing is an essential section of 5G network slicing due to its vulnerability to errors, affecting its ability to meet stringent reliability requirements. This paper presents a novel framework for optimizing resource allocation in 5G network slicing by integrating Double Deep Q-Network with Prioritized Experience Replay (DDQN-PER) and Pointer Network-based Long Short-Term Memory (PtrNet-LSTM). The proposed framework dynamically adjusts the attention coefficient, balancing Service Satisfaction Level (SSL) and Quality of Experience (QoE), improving system efficiency, spectrum efficiency, and user connectivity across diverse user scenarios. The experiment illustrates that the combined PtrNet-LSTM framework within DDQN-PER outperforms the baseline methods in terms of spectrum efficiency and user connectivity, demonstrating scalability and the potential to address challenges in dynamic wireless networks.

INDEX TERMS RAN, RL, pointer network, LSTM, DDQN-PER.

I. INTRODUCTION

Mobile communication networks have seen substantial advancements, progressing from 1G to 4G, offering increasingly streamlined and effective services. The process of evolution has propelled sociotechnical systems (STS), facilitating the progress of both social and economic growth. Future wireless networks are expected to expand and diversify across several dimensions. This expansion is driven by the rise in data sent across mobile networks, the growing number of people using mobile devices, and the diverse range of radio access technologies, services, and applications. To ensure this expansion, it's necessary to simultaneously pursue various objectives, such as reducing latency, increasing reliability, and improving data transmission speed tailored to the specific service. Furthermore, the distribution of resources must be dynamic, adapting to the ever-changing network circumstances and showcasing the resilience of the network.

According to Ericsson's estimation [1], the worldwide mobile data traffic is projected to exceed 160 exabytes per month by 2025. Network Slicing (NwS) is an innovative technology developed by research and development teams in industry and academic sectors. Its purpose is to enable the digital transformation of 5G networks and beyond 5G+. It divides the network into portions with unique features, catering to various user requirements. This approach deviates from the conventional 2G-4G framework, giving more adaptability, enhanced effectiveness, and innovative service options for Mobile Network Operators (MNOs). NwS technology enables the provision of 5G services and facilitates the introduction of cutting-edge applications, adapting to the advancements in 5G technology. The integration of this technology into 6G networks can be anticipated due to its flexibility and cost-effectiveness [2].

Network slicing includes the Core Network (CN) and the Radio Access Network (RAN). CN is a straightforward

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaodong Xu.

process that may be accomplished by increasing the scale of computer resources. RAN enables MNOs to segregate and isolate several virtual networks on a single infrastructure. Each RAN slice is ensured to be autonomous in a logical sense, enabling support for different types of 5G applications [3]. Nevertheless, implementing RAN slicing is still difficult because of the limited availability of radio resources, the unpredictable nature of wireless channel conditions, and the presence of interference. RAN slicing at the spectrum level provides much isolation and customization compared to other RAN slicing solutions at the Radio Resource Management (RRM) level [4]. Spectrum resources are divided into carriers, with each tenant being allocated a distinct carrier to provide total performance isolation among slices. This enables the customization of slices according to the individual requirements of each tenant across all RRM features. Furthermore, effective radio resource management enhances the provision of services, usage of resources, and generation of income for MNOs [5].

Machine learning (ML) is essential for advancing future wireless networks because of their intricate nature, lack of adequate models, and algorithm deficiencies. Machine learning can tackle the intricacies of network systems while delivering competitive performance. It can represent dynamic networks' complex and non-linear characteristics, which are typically challenging to predict using basic modelling methods. Moreover, ML may achieve an optimal equilibrium between satisfactory system performance and intricacy, guaranteeing that networks empowered by reinforcement learning (RL) can efficiently manage the many complexities of contemporary cellular networks [3]. RL strategies are employed to address the RAN slicing challenge because they can adjust to intricacies. Conventional approaches necessitate a mathematical expression that cannot be derived because of the diverse nature of slice Service Level Agreements (SLAs), the unpredictability of wireless communications, and the intricacy of resource-sharing methods based on queuing [6]. RL algorithms can adjust to various surroundings and automatically consider future requirements.

The RAN domain of 5G network slicing faces challenges in managing finite resources to meet diverse service categories, including Enhanced Mobile Broadband (eMBB) services that require high throughput, while Ultra-Reliable Low Latency Communication (URLLC) services require strict latency guarantees. These challenges should be balanced to guarantee a high Service Satisfaction Level (SSL) for eMBB users and overall network system capacity. Furthermore, the dynamic nature of user traffic, channel conditions, and limited radio resources increase the difficulty of achieving this balance

The main contribution of our study is the introduction of a novel solution to mitigate the trade-off between enhancing SSL and potential decreases in system capacity and Quality of Experience (QoE) for eMBB users, which arise from increasing the attention coefficient (α). We propose integrating DDQN-PER (Double Deep Q-Network with Prioritized Experience Replay) with PtrNet-LSTM (PointerNetwork Based Long Short-Term Memory) into the resource allocation system to adjust α in real-time dynamically. This integration optimizes the balance between SSL and QoE based on current network conditions. Leveraging PtrNet-LSTM's advanced learning capabilities enhances adaptability to varying inputs and network requirements, leading to more reliable and efficient network solutions. Ultimately, our approach allows for the improvement of SSL without significant sacrifices in system capacity or QoE, achieving a more equitable optimization of network performance.

This paper is structured as follows: In Section II, we conduct a thorough review of the relevant research. Section III explores the orchestration framework and outlines the issue formulation referred to as QoE trade-off analysis. We propose the integration of DDQN-PER with PtrNet-LSTM to address this problem. Section IV indicates the suggested reinforcement learning methodology, demonstrating the technical components of DDQN-PER and PtrNet-LSTM integration. Section V highlights the experimental configuration, including the training and testing methodologies used for model evaluation. Section VI examines the outcomes, highlighting improvements in SSL, QoE, and scalability. Section VII addresses the consequences, limitations, and prospective directions for future research, whereas Section VIII concludes the study by summarizing the main findings and contributions.

II. RELATED WORK

The variability of radio channel conditions and the specific features of broadcasting provide difficulties for RAN slicing, particularly in the context of End-to-End (E2E) network slicing. Radio resources are commonly represented as Physical Resource Blocks (PRBs) in both the time and frequency domains, necessitating careful allocation of limited spectrum resources. Researchers have investigated several ways to handle these difficulties, from traditional optimization techniques to recent reinforcement learning (RL) and hybrid deep learning methods. This section presents key contributions in these fields, with their advantages, disadvantages, and relevance to our suggested framework.

A. TRADITIONAL OPTIMIZATION APPROACH

Traditional optimization approaches such as Integer Linear Programming (ILP) and Convex Optimization have been widely explored for network slicing. Ma et al. [7] proposed a novel approach using mixed-integer programming to enhance spectral efficiency and ensure reliable Ultra-Reliable Low Latency Communication (URLLC) in a virtualized wireless network, while Li et al. [8] extended this by introducing two-level medium access controls (MACs) that allocate resource allocation dynamically. A complementary study by Marabissi and Fantacci [9] introduced a two-tier PRB scheduling technique that consists of an inter-slice scheduler that determines the allocation of resources for each slice and intra-slice schedulers that are particular to individual slices. In [10], the authors proposed a resource allocation strategy that segments RAN and its transport network to enhance URLLC performance while optimizing MNO profitability. This model employs a Flexible Functional Split methodology to allocate resources. The simulation findings indicate that there is no simple solution owing to the non-linear character of the problem.

B. REINFORCEMENT LEARNING METHODS

Given the limitations of traditional methods, RL has emerged as a promising approach for resource allocation in 5G networks. For example, Zhao et al. [11] presented a RL method to enhance long-term utility in heterogeneous cellular networks. Their approach utilized multi-agent reinforcement learning to optimize a distributed system. They also proposed a strategy called Dueling Double Deep Q-network (D3QN), which allowed dispersed User Equipment (UE) to efficiently acquire the whole state space while minimizing communication and quickly reaching a subgame-perfect Nash equilibrium (SPNE). The simulation findings showed that D3QN surpassed other reinforcement learning methods in effectively tackling complex learning issues on a broad scale. Similarity, Ren et al. [12] developed a two-level Cloud Radio Access Network (C-RAN). The deployment module utilizes a Weighted GraphSAGE-assisted DDQN-based two-stage service deployment (WDTSD) algorithm to optimize service deployment. On the other hand, the monitoring module employs a Bayesian Convolutional Neural Network (BCNN) state monitoring model to detect abnormal devices. The results indicate that the WDTSD algorithm surpasses current solutions regarding memory use, computational speed, end-to-end latency, and service access ratio. Additionally, [13] the resource allocation model for the Average age of Information (AoI) has been investigated in Wireless powered Internet of Things (WIoT) networks. These techniques aim to optimize transmission targets, channel selection, data transmission duration, and power allocation. The authors have implemented advanced reinforcement learning techniques such as Deep Deterministic Policy Gradient (DDPG) and Deep Q-Network (DQN) to resolve these challenges. Their experimental results have illustrated improved performance in terms of power consumption and data transmission rate. However, it lacks dynamic mode selection for energy harvesting Device-to-Device (D2D) networks. This was dealt with by Liu et al. [14], who used a Twin Delayed Deep Deterministic Policy Gradient (TD3) approach to create Mode Selection and Resource Allocation (MSRA). Their approach decreases Q-value overestimation and adapts bandwidth, power, and mode selection. The authors [15] provided an innovative RL approach for scheduling cellular network resources, utilizing the Proportional Fair (PF) scheduling method. They propose boosting the convergence speed of the DRL agent's performance during exploration. The proposal suggests deploying both algorithms as competitors and evaluating the RL reward by comparing the system's performance. In [16], the authors introduced an evolutionary approach to allocating PRB that relies on social connections among users connected to many network slices. This system operates dynamically and evolves to maximize transmission bit rate and resource usage. In this approach, individuals are categorized into distinct groups based on their similarities, assuming that all members of a group require comparable service. The simulation results demonstrate that the suggested technique outperforms [17] in terms of transmission rate, resource consumption, and request acceptance rate for both data and video slices. Previous investigations in high mobility environments have mostly focused on proactively managing user mobility to enable handover before the wireless channel quality significantly degrades [18]. The channel state of a very mobile user is often impaired owing to the Doppler effect [19]. However, there are exceptions to this. As highly mobile individuals move across a cell coverage area, their channel connection quality improves and worsens, leading to oscillations in the channel condition [20]. The reference [21] offers a PRB allocation system that utilizes distributed reinforcement learning. This scheme aims to enable each slice to allocate radio resources efficiently in parallel. In [22], the authors examined the situation where eMBB and URLLC coexist and proposed a combined scheduling approach to attain a long-term QoS compromise between eMBB services and URLLC services. The service satisfaction level (SSL) is intended to achieve user-centric compromises and fairness for both users. When a network controller effectively utilizes channel variations for resource scheduling, even users who are often moving can offset the decline in network performance.

C. HYBRID APPROACH METHOD

Recent studies have focused on hybrid models that combine deep learning with RL to improve resource allocation. Li et al. [23] integrated LSTM into a method based on deep reinforcement learning (DRL) to acquire the capability to monitor and predict user movement in the RAN slicing. Nevertheless, this methodology solely employs LSTM for forecasting alterations in packet arrival and exclusively focuses on users with limited mobility, specifically those traveling at 30 km/h or lower speeds. Recently, the authors [24] have been proposed a Deep Reinforcement Learning (DRL) framework for RAN slicing, which dynamically allocates network resources to each slice, ensuring quality of service (QoS) criteria are met. The framework is utilized the Action Factorization (AF) architecture, a soft-max layer, and LSTM to address performance concerns for users with frequent mobility. The system has been effectively distributed bandwidth among slices, optimizing rewards for achieving QoE for each user in high mobility situations. The authors in [25] propose a centralized architecture that utilizes the three-dimensional convolutional neural network (3DCNN) approach to estimate resource demand in slices. The objective is to reduce the total costs of resource provisioning. Similarly, the authors in reference [26] propose a modified version of LSTM to investigate the issue of predicting future demand for specific slice services. Achieving high accuracy in hourly traffic forecasting is possible by combining the sequence-to-sequence learning paradigm and a convolutional long short-term memory (S2SConvLSTM) network. The reference [27] investigated the resource allocation process in future cellular networks, where MNOs flexibly give resources to third-party service providers (SPs). The primary objective is to develop a cutting-edge technique empowering service providers to optimize resource allocation, thereby reducing potential financial risks actively. The study implemented a data-driven approach that integrates deep neural networks (DNNs) and LSTM recurrent networks solution for reserving resources from the perspective of a SP. It showed that this solution achieves higher accuracy compared to a hypothetical baseline technique.

Resource management frameworks for network slicing have integrated traffic forecasting features. The authors in [28] develop a powerful collaborative learning framework that utilizes LSTM for long-term hourly traffic forecasting of each slice and Asynchronous Actor-Critic Agent (A3C) for short-term traffic scheduling in the range of milliseconds. This framework is designed to optimize resource utilization and maintain performance isolation among slices. In addition, the authors in reference [29] combine LSTM with a heuristic-based approach to significantly enhance the user acceptance rate in their resource management framework. LSTM is implemented to predict the bandwidth demand of each slice, thereby improving the decision-making process of the entire framework. It's worth noting that all of these initiatives are centralized forecasting methodologies, which do not take into account the data privacy of slice tenants and end-users. It is essential to mention that these studies analyze traffic aggregated at the slice level rather than at the level of individual base stations.

Reference [30] introduced a machine learning framework designed to accelerate the convergence of deep reinforcement learning agents in the scenario of downlink resource allocation for URLLC. The authors utilize generative adversarial neural networks (GANs) to pretrain the DRL architecture by combining authentic and artificial data. This approach allows the DRL agent to gain offline experience by exposing it to various network situations before being deployed in an actual network. Moreover, this strategy has the potential to help the DRL agent recover quickly when it encounters a severe situation in the actual network.

In [31], a meta-learning strategy is suggested to fine-tune an RL solution. The authors propose that a meta-tuned RL agent would exhibit accelerated convergence in unfamiliar situations. The proposal suggests utilizing a reinforcement learning (RL) agent to address an optimum coverage problem by controlling drone base stations (DBSs). In this scenario, DBSs must offer uplink connectivity to ground users in response to their random access requests. In [32], the authors presented a Federated Proximal Long Short-Term Memory (FPLSTM) framework for mobile network operators (MVNOs) that utilizes Federated Learning (FL). This framework enables local models to be trained using their datasets and then share the weight with a central entity. This strategy attains comparable forecasting precision while decreasing communication and computing expenses compared to centralized systems. Nevertheless, the costs associated with super dense network installations in networks beyond 5G could represent significant barriers. The authors suggested an Information-based Clustering FPLSTM (IC-FPLSTM) clustering method to address the challenges of managing large-scale networks and achieving computational cost efficiency. The IC-FPLSTM exceeds current centralized solutions and baseline models in terms of accuracy, scalability, heterogeneity, sample efficiency, communication, and computing efficiency.

Furthermore, several studies [33], [34], [35] have proven the efficacy of satellite caching. The symbiotic connection between satellite caching and RAN resources can be utilized when RAN edge resources are scarce. While there has been notable progress in studying 3C resources in RAN, the complexity and multi-dimensional nature of 3C resource allocation highlights the necessity for more research and comprehension. The simultaneous examination of 3C resources for RAN network slicing is a subject that requires substantial further investigation. The authors of [36] have investigated the application of network slicing in low earth orbit (LEO) satellite caching-assisted communication. The system suggested a resource-slicing system that employs the sequential quadratic programming (SQP) iteration method while incorporating the LSTM and Soft Actor-Critic (SAC) approach hierarchically. The technique is specifically developed to tackle the resource allocation problem in LEO-RAN edge settings, known as the 3C challenge. It enhances the speed and decreases the delay while meeting SLAs. Simulations demonstrated that the method achieves a balanced 3C resource allocation more effectively than the theoretical optimal option.

The study [44] examines the implementation of the DDQN-PER algorithm for managing the distribution of radio resources in RAN. When evaluated with 15 URLLC customers and 5 eMBB users, this algorithm showed better learning effects and stability than natural DQN and DDQN, especially during 500 training episodes. Upon closer examination of the attention coefficient (α), it is clear that raising α enhances the SSL but leads to a decrease in system capacity and QoE for eMBB users, indicating a trade-off. We proposed a solution to tackle this problem is the integration of DDQN-PER with PtrNet-LSTM (Pointer Network Based Long Short-Term Memory) into the resource allocation system. PtrNet-LSTM can dynamically adjust α in real-time, optimizing the balance between SSL and QoE based on current network conditions. The increased learning capabilities of the system allow for improved adaptability to varying inputs and network requirements, resulting in more reliable and efficient solutions. By utilizing PtrNet-LSTM, it is feasible to improve SSL without substantially sacrificing system capacity or QoE, achieving a more equitable optimization of network performance. Table 1

TABLE 1. Comparison of resource allocation methods.

Ref.	Algorithm	Focus Area	Optimization Goal	Use Case	Training Method	Development Environment
[10]	Integer Linear Programming (ILP)	5G Hybrid C-RAN	Optimal computational resource allocation and network slicing	eMBB, URLLC, mMTC	Centralized	Simulation
[15]	Deep Reinforcement Learning (DRL)	Resource Scheduling	Optimize scheduling decisions to balance throughput and fairness	Cellular networks with proportional fairness	Centralized	Simulation (TensorFlow)
[22]	Deep Deterministic Policy Gradient (DDPG)	Joint Scheduling	Achieve QoS tradeoff between eMBB and URLLC	5G Networks with eMBB and URLLC	Centralized	Simulation
[23]	LSTM-based Advantage Actor-Critic (A2C)	Resource Management	Optimize resource allocation with SLA satisfaction and spectrum efficiency	eMBB, URLLC, VoLTE	Centralized	Simulation
[24]	DRL with Action Factorization and LSTM	RAN	Maximize QoE while addressing scalability and mobility challenges	5G RAN slicing with highly mobile users	Centralized	Simulation
[28]	A3C	RAN	Resource maximization while ensuring slice separation	Not specified	Distributed	Emulation (TensorFlow)
[36]	LSTM-SAC	LEO-RAN Slicing	Optimize 3C resource allocation	LEO-RAN with joint communication, computing, and caching	Centralized	Simulation
[37]	DQN	RAN	Improve resource consumption and slice isolation	Continuous bit rate, lowest bit rate	Centralized	Simulation
[38]	Q-learning	RAN	Maximize resource use during communication assembly	Haptic communications	Centralized	Simulation
[39]	Q-learning, SARSA, Monte Carlo	RAN	Efficient resource use for low-latency demands	Internet of Things	Centralized	Simulation
[40], [41]	DDQN, Dueling DQN	RAN	Maximize long-term profits while serving multiple tenants	Manufacturing, automotive, utilities	Centralized	Emulation (TensorFlow)
[42]	DQN	RAN	Maximize radio resource utilization while preserving QoS	eMBB, URLLC	Centralized	Simulation
[43]	LSTM	RAN	Maximize spectrum efficiency and SLA satisfaction	eMBB, URLLC	Centralized	Simulation
This Paper	DDQN-PER with PtrNet-LSTM	RAN	Dynamic optimization balancing QoE, SSL and spectrum efficiency	eMBB, URLLC	Centralized	Simulation (Python, PyTorch and TensorFlow)

depicts the RL-based resource allocation methods, comparing their algorithms, focus areas, optimization goals, use cases, training methods, and development environments.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

This research introduces an innovative approach to resource allocation in the context of RAN slicing, aimed at enhancing network performance through the integration of advanced reinforcement learning techniques. The method combines Double Deep Q-Network with Prioritized Experience Replay (DDQN-PER) and Pointer Network-based Long Short-Term Memory (PtrNet-LSTM), enabling dynamic, real-time resource distribution under unpredictable network conditions.

Our system seeks to optimize the allocation of radio resources by simultaneously improving the SSL and ensuring a high QoE. A key feature of this approach is the adaptive adjustment of the attention parameter, α, within the reward function. This adjustment enables the system to dynamically balance SSL and QoE based on current network conditions. The proposed system offers significant improvements in flexibility and resource utilization compared to conventional solutions [44], which often rely on static parameters and are less effective in responding to dynamic fluctuations in user demand and network performance. The system model is characterized as follows:

1) PRB ALLOCATION FRAMEWORK

The RAN controller operates on a matrix A_{MXN} is used to represent the PRB assignment result of the BS. Each element $A(m, n)$ is a binary decision variable that indicates whether a specific PRB m is allocated to a user n. The meaning of $A(m, n)$ is formally defined as:

$$A(m,n) = \begin{cases} 1, & \text{if PRB } m \text{ is allocated to user } n, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In this definition:

- $m \in M = \{1, 2, \ldots, M\}$ represents the index of available PRBs, which are the fundamental time-frequency resource units in the RAN.
- $n \in N = \{1, 2, \ldots, N\}$ represents the index of users in the network, each with specific Quality of Service (QoS) requirements, such as throughput, latency, and reliability.

The allocation matrix $A(m, n)$ serves as a central decision variable in resource allocation, determining how PRBs are distributed among users to meet their QoS requirements. Its binary nature ensures that a PRB is either allocated to a single user ($A(m, n) = 1$) or left unallocated ($A(m, n) = 0$).

2) KEY CONSTRAINTS

According to Constraint in equation (2), the maximum number of PRBs that users can occupy is limited to the total number of PRBs, and Constraint in equation (3) indicates the limit for PRB allocation. P_m is the symbol for the BS transmit power of PRB. The maximum overall power for the BS is defined in equation (4).

$$\sum_m \sum_n A\left(m, n\right) \leq M \tag{2}$$

$$\sum_n A\left(m, n\right) \leq 1 \tag{3}$$

$$\sum_m P_m \leq P_{\max} \tag{4}$$

3) THROUGHPUT MODEL

To find the transmission rate, assume that user n is in the PRB m as mentioned in equation (5), in which σ denotes spectral density of noise, and I represent interference from other BSs. The channel gain is denoted by g_n, is a key parameter in determining the achievable data rate. It is formulated to account for large-scale fading effects between the BS and the user, including both path loss and shadow fading. Specifically, the path loss is calculated using a logarithmic distance-based model: $L(\text{dist}) = 37 + 30\log(\text{dist})$ [45], [46] where distance is the Euclidean distance between the BS and the user. To account for environmental variations caused by obstructions such as buildings and terrain, shadow fading is modeled as a zero-mean Gaussian random variable with a standard deviation of $\sigma = 8\,\text{dB}$. The combined effects of path loss and shadow fading are then used to compute the channel gain in linear scale as: $g_n = 10^{-\frac{L(\text{distance})+\text{Shadow Fading}}{10}}$. This formulation ensures that the channel gain reflects realistic propagation environments, making the derived data rate consistent with practical wireless scenarios. The measurement of $RB_{m,n}$ is in kbps.

$$RB_{m,n} = A\left(m, n\right) B_l \log_2 \left(1 + \frac{P_m g_n}{\sigma B_l + I}\right) \tag{5}$$

The user throughput that has been estimated as mentioned in equation (6).

$$R_n = \sum\nolimits_m A\left(m, n\right) RB_{m,n} \tag{6}$$

4) LATENCY MODEL

The operational delay observed by the URLLC service traffic can be disregarded. To simplify the transmission model, consider URLLC service traffic as an M/M/1 queuing model. Let's assume that the average packet length is represented by the variable μ, and the rate at which packets arrive is represented by the variable λ. The URLLC user's average transmission delay on the channel may be mathematically expressed as follows in equation (7):

$$\tau_n^1 = \frac{1}{\frac{R_n}{\mu} - \lambda} = \frac{\mu}{R_n - \mu\lambda} \tag{7}$$

The data streams carried by the eMBB protocol are regular and periodic. Due to its ongoing transmission, data must be sent within a specific time frame. In order to decrease transmission delay and reduce RAN congestion, it is efficient to cache popular files near the network edge, such as at the BS. The request "leaves" the network until the desired file has been totally downloaded when an eMBB user performs a specific file request handled by a related BS.

Every file in the BS should have a unique virtual queue. Equation (8) [47] describes the edge delay that an eMBB transmission encounters on its way from the base station to the end user:

$$\tau_n^1 = \upsilon + \frac{L_q(t)}{S(t)\,(1 - P_K)} \tag{8}$$

where:

- The variables υ are the processing time of the BS.
- $L_q(t)$ is the number of packets in the queue.
- $S(t)$ is the number of packets serviced in a specific time.
- P_K the packet loss rate from the queue.

$$\begin{aligned} \tau_n^1 = \upsilon + \frac{L_q(t)}{\frac{R_n}{\mu}\,(1 - P_K)} &= \upsilon + \frac{\mu L_q(t)}{R_n\,(1 - P_K)} \\ &= \upsilon + \frac{L}{R_n\,(1 - P_K)} \end{aligned} \tag{9}$$

$$R_n \geq R^{rsv} \tag{10}$$

In the equation (9), L denotes the size of the requested file. The eMBB users should meet throughput constraint (10). Each user makes a distinct file request, which prevents a backlog from forming at the BS. The variable R_n determines the number of packets processed over time, while the user's perception of edge delay is illustrated in equation (10).

The latency constraint for all users is then given by:

$$\tau_n^1 \leq \tau_n^{\text{RAN}} \tag{11}$$

where τ_n^{RAN} is the maximum allowable RAN latency for user n.

B. PROBLEM FORMULATION

The objective function of our system is to maximize both the SSL for eMBB users and the overall system capacity QoE. The optimization problem is framed as follows:

$$\text{Maximize:}\ \alpha \cdot \underbrace{\sum_{n\in\text{eMBB}} \log_{10}\left(1 + \frac{R_n}{R^{\text{rsv}}}\right)}_{\text{SSL}} + (1-\alpha) \cdot \underbrace{\frac{\sum_{n\in\text{users}} x_n}{N}}_{\text{QoE}} \tag{12}$$

where:

- $\alpha \in [0, 1]$ is the attention coefficient that adjusts the trade-off between service satisfaction and system capacity.
- R^{rsv} is the reserved throughput level for eMBB users.
- x_n is a binary variable that indicates whether user n successfully accesses the network (1 if successful, 0 otherwise).
- N is the total number of users.

Subject to the following constraints:

- **PRB Allocation:** Each PRB can only be allocated to one user:

$$\sum_{n=1}^{N} A(m, n) \leq 1, \quad \forall m \in M. \tag{12a}$$

- **Latency:** URLLC users must meet their latency requirements:

$$\tau_n^1 \leq \tau_n^{\text{RAN}}, \quad \forall n \in \text{URLLC users.} \tag{12b}$$

- **Throughput:** eMBB users must achieve the minimum reserved data rate:

$$R_n \geq R^{rsv}, \quad \forall n \in \text{eMBB users.} \tag{12c}$$

- **Power Constraint:** The total power consumption must not exceed the maximum allowable power:

$$\sum_{m=1}^{M} P_m \leq P_{\max}. \tag{12d}$$

- **Binary Allocation:** PRB allocation is binary:

$$A(m, n) \in \{0, 1\}, \quad \forall m \in M, \forall n \in N. \tag{12e}$$

The SSL for eMBB users is computed as:

$$\text{SSL} = \sum_{n\in\text{eMBB}} \log_{10}\left(1 + \frac{R_n}{R^{\text{rsv}}}\right) \tag{13}$$

The Equation (13) utility function is used to ensure fairness in resource allocation among eMBB users. By employing this function, the system avoids the over-allocation of resources to a small subset of eMBB users with already high throughput, thus promoting a more balanced distribution of resources. This approach ensures that users with lower throughput receive a proportional allocation of resources, improving their service satisfaction.

The QoE metric is defined as the ratio of successfully served eMBB and URLLC users to the total number of eMBB and URLLC users. This can be mathematically expressed as:

$$\text{QoE} = \frac{\sum_{n=1}^{N_{\text{eMBB}}} x_n^{\text{eMBB}} + \sum_{n=1}^{N_{\text{URLLC}}} x_n^{\text{URLLC}}}{N_{\text{eMBB}} + N_{\text{URLLC}}} \tag{14}$$

where:

- N_{eMBB} is the total number of eMBB users,
- N_{URLLC} is the total number of URLLC users,
- x_n^{eMBB} is a binary variable for each eMBB user, where $x_n^{\text{eMBB}} = 1$ if the eMBB user n successfully receives the required throughput, and $x_n^{\text{eMBB}} = 0$ otherwise,
- x_n^{URLLC} is a binary variable for each URLLC user, where $x_n^{\text{URLLC}} = 1$ if the URLLC user n successfully meets the latency requirement, and $x_n^{\text{URLLC}} = 0$ otherwise.

In (14) formulation, the QoE metric reflects the proportion of users who are successfully served, incorporating both the high throughput requirements of eMBB users and the stringent latency demands of URLLC users. The combination of both user types in the metric allows for a comprehensive evaluation of network performance across heterogeneous traffic profiles.

IV. PROPOSED APPROACH

DDQN enhanced resource allocation by addressing key challenges in dynamic environments, such as network slicing [48]. DDQN mitigated overestimation bias in traditional Q-learning by using separate target networks, ensuring stable and reliable decision-making. PER improved learning efficiency compared to traditional experience replay techniques, which treat all experiences equally [49]. PRB enhanced the processes by focusing on transactions with huge errors and rewards, resulting in faster convergence and enhanced decision-making in dynamic environments. However, testing found that DDQN-PER struggled to capture temporal dependencies and adapt to rapidly changing network conditions. The proposed hybrid approach addresses the optimization problem in **Equation 12** with constraints (12a–12e). The issue is addressed using a two-stage process combining *Double Deep Q-Network with Prioritized Experience Replay (DDQN-PER)* and *Pointer Network-based Long Short-Term Memory (PtrNet-LSTM)*. In the first stage, DDQN-PER tackles resource allocation by learning policies that meet to **PRB allocation (12a)** and **latency constraints (12b)**. In the second stage, PtrNet-LSTM refines DDQN-PER's outputs to further optimize **throughput (12c)** and enforce **binary constraints (12e)**, ultimately improving system performance metrics such as **SSL** and **QoE**.

A. DDQN-PER

RL is a process in which an agent learns by actively engaging with the environment, taking actions, and getting feedback on its performance. The primary objective is to optimize the total reward, which may assist in finding suboptimal solutions to intricate issues. The objective of this work is to optimize the QoE and reward value for all eMBB and URLLC users. The RAN domain controller functions as the agent. The user's SINR (Signal-to-Interference-plus-Noise Ratio) is represented by the state, and it is normalized before being inputted into the neural network. Normalization assists in accelerating the rate at which the network reaches convergence. Equation (15) illustrates the set as the state, while equation (16) defines the action as assigning *PRB* and power to the user.

$$s_t = \left\{\mathrm{SINR}_t^n\right\}_{1\times N} \tag{15}$$

$$a_t = \{\{a_t^n\}_{1\times M}, \{P_t^n\}_{1\times M}\}_{1\times N} \tag{16}$$

Equation (17) defines the available choices for the user's transmission power.

$$P_n \in \left\{0, \frac{P_{\max}}{M-1}, \frac{P_{\max}}{M-2}, \ldots, P_{\max}\right\} \tag{17}$$

The reward function $R = \alpha SSL + (1-\alpha)QoE$ can be established according to the objective function.

DQN, a deep reinforcement learning method, leverages the Double Deep Q-Network with a Prioritized Experience Replay (DDQN-PER) technique. This method, unlike the traditional Q-Learning approach, determines the Q-value using a neural network. DQN uses a neural network to determine the Q-value for each action by considering the present condition. DQN incorporates key concepts from Q-Learning, replacing the previous Q value (s, a) with the updated version $(s, a; \theta)$, which represents the parameters of the neural network. To ensure the effectiveness of the training process and minimize data correlation, DQN implements two powerful strategies: experience replay and fixed Q-targets, which have been proven to significantly enhance the performance of the method.

DDQN addresses the issue of inaccurate estimation present in DQN, which primarily arises from the maximization operation in Q-Learning. Overestimation occurs when the calculated value function exceeds the actual value function. The loss function differs in DDQN compared to DQN, but otherwise, the two methods are functionally similar. Unlike DQN, which uses a single Q-network, DDQN introduces two separate Q-networks: one for action selection and another for action evaluation. This separation helps mitigate the overestimation issue.

In DDQN, the target Q-value is determined in two steps:

1) **Action Selection**: The Q-network estimates the action that maximizes the Q-value as demonstrated in equation (18):

$$a_{\max} = \arg\max_{a'} Q(s', a'; \theta) \tag{18}$$

where θ is the parameter of the online neural network.

2) **Target Calculation**: The target Q-network then calculates the corresponding Q-value for this selected action as illustrated in equation (19):

$$Q_T = r + \gamma Q(s', a_{\max}; \theta^-) \tag{19}$$

where γ is the discount factor, controlling the impact of future rewards.

By merging the two procedures, one may obtain the DDQN loss function, which allows for unbiased action estimates during training, as demonstrated in equation (20). The loss function used in the DDQN-PER algorithm is crucial for optimizing the Q-network during the training process. The loss function is defined as:

$$\mathcal{L}_{\mathrm{DDQN}} = \left[r + \gamma Q\left(s', \arg\max_{a'} Q(s', a'; \theta); \theta^-\right) - Q(s, a; \theta)\right]^2 \tag{20}$$

The loss function expressed in equation (20) serves as the primary mechanism for refining the action-value function estimates within the DDQN framework. The minimization of this loss function is conducted iteratively through the use of the Adam optimizer, enabling the network to converge towards an optimal policy. Below is a step-by-step description of how the loss function is utilized in the training process:

- **Initialization:** The parameters of the neural network, denoted by θ, are initialized. These parameters represent the weights and biases of the neural network used to approximate the Q-values.

- **State Input and Action Selection:** The state s_t, which in this case is the normalized SINR of the users, is passed as input to the neural network. The DDQN algorithm then selects an action a_t by computing the Q-values for all possible actions and choosing the one that maximizes the Q-value as demonstrated in equation (18).
- **Reward Calculation:** Once the action is taken, the environment RAN returns a reward r, which is a function of SSL and QoE, as defined by the reward function in equation (19).
- **Target Q-Value Calculation:** The target Q-value is calculated based on the reward r and the next state s'. The target Q-value is computed using the target network, with fixed parameters θ^-, as illustrated in equation (19).
- **Loss Function Computation:** The loss function, shown in equation (20), represents the squared difference between the target Q-value and the predicted Q-value for the current state-action pair. This loss function measures how close the network's Q-value predictions are to the target Q-values.
- **Parameter Update via Adam Optimizer:** The Adam optimizer is employed to update the neural network's parameters θ. The gradients of the loss with respect to the parameters are computed, and Adam adjusts the parameters based on these gradients, using adaptive learning rates and momentum (first and second moments). The update rule for Adam is:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{21}$$

  Here, $\hat{m}_t$ and $\hat{v}_t$ are the biased-corrected first and second moments of the gradients, α is the learning rate, and ϵ is a small constant to prevent division by zero.
- **Replay Buffer with Prioritized Experience Replay (PER):** The experience, consisting of the state, action, reward, and next state, is stored in the replay buffer. Unlike standard DQN, which samples uniformly from the buffer, PER prioritizes transitions based on their Temporal-Difference (TD) error. The TD error is calculated as:

$$\delta = |r + \gamma Q(s', a_{\max}; \theta^-) - Q(s, a; \theta)| \tag{22}$$

  Larger TD errors indicate that the model made significant mistakes in predicting the Q-value for a transition, making those experiences more important for learning. Random sampling in the DQN procedure can reduce the effectiveness of learning when the replay buffer grows large. The goal of PER is to increase the likelihood of sampling transitions with higher TD errors. PER assigns a priority value p_i to each transition, defined as:

$$p_i = \delta + \zeta \tag{23}$$

  where ζ is a small constant added to ensure that even transitions with low TD errors still have a non-zero chance of being sampled. The sampling probability of a transition is proportional to its priority:

$$P(i) = \frac{p_i^\beta}{\sum_j p_j^\beta} \tag{24}$$

  where β controls the level of prioritization. To efficiently implement these priorities, a Sum-Tree data structure is used, allowing for fast updates and sampling of priorities.

  This prioritization enhances the learning process by focusing the network on more critical experiences, which improves both the speed of training and decision-making performance.
- **Iterative Training:** The neural network is trained iteratively by following the above steps. As the network's parameters θ are continuously updated to minimize the loss function, the Q-value estimates become increasingly accurate. Over time, this leads to improved policy performance as the network learns to select actions that maximize long-term rewards. By leveraging DDQN to reduce overestimation bias and PER to prioritize important transitions, the training process becomes more efficient and yields better results compared to standard DQN approaches.

The DDQN-PER for the RAN resource allocation process is summarized in Algorithm 1.

B. DDQN-PER WITH PTRNET-LSTM

Large-scale network topologies in the real world are complicated, leading to a fast growth in physical complexity. Like genetic algorithms, heuristic algorithms tend to become trapped in local optimal solutions and exhibit delayed convergence. Popular RL techniques, like Q-learning, are time-consuming when exploring each state, making them impractical for practical use.

Combinatorial optimization (CO) seeks to identify the optimal selection of variables in a discrete decision space, which aligns with the essential attribute of sequential decision-making in RL [50], [51]. Integrating reinforcement learning with pointer networks enhances the use of physical nodes and increases the long-term benefit-cost ratio [45]. To increase the amount of data available for use in predicting the Q-value of the component migration issue, we design a PtrNet by incorporating the LSTM layer into a DDQN. This allows us to recall data from earlier time slots. To predict the current Q-value, we consider both recent experiences $(s, a; \theta)$ and significant events that occurred in the distant past. Fig.1 illustrates the framework of the proposed PtrNet-LSTM with DDQN.

Vinyals et al. [52] initially introduced the pointer network design to learn difficulties in combinatorial optimization of low and medium dimensions. Employing LSTM memory cells in a DDQN agent offers several theoretical benefits. Previous research has demonstrated the effectiveness of LSTM in various applications, such as forecasting user mobility [53],

Algorithm 1 DDQN-PER for RAN Resource Allocation

```
Input: $s_t = \{\mathrm{SINR}_{nt}\}_{1\times N}$, Replay buffer $D$, Learning
       rate $\alpha$, Discount factor $\gamma$, Prioritization exponent
       $\beta$, Target update rate $\tau$, Max buffer size $|D|$,
       Power levels $\{0, \frac{P_{\max}}{M-1}, \dots, P_{\max}\}$, Max BS power
       $P_{\max}$
Output: Optimal PRB and power allocation policy
Initialization: Initialize Q-network parameters $\theta$ and
  $\theta^-$, replay buffer $D$ with PER, exploration rate $\epsilon$, and
  prioritization parameters.
for each episode do
    for each time step t do
        Input normalized SINR values $s_t$ into the
          Q-network;
        Select action $a_t$ using epsilon-greedy policy;
        Execute action $a_t$, update PRB allocation matrix
          $A_{MXN}$, and compute reward $r_t$;
        Ensure system constraints: PRB allocation and
          power limits;
        Store transition $(s_t, a_t, r_t, s_{t+1})$ in $D$ and
          compute TD error;
        Assign priority to the transition based on TD
          error;
        Sample a mini-batch from $D$ using priority
          probabilities;
        Compute target Q-value $Q_T$ using the target
          network $\theta^-$;
        Minimize the loss between predicted Q-value
          and target Q-value using Adam optimizer;
        if every $\tau$ steps then
            Update target network parameters: $\theta^- \leftarrow \theta$;
    Decay exploration rate $\epsilon$ towards $\epsilon_{\min}$;
Return optimal PRB and power allocation policy;
```

solving task scheduling problems [54], [55], and optimizing resource allocation [56]. LSTM is an excellent tool for solving time series issues with long-term dependencies because of its capacity to remember past experiences [57]. This network consists of two sub-networks, specifically an encoder and a decoder. The neural network architecture is a modified version of the sequence-to-sequence model that incorporates the attention mechanism. This architecture can address combinatorial optimization problems where the size of the output dictionary is defined by the length of the input sequence. The pointer network's fundamental concept is to represent the output as a sequence of pointers that indicate the components of the input sequence with a certain probability.

The input for the encoder in this research is a vector $\{s_1, s_2, \dots, s_{N_s}\}$ consisting of DDQN nodes. The decoder's output is an arrangement of the nodes' coordinates. The embedding layer applies a linear transformation to each physical input node and sends the n-dimensional embedding information to the encoder network. The linear transformation is defined in equation (25)

$$y = x \cdot \mathrm{weight} + \mathrm{bias} \tag{25}$$

The weight is a matrix with dimensions $(\mathrm{out}_{\mathrm{feature}}, \mathrm{in}_{\mathrm{feature}})$ initialized from $U(-k, k)$, where $k = \frac{1}{\sqrt{\mathrm{in}_{\mathrm{feature}}}}$. The number of physical nodes determines the feature, whereas the current number of DDQN nodes to be mapped determines the feature. At each step, the encoder LSTM receives a new node as input and transforms it into a collection of hidden states $\{e_1, e_2, \dots, e_{|N_s|}\}$, where each e_i belongs to the set of real numbers $R|N_s|$. The encoder transfers the state of the final encoder step to the initial decoder step once it has read all nodes. The decoder additionally stores the hidden state $\{d_1, d_2, \dots, d_{|N_n|}\}$, where each d_i is a real-valued vector of size Ns. During each decoding phase, the decoder attention mechanism calculates the probability distribution for all DDQN nodes and passes the chosen nodes to the subsequent decoder step. The symbol g represents the n-dimensional zero matrix used during the decoder's initial step. Every subsequent step updates this matrix. During the first decoding stage, especially when t = 0, the decoder uses the attention mechanism to compute the likelihood of each node by considering the hidden state. The black arrow pointing towards the encoder indicates the node with the highest probability for mapping. In the subsequent decoding phase, the LSTM considers the previous output and the feature vector of the selected node of the prior step. It then employs the attention mechanism in sequence-to-sequence learning to compute the probability of each node. This method is formally represented as follows using specified symbols:

$$u_i^t = \begin{cases} v^T \tanh\left(\begin{bmatrix} W_e & W_d \end{bmatrix} \begin{bmatrix} e_i \\ d \end{bmatrix}\right), & \textit{if } (1) \textit{ holds} \\ -\infty, & \textit{otherwise} \end{cases} \tag{26}$$

$$\tilde{A_q} = \left[a_1, a_2, \dots, a_{|N_s|}\right]^T = \mathit{softmax}\left(u^t\right) \tag{27}$$

where $v \in \mathbb{R}^p$is an attention vector, and $W_e, W_d \in \mathbb{R}^{p\times p}$ are attention matrices. $A_{\mathrm{q}}[\dots]$ is the attention function, and softmax$(\cdot)$ is the softmax function, also known as the normalized exponential function. The nodes are consecutively chosen in this manner until a full selection is completed. Furthermore, the decoder now refrains from choosing the subsequent node once each node is picked, in order to guarantee the accuracy of the outcome.

V. EXPERIMENT

A. OVERVIEW OF EXPERIMENTAL SETUP

1) SIMULATION ENVIRONMENT

In this study, we designed and implemented an experiment to evaluate the proposed network slicing resource allocation framework. The experiment simulates a RAN slicing environment comprising eMBB and URLLC services, utilizing a reinforcement learning approach based on DDQN-PER and a PtrNet-LSTM model. The environment was configured to simulate dynamic resource demands across different services, allowing the model to adaptively manage resources while considering both latency and throughput. The network model was simulated within a 100m × 100m region, where a base station (BS) was placed and several users were distributed randomly. The maximum achievable transmission power $P_{\max}$ was set to 46 dBm. The user's distance from the BS was denoted as *dist*, with the path loss calculated using the equation $L(\text{dist}) = 37 + 30\log(\text{dist})$ [45], [46].

2) MODEL AND FRAMEWORK

The simulation was executed using Python with PyTorch, with key parameters including a learning rate of 0.005 and a discount factor of 0.995. The DDQN-PER model was initialized with random weights, and PtrNet-LSTM was integrated to model temporal sequences and capture long-term dependencies. An experience replay memory buffer, configured with 5000 samples, was employed. The simulation model was constructed using both TensorFlow and PyTorch. TensorFlow was selected for its scalability and comprehensive ecosystem, which supports large-scale simulations with significant computational demands. The integration of TensorBoard facilitated real-time monitoring, automatic differentiation, and distributed computing, optimizing the training of complex models. In contrast, PyTorch's dynamic computation graph provided greater flexibility for rapid prototyping and model refinement. Its strong integration with Python's native debugging tools further streamlined the experimental workflow. The combination of TensorFlow's robustness and PyTorch's flexibility resulted in optimized reinforcement learning models for 5G network slicing.

3) TRAINING AND TUNING PROCEDURES

The primary objective of the experiment was to assess the scalability and robustness of the proposed framework under varying user densities and traffic demands. The system was evaluated for three distinct user densities: 30, 40, and 50 users, reflecting realistic network scenarios with fluctuating loads. The training process aimed to optimize long-term network performance by balancing SSL and QoE through dynamic adjustment of Physical Resource Block (PRB) allocation and power levels. The model training included adjusting the attention coefficient (α) to values of (0, 0.5, 0.75, and 1), with each configuration evaluated for its effect on SSL and QoE under different network loads. Increased α values improved SSL by allocating more resources to eMBB services, while decreased values promoted a balance that maintained URLLC latency requirements. This adjustment allowed the model to enhance spectral efficiency and connection across many conditions, successfully handling network fluctuations. We selected the optimal configurations based on their ability to maintain QoE under varying user densities and fluctuating traffic demands. The model was trained over multiple episodes, with each episode adjusting α based on current network states to maximize both SSL and QoE. We incorporated prioritized experience replay to improve training efficiency, which focused learning on critical network states.

TABLE 2. System parameters.

Parameter Name	Value
Bandwidth of PRB B_l	18 kHz
BS transmit power $P_{\max}$	46 dBm
Cell coverage area	100 m × 100 m
Number of users	30, 40, 50
Noise spectrum density σ	-174 dBm/Hz
eMBB users' rate constraint	300 kbps
URLLC maximum delay constraint	10 ms
eMBB maximum delay constraint	200 ms
Packet loss rate P_K	1%
Activation Function	ReLU
Mini-batch rate	32
Discount rate γ	0.995

4) SYSTEM PARAMETERS

To evaluate the model's performance, the proposed PtrNet-LSTM integrated with DDQN-PER was tested in a RAN environment. System parameters, including the attention coefficient, were modified to thoroughly assess performance under varying network conditions. The specific parameters used in the simulation, such as PRB bandwidth, base station transmit power, and delay constraints, are summarized in Table 2. These parameters were chosen to model a realistic 5G RAN environment with varying user demands and network loads. Each user was assigned one PRB, and the static latency runs were referenced from [44]. The RAN's DDQN-PER algorithm and the PtrNet-LSTM algorithm were utilized to dynamically assign radio resources based on the adjusted proportion. The system was configured with a delay budget of 0.5 for the RAN, with scenarios involving 30, 40, and 50 URLLC users and 15 eMBB users.

5) EVALUATION METRICS

The experiment was designed to balance the trade-off between enhancing SSL and maintaining high QoE for both eMBB and URLLC users. Evaluation metrics such as throughput, latency, and system capacity were used to determine the model's ability to adapt to changing network conditions and user requirements. The focus was on ensuring that the model could dynamically allocate PRBs and power levels to minimize URLLC service delays while maximizing eMBB throughput within the constraints of real-world 5G environments. The attention coefficient α was assigned values of 0, 0.50, 0.75, and 1.0 to evaluate its effect on system performance.

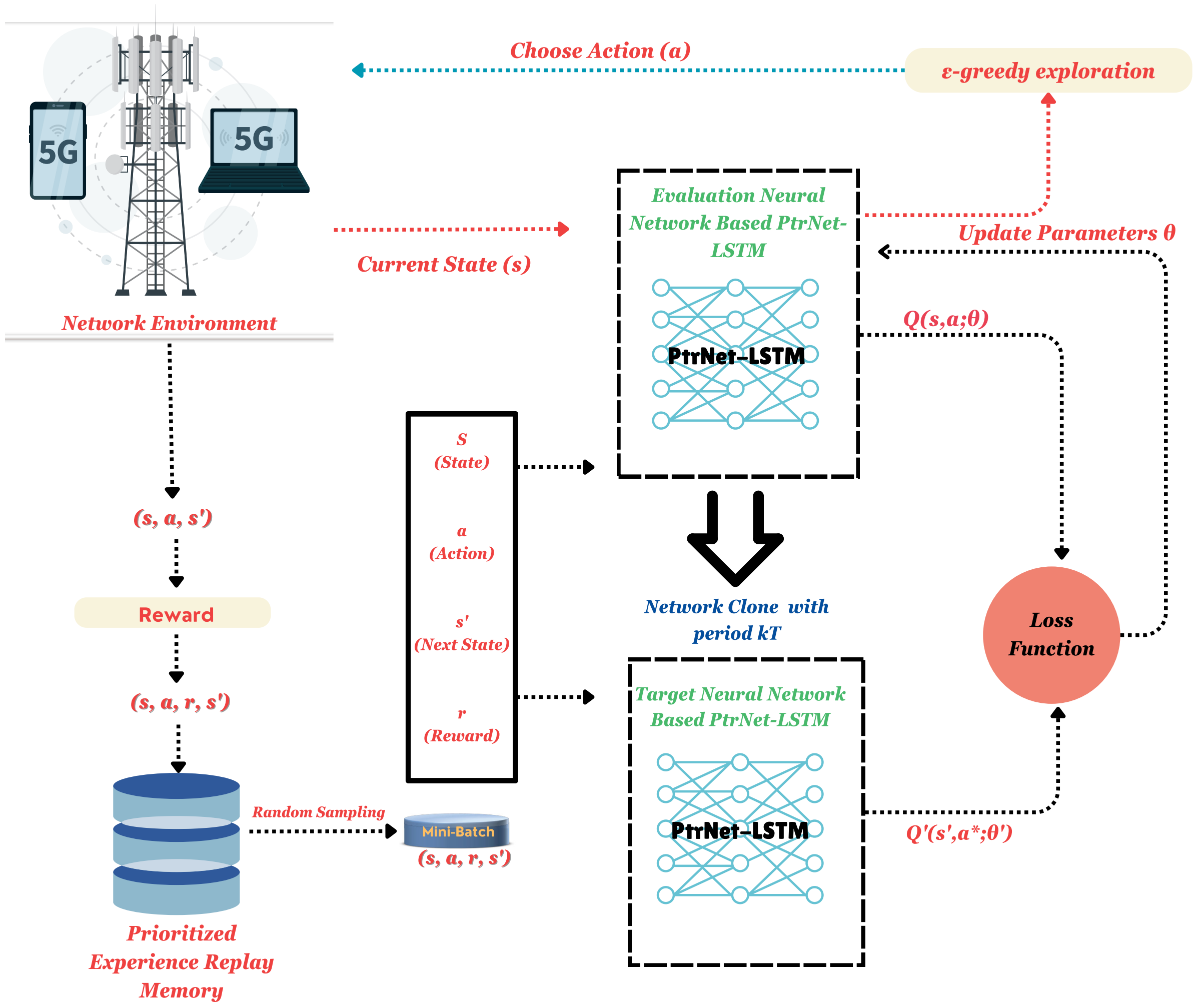


FIGURE 1. Agent-environment interaction with DDQN-PER strategy.

B. TRAINING, TESTING, AND VALIDATION PROCESS

1) DDQN-PER TRAINING

The training of the DDQN-PER model follows a standard reinforcement learning procedure, where the agent interacts with the environment, takes actions, and learns from the feedback (rewards). The key training steps are as follows:

- **State Representation**: The state space comprises network metrics such as the packet transmission success rate, SINR, and average latency across different services.
- **Action Selection**: The action space includes the possible resource allocation decisions, such as selecting the power levels to allocate to each user.
- **Reward Function**: The reward function is designed to balance QoE and throughput. For each action taken, the agent receives a reward based on the performance metrics (e.g., higher QoE and higher throughput result in better rewards).
- **Network Update**: The Q-network (evaluation network) is updated by minimizing the TD error, which is the difference between the predicted Q-value and the target Q-value (generated from the target network). The TD error is calculated in equation 22.
- **Experience Replay**: Transitions (state, action, reward, next state) are stored in a replay buffer. Experiences are sampled based on their TD error using PER, meaning transitions with higher TD errors are prioritized for replay, which accelerates learning.
- **Training Episodes**: The agent is trained over multiple episodes. During each episode, the agent interacts with the environment for a certain number of time steps, takes actions, receives rewards, and learns from these experiences through backpropagation.

2) INTEGRATION PTRNET-LSTM INTO DDQN-PER

Once the DDQN-PER model converged and stabilized (i.e., the training reward plateaus), the trained output from the DDQN-PER model (i.e., the action-value function $Q(s, a)$) was used as input for the PtrNet-LSTM model. The LSTM model is responsible for capturing temporal dependencies in resource demand and service requirements that change over time. The key steps are:

- **Sequence Input**: The LSTM receives a sequence of feature vectors, each representing the output from the DDQN-PER model over consecutive time steps. This sequence represents historical resource demands.
- **Memory Function**: The LSTM's memory function allows it to track changes in resource demand over time, making the model more adaptive to fluctuating network conditions.
- **Action Prediction**: The output of the LSTM is passed through fully connected layers that predict the action to take in the next time step. These layers adjust the weights iteratively to minimize the difference between the predicted Q-values and the target Q-values.
- **End-to-End Training**: The entire process (DDQN-PER + PtrNet-LSTM) is trained end-to-end in the later stages to refine both components. This training helps optimize the final policy for resource allocation in the network slicing environment.

3) TESTING AND VALIDATION

- **Testing**: After training, the model is tested using a separate testing dataset. This dataset includes scenarios that were not seen during training, allowing for an unbiased evaluation of the model's performance.
- **Validation**: Validation is performed on a validation dataset, which is distinct from both the training and testing datasets. The model's performance is measured based on several metrics, including latency, throughput, and QoE for both eMBB and URLLC services.
- **Performance Metrics**: The primary performance metrics used in this study include:
  - **Latency**: Measures the time delay for packet transmission, with a focus on minimizing latency for URLLC users.
  - **Throughput**: Represents the rate of successful message delivery over the communication channel.
  - **QoE**: Quality of Experience, focusing on user satisfaction levels across the network.

The validation results demonstrated that the LSTM model effectively adapted to varying user demands, significantly reducing latency while improving QoE. Additionally, the model consistently outperformed traditional resource allocation algorithms, especially in high-demand scenarios. The procedure is demonstrated in Fig. 1.

VI. RESULTS ANALYSIS

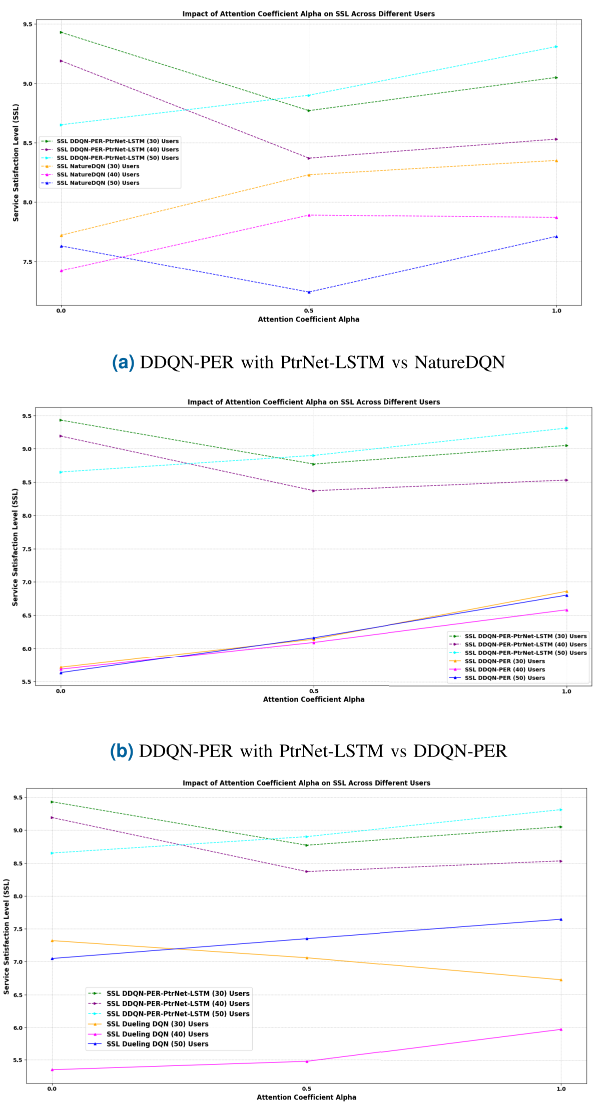
In this section, we evaluate the performance of the proposed DDQN-PER with the PtrNet-LSTM model across key metrics, including SSL, QoE, attention coefficient impact on network performance, and reward optimization in adaptive resource allocation. Each metric is analyzed in detail to demonstrate the model's advantages over baseline methods in managing resources effectively, particularly under varying user densities and dynamic network conditions.

A. IMPACT ON SERVICE SATISFACTION LEVEL (SSL)

This part examines the efficacy of the proposed DDQN-PER with PtrNet-LSTM in sustaining a high SSL across varying user densities and attention coefficients (α). The attention coefficient values (0, 0.5, 0.75, and 1) were assessed to determine their effect on SSL. As depicted in Fig. 2a, the DDQN-PER model with PtrNet-LSTM consistently outperforms the NatureDQN model in terms of SSL across varying values of the attention coefficient α and different user densities. With increasing α, which shifts the focus from maximizing service satisfaction to optimizing system capacity, both models exhibit improvements in SSL. However, the DDQN-PER with PtrNet-LSTM maintains consistently higher and more stable SSL, particularly in scenarios with higher user counts (30, 40, and 50 users). This indicates that the DDQN-PER with PtrNet-LSTM is particularly adept at handling resource allocation under dynamic network conditions. Conversely, although the NatureDQN model shows improvement with higher α values, its performance is more variable, especially as the number of users increases, suggesting limitations in balancing system capacity with user satisfaction. These results underscore the effectiveness of the DDQN-PER with PtrNet-LSTM in efficiently managing resources, making it a promising approach for complex network environments where maintaining high QoE is essential.

The comparison between the DDQN-PER with PtrNet-LSTM and the standard DDQN-PER, illustrated in Fig. 2b, reveals that the PtrNet-LSTM-enhanced approach consistently achieves higher SSL across all tested α values and user counts. For instance, while the standard DDQN-PER's SSL ranges from 5.71 to 6.86 for 30 users, the DDQN-PER with PtrNet-LSTM significantly outperforms it, with SSL values ranging from 8.77 to 9.43. This trend continues as the user count increases, with the PtrNet-LSTM model maintaining superior SSL even under more demanding conditions with 50 users, where it achieves SSL values between 8.65 and 9.31 compared to the standard model's 5.63 to 6.80. These findings suggest that integrating PtrNet-LSTM into the DDQN-PER framework enhances its capability to manage resources effectively, particularly in scenarios where balancing service satisfaction with system capacity is critical. The consistently better performance of the PtrNet-LSTM model across varying conditions highlights its robustness and suitability for deployment in complex network environments.

In particular, the proposed DDQN-PER with PtrNet-LSTM framework excels in maintaining a high SSL across varying user densities, outperforming alternative models such as DDQN-PER with PtrNet-LSTM and Dueling DQN. As shown in Fig. 2c With 30 users, the SSL achieved by DDQN-PER with PtrNet-LSTM ranges from 8.77 to 9.43, significantly higher than Dueling DQN's range of 6.72 to 7.32. This advantage continues with 40 users, where SSL

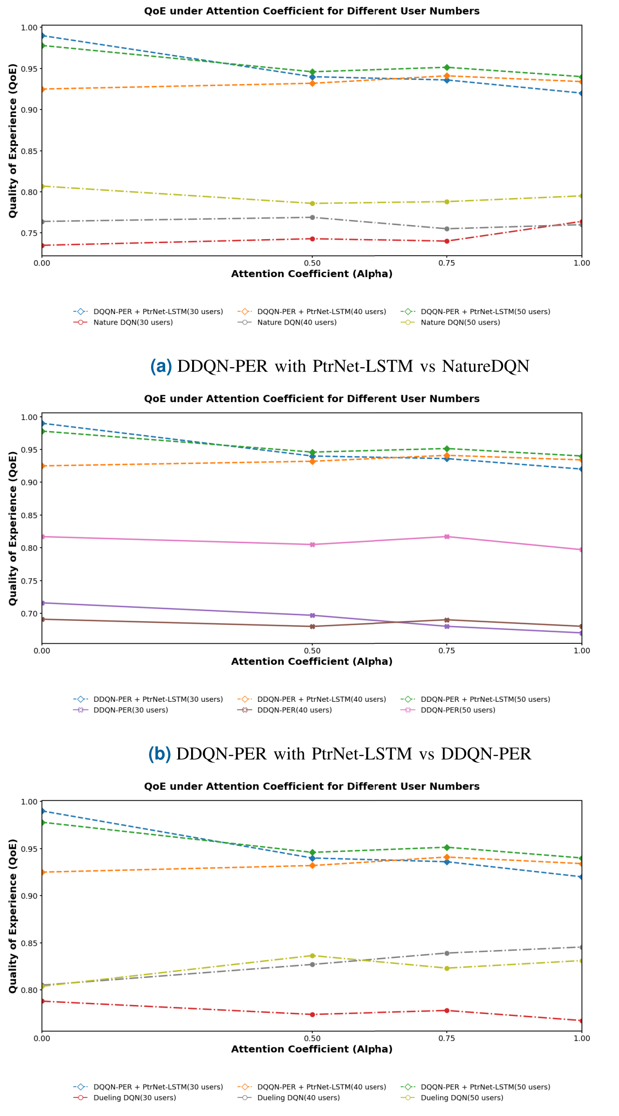


(a) DDQN-PER with PtrNet-LSTM vs NatureDQN

(b) DDQN-PER with PtrNet-LSTM vs DDQN-PER

(c) DDQN-PER with PtrNet-LSTM vs Dueling DQN

FIGURE 2. SSL variation with PtrNet-LSTM versus other state-of-the-art methods for resource allocation under different alpha values and user connectivity.



(a) DDQN-PER with PtrNet-LSTM vs NatureDQN

(b) DDQN-PER with PtrNet-LSTM vs DDQN-PER

(c) DDQN-PER with PtrNet-LSTM vs Dueling DQN

FIGURE 3. QoE variation with PtrNet-LSTM versus other state-of-the-art methods for resource allocation under different alpha values and user connectivity.

values for DDQN-PER with PtrNet-LSTM vary between 8.37 and 9.19, compared to the lower range of 5.35 to 5.97 for Dueling DQN. Even as the user count increases to 50, DDQN-PER with PtrNet-LSTM sustains high SSL values from 8.65 to 9.31, while Dueling DQN only reaches 7.05 to 7.64. These results demonstrate that DDQN-PER with PtrNet-LSTM more effectively handles the complexities of resource allocation under high network load than Dueling DQN. The PtrNet-LSTM's ability to prioritize critical learning events and account for temporal dependencies allows it to adapt efficiently to dynamic resource demands. This consistent performance advantage makes the DDQN-PER with PtrNet-LSTM framework particularly suited for applications requiring high user satisfaction in dynamic network environments, highlighting its scalability and resilience.

B. EFFECT ON QUALITY OF EXPERIENCE (QoE)

This section analyzes the QoE performance, particularly in contexts of elevated user density and changing network requirements. QoE is assessed by analyzing user satisfaction levels under varying network loads. The performance analysis of the DDQN-PER model, as compared to NatureDQN and the enhanced DDQN-PER with PtrNet-LSTM, reveals significant disparities, particularly as user load increases. As shown in Figure 3, using DDQN-PER during the initial stages of user expansion results in a notable reduction in performance, especially concerning the QoE. This decline is primarily attributed to DDQN-PER's limited capacity to manage dynamic, user-driven environments that require decision-making based on temporal dependencies.

TABLE 3. Comparison of QoE with and without PtrNet-LSTM across different user counts and attention coefficients.

Users	α	DDQN-PER + PtrNet-LSTM	DDQN-PER	NatureDQN	Dueling DQN
30	0	**0.990**	0.716	0.735	0.788
	0.5	**0.940**	0.697	0.743	0.7739
	0.75	**0.936**	0.680	0.740	0.778
	1	**0.920**	0.670	0.764	0.7673
40	0	**0.925**	0.691	0.764	0.805
	0.5	**0.932**	0.680	0.769	0.827
	0.75	**0.941**	0.690	0.755	0.839
	1	**0.934**	0.680	0.760	0.8454
50	0	**0.978**	0.817	0.807	0.8038
	0.5	**0.946**	0.805	0.786	0.8364
	0.75	**0.951**	0.817	0.788	0.823
	1	**0.940**	0.797	0.795	0.831

For instance, as presented in Table 3, when the number of users increases to 50, the QoE without PtrNet-LSTM integration drops to 0.817 at $\alpha = 0$, compared to a significantly higher QoE of 0.978 with PtrNet-LSTM. In contrast, NatureDQN performs even less effectively under similar conditions, with a QoE of 0.807 at $\alpha = 0$ for 50 users, further highlighting the limitations of traditional models in handling increased user demands. Systems incorporating PtrNet-LSTM demonstrate superior scalability and resilience, consistently maintaining higher QoE levels as user traffic intensifies. Across all user scenarios, PtrNet-LSTM consistently enhances QoE. For example, with 40 users at $\alpha = 0.75$, the QoE with PtrNet-LSTM reaches 0.941, whereas without PtrNet-LSTM, it significantly drops to 0.690, and NatureDQN performs similarly poorly with a QoE of 0.755. Furthermore, as the user count increases, PtrNet-LSTM plays an increasingly critical role in optimizing system performance, particularly in environments with larger user populations, thereby enhancing the overall user experience. These findings, derived from experimental data involving 30, 40, and 50 users as detailed in Table 3, underscore the importance of integrating PtrNet-LSTM to ensure robust and scalable performance in environments with growing user demands. The results suggest that while DDQN-PER is a reliable model, the integration of PtrNet-LSTM offers substantial performance enhancements. Meanwhile, NatureDQN, although functional, consistently underperforms in comparison, particularly in complex, high-demand environments, making the case for advanced model integrations like PtrNet-LSTM even stronger. Compared to Dueling DQN, the DDQN-PER model with PtrNet-LSTM shows stronger performance across different user densities and α settings. Its higher QoE scores across varying user loads and α values suggest that DDQN-PER with PtrNet-LSTM is better suited for adapting to changing network conditions. For example, with 30 users and an α of 0, DDQN-PER with PtrNet-LSTM achieves a QoE of 0.990, outperforming Dueling DQN under the same conditions. This advantage comes largely from PtrNet-LSTM, which helps DDQN-PER detect temporal patterns and focus on key learning events. Dueling DQN, on the other hand, separates value from advantage to improve decisions but lacks PtrNet-LSTM's flexibility. As a result, DDQN-PER with PtrNet-LSTM is more suitable for applications that need high user satisfaction in fluctuating network conditions.

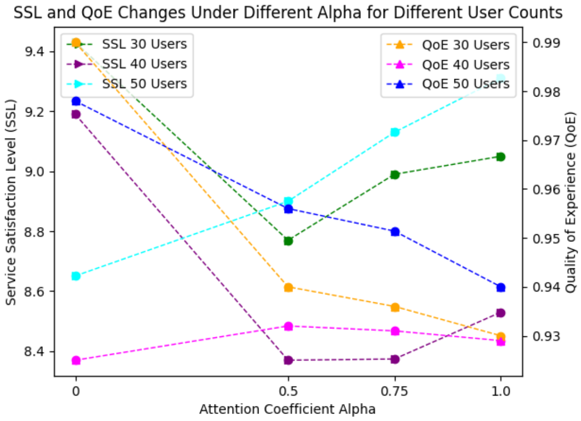


FIGURE 4. Different attention coefficients impact SSL and QoE with different users.

C. ANALYSIS OF ATTENTION COEFFICIENTS ON NETWORK PERFORMANCE

Spectrum efficiency, measured in bits per second per Hertz (bps/Hz), is an essential parameter for assessing the proposed framework's resource use. The research examines the influence of the attention coefficient α on SSL and QoE. The attention coefficient was evaluated at various levels (0, 0.50, 0.75, and 1) to detect changes in these performance indicators. The findings, as indicated in Fig. 4, show that augmenting the attention coefficient α enhances both SSL and QoE measurements.

The implementation of the DDQN-PER with PtrNet-LSTM model demonstrates a significant impact on system performance, particularly when varying the attention coefficient α. As illustrated in Figure 5, increasing the value of α enhances session success rates and user experiences, thereby boosting overall system performance. This improvement is evident in the observed rise in spectrum efficiency (bps/Hz) as α increases, reflecting the model's ability to optimize resource allocation effectively.

Specifically, the prioritization of radio resources for eMBB services becomes more pronounced with higher α levels,

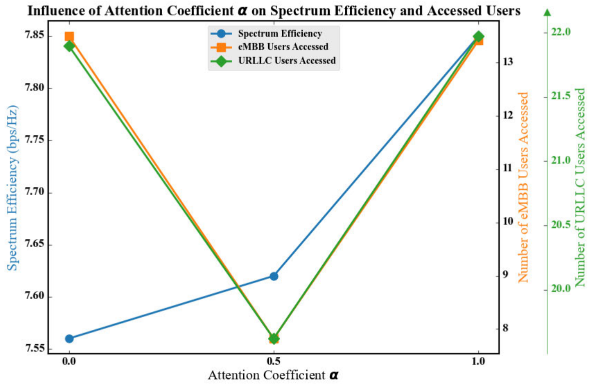
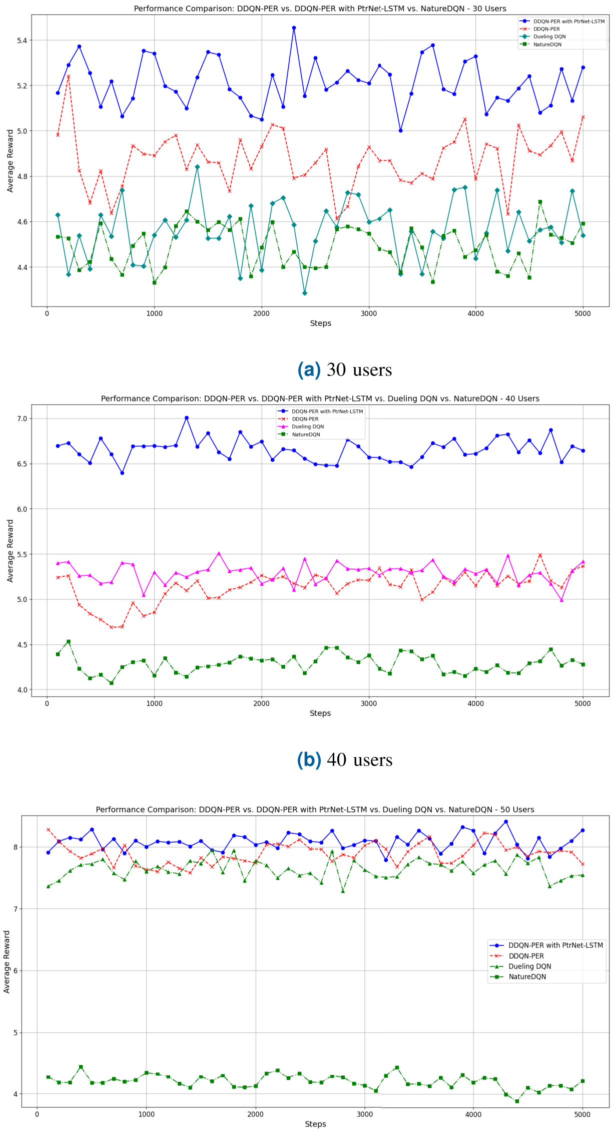


FIGURE 5. Effect of attention coefficient α on spectrum efficiency and user connectivity.

as indicated by the consistent increase in the number of eMBB users accessed. Initially, the number of accessed URLLC users also rises, indicating that the DDQN-PER with PtrNet-LSTM model efficiently manages resource availability for URLLC services. However, as α continues to increase, the focus shifts towards maximizing eMBB service access, highlighting a trade-off in resource allocation between different service types. On the other hand, reducing α facilitates a more equitable distribution of resources across users, while still maintaining high success rates and positive user experiences. This balance is essential for maximizing system capacity and ensuring user satisfaction across varying service demands. These findings underscore the importance of optimizing α within the DDQN-PER with PtrNet-LSTM framework to enhance spectrum efficiency and meet the diverse needs of eMBB and URLLC users effectively.

D. REWARD OPTIMIZATION IN ADAPTIVE RESOURCE ALLOCATION WITH DDQN-PER AND PtrNet-LSTM INTEGRATION

The study evaluates reward performance metrics to assess the model's effectiveness in resource allocation under dynamic network conditions. Specifically, it compares the performance of NatureDQN, DDQN-PER, Dueling DQN, and DDQN-PER enhanced with PtrNet-LSTM over a 5000-step period, as illustrated in Fig. 6. NatureDQN, serving as the baseline model, consistently underperforms compared to the other two models, yielding the lowest average rewards throughout the evaluation period. This underperformance becomes particularly evident in complex environments with an increasing number of users, where NatureDQN struggles with stability and reward optimization. The limitations of NatureDQN underscore its inadequacy in environments that require higher adaptability and effective reward maximization. While DDQN-PER shows improved performance over NatureDQN, it still exhibits variability in its reward trajectory, suggesting that although DDQN-PER is more capable than NatureDQN, it encounters challenges in maintaining consistent learning and stability as the environment's complexity increases, especially with more users. However, the integration of PtrNet-LSTM with DDQN-PER significantly boosts performance across all key metrics. The results demonstrate that the combined model consistently outperforms the standard DDQN-PER. The graph illustrates that DDQN-PER with PtrNet-LSTM achieves higher average rewards and follows a more gradual and consistent growth trajectory, indicating enhanced learning and adaptation capabilities. This integration is particularly effective in scenarios that require long-term learning, making the system more proficient in maximizing rewards in environments with numerous users. As the number of users increases, DDQN-PER with PtrNet-LSTM continues to show a distinct performance advantage right from the initial steps and widens the gap as training progresses towards 5000 steps, reinforcing the robustness of the research findings. Overall,



(a) 30 users

(b) 40 users

(c) 50 users

FIGURE 6. Comparison of the performance reward of Nature DQN, DDQN-PER, Dueling DQN and DDQN-PER with PtrNet-LSTM for different numbers of users.

the results indicate that DDQN-PER combined with PtrNet-LSTM is a more effective and efficient approach for handling the complexities of multi-user systems, offering superior stability, scalability, and reward optimization. This enhanced model surpasses NatureDQN in all respects and sets a new benchmark for performance in environments that demand high adaptability and advanced reinforcement learning capabilities.

Regarding optimization of rewards, stability, and flexibility under various network loads, the performance comparison between DDQN-PER with PtrNet-LSTM and Dueling DQN demonstrates a clear advantage for DDQN-PER with PtrNet-LSTM. The results show that DDQN-PER with PtrNet-LSTM is more stable and resilient, with higher average payouts and fewer fluctuations across all situations examined (30, 40, and 50 users). However, Dueling DQN exhibits more volatility and lower average rewards under significant user loads, which might be a sign of dynamic resource allocation restrictions. These findings suggest that PtrNet-LSTM integrated with DDQN-PER improves resource allocation, which makes it an excellent choice for complicated 5G network slicing applications.

VII. DISCUSSION

This study introduces a hybrid reinforcement learning framework that integrates DDQN-PER and PtrNet-LSTM to address dynamic resource allocation challenges in 5G RAN slicing. Experimental results demonstrate that the proposed model effectively balances SSL and QoE across varied network conditions, achieving notable improvements over traditional baseline models.

A. KEY FINDINGS AND ANALYSIS

The findings reveal that modifying the attention coefficient, α, significantly impacts both SSL and QoE, allowing an adaptive balance between eMBB and URLLC services. Increasing α directs more resources towards eMBB, thereby enhancing spectral efficiency and connectivity, while reducing α prioritizes the latency requirements of URLLC services. This adaptability is essential in environments with fluctuating network loads and diverse user densities, demonstrating the model's capability to optimize resource utilization across heterogeneous traffic demands.

B. ADVANTAGES OVER BASELINE MODELS

The enhanced model, incorporating DDQN-PER with PtrNet-LSTM, demonstrates clear advantages over conventional models such as NatureDQN, particularly under high network loads. The PtrNet-LSTM component improves temporal learning, enabling the model to adjust its resource allocation strategies based on historical usage patterns. This results in consistently higher SSL and QoE performance in high-demand scenarios, highlighting the framework's effectiveness in dynamic resource allocation. PtrNet-LSTM's ability to handle temporal dependencies also strengthens the model's scalability and stability in complex, multi-user environments.

C. LIMITATIONS AND POTENTIAL EXTENSIONS

The PtrNet-LSTM model requires an expansion for further validation. However, it has shown promising performance under diverse user densities and traffic demands. Experiments should be conducted with varying network circumstances and load intensities to assess its resilience further. Improving factors like energy efficiency and user fairness might provide further insights into its suitability for real-world applications. There is a lack of testing on the model's scalability to more significant, more diverse network topologies, which is a problem since next-generation networks often include large-scale, multi-cell situations. On top of that, situations with limited resources could struggle with PtrNet-LSTM because of its computational complexity.

D. FUTURE RESEARCH DIRECTIONS

Further studies may assess the efficacy of this framework under diverse load conditions and other network topologies to enhance its use. The best possible outcomes may be attained in dynamic network environments via the use of multi-agent frameworks or advanced hybrid reinforcement learning techniques. Research may test the framework in simulated or real-world settings to get beyond these restrictions, considering hardware limits, interference patterns, and cellular interactions. Investigating computational advances for PtrNet-LSTM and implementing adaptive interference management methodologies are necessary to improve the framework for 6G applications. These improvements would provide the groundwork for future adaptable and scalable networks. If these problems could be resolved, the credibility of the framework would be increased, and methods for managing resources in wireless networks would advance.

VIII. CONCLUSION

This study presents a novel solution for optimizing resource allocation in a RAN environment by integrating a PtrNet-LSTM with the DDQN-PER algorithm. The primary objective of this research was to enhance the SSL and QoE while maintaining system capacity under varying network conditions. The proposed solution achieves this by dynamically adjusting the attention coefficient in real-time, effectively balancing SSL and QoE based on current network conditions. Simulation results, implemented using Python frameworks, demonstrated significant improvements in SSL and QoE metrics across a wide range of user loads, validating the approach's effectiveness. Furthermore, the study underscored the importance of considering different user categories, such as eMBB and URLLC users, in optimizing system throughput and resource allocation. By addressing these distinct user needs, the solution ensures a more efficient and fair distribution of network resources.

The integration of PtrNet-LSTM with DDQN-PER offers a promising and scalable approach for optimizing resource allocation in RAN environments. This approach meets the research objectives by delivering enhanced network performance and adaptability in dynamic and high-demand scenarios. This approach not only meets but exceeds the original objectives by providing a robust and adaptable framework, with potential applications extending to future wireless networks, including 6G.

REFERENCES

[1] P. Jonsson, ''Ericsson mobility report (Review. Part II),'' *Synchroinfo J.*, vol. 8, no. 4, pp. 33–41, 2022, doi: 10.36724/2664-066x-2022-8-4-33-41.

[2] S. Ebrahimi, F. Bouali, and O. C. L. Haas, ''Resource management from single-domain 5G to end-to-end 6G network slicing: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 26, no. 4, pp. 2836–2866, 4th Quart., 2024, doi: 10.1109/COMST.2024.3390613.

[3] O. O. Erunkulu, A. M. Zungeru, C. K. Lebekwe, M. Mosalaosi, and J. M. Chuma, ''5G mobile communication applications: A survey and comparison of use cases,'' *IEEE Access*, vol. 9, pp. 97251–97295, 2021, doi: 10.1109/ACCESS.2021.3093213.

[4] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, ''On radio access network slicing from a radio resource management perspective,'' *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, Oct. 2017, doi: 10.1109/MWC.2017.1600220WC.

[5] M. Zangooei, N. Saha, M. Golkarifard, and R. Boutaba, ''Reinforcement learning for radio resource management in RAN slicing: A survey,'' *IEEE Commun. Mag.*, vol. 61, no. 2, pp. 118–124, Feb. 2023, doi: 10.1109/MCOM.004.2200532.

[6] A. M. Nagib, H. Abou-Zeid, and H. S. Hassanein, ''Transfer learning-based accelerated deep reinforcement learning for 5G RAN slicing,'' in *Proc. IEEE 46th Conf. Local Comput. Netw. (LCN)*, Oct. 2021, pp. 249–256, doi: 10.1109/LCN52139.2021.9524965.

[7] T. Ma, Y. Zhang, F. Wang, D. Wang, and D. Guo, ''Slicing resource allocation for eMBB and URLLC in 5G RAN,'' *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–11, Jan. 2020, doi: 10.1155/2020/6290375.

[8] X. Li, R. Ni, J. Chen, Y. Lyu, Z. Rong, and R. Du, ''End-to-end network slicing in radio access network, transport network and core network domains,'' *IEEE Access*, vol. 8, pp. 29525–29537, 2020, doi: 10.1109/ACCESS.2020.2972105.

[9] D. Marabissi and R. Fantacci, ''Highly flexible RAN slicing approach to manage isolation, priority, efficiency,'' *IEEE Access*, vol. 7, pp. 97130–97142, 2019, doi: 10.1109/ACCESS.2019.2929732.

[10] A. De Domenico, Y.-F. Liu, and W. Yu, ''Optimal computational resource allocation and network slicing deployment in 5G hybrid C-RAN,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6, doi: 10.1109/ICC.2019.8762089.

[11] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, ''Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks,'' *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019, doi: 10.1109/TWC.2019.2933417.

[12] Y. Ren, S. Guo, B. Cao, and X. Qiu, ''End-to-end network SLA quality assurance for C-RAN: A closed-loop management method based on digital twin network,'' *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4405–4422, May 2024, doi: 10.1109/TMC.2023.3291012.

[13] K. Zheng, R. Luo, X. Liu, J. Qiu, and J. Liu, ''Distributed DDPG-based resource allocation for age of information minimization in mobile wireless-powered Internet of Things,'' *IEEE Internet Things J.*, vol. 11, no. 17, pp. 29102–29115, Sep. 2024.

[14] X. Liu, J. Xu, K. Zheng, G. Zhang, J. Liu, and N. Shiratori, ''Throughput maximization with an AoI constraint in energy harvesting D2D-enabled cellular networks: An MSRA-TD3 approach,'' *IEEE Trans. Wireless Commun.*, vol. 24, no. 2, pp. 1448–1466, Feb. 2025.

[15] J. Wang, C. Xu, Y. Huangfu, R. Li, Y. Ge, and J. Wang, ''Deep reinforcement learning for scheduling in cellular networks,'' in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019, pp. 1–6, doi: 10.1109/wcsp.2019.8927868.

[16] D. Wu, Z. Zhang, S. Wu, J. Yang, and R. Wang, ''Biologically inspired resource allocation for network slices in 5G-enabled Internet of Things,'' *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9266–9279, Dec. 2019, doi: 10.1109/JIOT.2018.2888543.

[17] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, ''CellSlice: Cellular wireless resource slicing for active RAN sharing,'' in *Proc. 5th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2013, pp. 1–10, doi: 10.1109/COMSNETS.2013.6465548.

[18] I. Shayea, M. Ergen, M. Hadri Azmi, S. Aldirmaz Çolak, R. Nordin, and Y. I. Daradkeh, ''Key challenges, drivers and solutions for mobility management in 5G networks: A survey,'' *IEEE Access*, vol. 8, pp. 172534–172552, 2020, doi: 10.1109/ACCESS.2020.3023802.

[19] P. Fan, J. Zhao, and I. C.-Lin, ''5G high mobility wireless communications: Challenges and solutions,'' *China Commun.*, vol. 13, no. 2, pp. 1–13, Feb. 2016, doi: 10.1109/CC.2016.7833456.

[20] C.-X. Wang, A. Ghazal, B. Ai, Y. Liu, and P. Fan, ''Channel measurements and models for high-speed train communication systems: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 974–987, 2nd Quart., 2016, doi: 10.1109/COMST.2015.2508442.

[21] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno, ''Flexible resource block allocation to multiple slices for radio access network slicing using deep reinforcement learning,'' *IEEE Access*, vol. 8, pp. 68183–68198, 2020, doi: 10.1109/ACCESS.2020.2986050.

[22] J. Li and X. Zhang, ''Deep reinforcement learning-based joint scheduling of eMBB and URLLC in 5G networks,'' *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1543–1546, Sep. 2020, doi: 10.1109/LWC.2020.2997036.

[23] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, ''The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility,'' *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sep. 2020, doi: 10.1109/LCOMM.2020.3001227.

[24] S. Choi, S. Choi, G. Lee, S.-G. Yoon, and S. Bahk, ''Deep reinforcement learning for scalable dynamic bandwidth allocation in RAN slicing with highly mobile users,'' *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 576–590, Jan. 2024, doi: 10.1109/TVT.2023.3302416.

[25] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Pérez, ''DeepCog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting,'' *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 361–376, Feb. 2020, doi: 10.1109/JSAC.2019.2959245.

[26] C. Zhang, M. Fiore, and P. Patras, ''Multi-service mobile traffic forecasting via convolutional long short-term memories,'' in *Proc. IEEE Int. Symp. Meas. Netw.*, Jul. 2019, pp. 1–6, doi: 10.1109/IWMN.2019.8804984.

[27] J.-B. Monteil, J. Hribar, P. Barnard, Y. Li, and L. A. DaSilva, ''Resource reservation within sliced 5G networks: A cost-reduction strategy for service providers,'' in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6, doi: 10.1109/ICCWORKSHOPS49005.2020.9145374.

[28] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, ''Intelligent resource scheduling for 5G radio access network slicing,'' *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7691–7703, Aug. 2019, doi: 10.1109/TVT.2019.2922668.

[29] T. V. K. Buyakar, H. Agarwal, B. R. Tamma, and A. A. Franklin, ''Resource allocation with admission control for GBR and delay QoS in 5G network slices,'' in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2020, pp. 213–220, doi: 10.1109/COMSNETS48256.2020.9027310.

[30] A. T. Z. Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, ''Experienced deep reinforcement learning with generative adversarial networks (GANs) for model-free ultra reliable low latency communication,'' *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 884–899, Feb. 2021, doi: 10.1109/TCOMM.2020.3031930.

[31] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, ''Meta-reinforcement learning for trajectory design in wireless UAV networks,'' in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6, doi: 10.1109/GLOBECOM42002.2020.9322414.

[32] H. P. Phyu, R. Stanica, and D. Naboulsi, ''Multi-slice privacy-aware traffic forecasting at RAN level: A scalable federated-learning approach,'' *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 4, pp. 5038–5052, Dec. 2023, doi: 10.1109/TNSM.2023.3267725.

[33] S. Liu, X. Hu, Y. Wang, G. Cui, and W. Wang, ''Distributed caching based on matching game in LEO satellite constellation networks,'' *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 300–303, Feb. 2018, doi: 10.1109/LCOMM.2017.2771434.

[34] X. Zhu, C. Jiang, L. Kuang, and Z. Zhao, ''Cooperative multilayer edge caching in integrated satellite-terrestrial networks,'' *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 2924–2937, May 2022, doi: 10.1109/TWC.2021.3117026.

[35] D. Jiang, F. Wang, Z. Lv, S. Mumtaz, S. Al-Rubaye, A. Tsourdos, and O. Dobre, "QoE-aware efficient content distribution scheme for satellite-terrestrial networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 1, pp. 443–458, Jan. 2023, doi: 10.1109/TMC.2021.3074917.

[36] G. Chen, S. Qi, F. Shen, Q. Zeng, and Y.-D. Zhang, "Information-aware driven dynamic LEO–RAN slicing algorithm joint with communication, computing, and caching," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 5, pp. 1044–1062, May 2024, doi: 10.1109/JSAC.2024.3365893.

[37] R. Mijumbi, J. Gorricho, J. Serrat, M. Claeys, F. D. Turck, and S. Latré, "Design and evaluation of learning algorithms for dynamic resource management in virtual networks," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.

[38] L. Le, T. N. Nguyen, K. Suo, and J. He, "Efficient embedding VNFs in 5G network slicing: A deep reinforcement learning approach," 2022, *arXiv:2207.11822*.

[39] M. Masoudi, Ö. T. Demir, J. Zander, and C. Cavdar, "Energy-optimal end-to-end network slicing in cloud-based architecture," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 574–592, 2022.

[40] A. Aijaz, "Hap—SliceR: A radio resource slicing framework for 5G networks with haptic communications," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2285–2296, Sep. 2018.

[41] A. Nassar and Y. Yilmaz, "Reinforcement learning for adaptive resource allocation in fog RAN for IoT with heterogeneous latency requirements," *IEEE Access*, vol. 7, pp. 128014–128025, 2019.

[42] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5G infrastructure markets: The business of network slicing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM -* , May 2017, pp. 1–9.

[43] K. Xiong, S. Samuel Rene Adolphe, G. O. Boateng, G. Liu, and G. Sun, "Dynamic resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Access*, vol. 7, pp. 115440–115453, 2019.

[44] H. Bai, Y. Zhang, Z. Zhang, and S. Yuan, "Latency equalization policy of end-to-end network slicing based on reinforcement learning," *IEEE Trans. Netw. Service Manage.*, vol. 20, no. 1, pp. 88–103, Mar. 2023, doi: 10.1109/TNSM.2022.3210012.

[45] V. Niemi and K. Nyberg, "3GPP algorithm specification principles," in *Universal Mobile Telecommunications System Security*. Hoboken, NJ, USA: Wiley, 2003, pp. 131–133. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0470091584.ch5

[46] Z. Yang, W. Xu, J. Shi, H. Xu, and M. Chen, "Association and load optimization with user priorities in load-coupled heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 324–338, Jan. 2018, doi: 10.1109/TWC.2017.2765322.

[47] J. Kwak, L. B. Le, H. Kim, and X. Wang, "Two time-scale edge caching and BS association for power-delay tradeoff in multi-cell networks," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5506–5519, Aug. 2019, doi: 10.1109/TCOMM.2019.2913409.

[48] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.

[49] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver, "Distributed prioritized experience replay," 2018, *arXiv:1803.00933*.

[50] Q. Wang and C. Tang, "Deep reinforcement learning for transportation network combinatorial optimization: A survey," *Knowl.-Based Syst.*, vol. 233, Dec. 2021, Art. no. 107526, doi: 10.1016/j.knosys.2021.107526.

[51] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour d'horizon," *Eur. J. Oper. Res.*, vol. 290, no. 2, pp. 405–421, Apr. 2021, doi: 10.1016/j.ejor.2020.07.063.

[52] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proc. Adv. neural Inf. Process. Syst.*, vol. 289, 2015, pp. 1–9.

[53] C. Wang, L. Ma, R. Li, T. S. Durrani, and H. Zhang, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. 7, pp. 101441–101452, 2019, doi: 10.1109/ACCESS.2019.2929430.

[54] G. Rjoub, J. Bentahar, O. A. Wahab, and A. S. Bataineh, "Deep smart scheduling: A deep learning approach for automated big data scheduling over the cloud," in *Proc. 7th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2019, pp. 189–196, doi: 10.1109/ficloud.2019.00034.

[55] M. Hassan, H. Chen, and Y. Liu, "DEARS: A deep learning based elastic and automatic resource scheduling framework for cloud applications," in *Proc. IEEE Intl Conf Parallel Distrib. Process. Appl., Ubiquitous Comput. Commun., Big Data Cloud Comput., Social Comput. Netw., Sustain. Comput. Commun. (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, Dec. 2018, pp. 541–548, doi: 10.1109/BDCLOUD.2018.00086.

[56] D. Yi, X. Zhou, Y. Wen, and R. Tan, "Toward efficient compute-intensive job allocation for green data centers: A deep reinforcement learning approach," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 634–644, doi: 10.1109/ICDCS.2019.00069.

[57] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 114–119, Jun. 2019, doi: 10.1109/MCOM.2019.1800155.

ZAHRAA ZAKARIYA SALEH received the B.Sc. degree in information technology from Ishik University, Iraq, in 2016, and the M.Sc. degree in computer systems engineering from the University of Kurdistan Hewlêr, Iraq, in 2019. She is currently pursuing the Ph.D. degree in electronic and electrical engineering with Brunel University of London. She was an Assistant Lecturer at different universities in Iraq. Her research interests include 5G and beyond slicing networks, quality of experience/QoE enhancement, and reinforcement learning.

MAYSAM F. ABBOD (Senior Member, IEEE) received the Ph.D. degree in control engineering from The University of Sheffield, in 1992. From 1993 to 2006, he was with the Department of Automatic Control and Systems Engineering, The University of Sheffield, as a Research Associate and a Senior Research Fellow. His developed systems were applied to industrial and biomedical modeling and computer control of manufacturing systems. His main research interests include intelligent systems for modeling, control, and optimization.

RAJAGOPAL NILAVALAN (Senior Member, IEEE) received the B.Sc. (Eng.) degree in electrical and electronics engineering from the University of Peradeniya, Sri Lanka, in 1995, and the Ph.D. degree in radio frequency systems from the University of Bristol, Bristol, U.K., in 2001. From 1999 to 2005, he was a Researcher with the Centre for Communications Research (CCR), University of Bristol, U.K. At Bristol, his research involved theoretical and practical analyses of post-reception synthetic focusing concepts for near-field imaging and research on numerical FDTD techniques. Since 2005, he has been with the Department of Electronics and Computer Engineering, Brunel University of London, U.K., where he is currently a Reader in wireless communications. His main research interests include antennas and propagation, microwave circuit designs, numerical electromagnetic modeling, and wireless communication systems. He has published over 100 papers in journals and international conferences in his research areas. He was a member of the European Commission, Network of Excellence on Antennas, from 2002 to 2005, and a member of the IET.