QoS-Aware Content Delivery in 5G-enabled Edge Computing: Learning-based Approaches

Erfan Farhangi Maleki, *Student member, IEEE,* Weibin Ma, *Student member, IEEE,* Lena Mashayekhy, *Senior Member, IEEE,* and Humberto La Roche, *Member, IEEE*

Abstract—The increasing demand for high-volume multimedia services through mobile user equipment (UEs) has imposed a significant burden on mobile networks. To cope with this growth in demand, it is necessary to extend the 5G network's ability to meet quality-of-service (QoS) requirements. The integration of Multi-access Edge Computing (MEC) with 5G technology, 5G-MEC, emerges as a pivotal solution, offering ultra-low latency, ultra-high reliability, and continuous connectivity to support various latency-sensitive applications for UEs. Despite these advancements, the mobility of UEs introduces significant spatio-temporal uncertainties, posing a major challenge on optimizing content delivery routes and directly impacting both latency and service continuity for UEs. Addressing this challenge necessitates suitable approaches for selecting optimal 5G-MEC components, with the goal of minimizing latency and reducing the frequency of handovers, ultimately ensuring a seamless content delivery experience. This paper proposes two learning-based approaches to tackle the problem of 5G-MEC component selection to facilitate QoS-aware content delivery in the absence of complete information about the dynamics of the 5G-MEC environment. First, we design an online sequential decision-making approach, called QCS-MAB, to decide on the content delivery routes in real-time while achieving a bounded performance. We then propose a deep learning approach, called QCS-DNN, to efficiently solve large-scale 5G-MEC component selection problems. We evaluate the effectiveness of our proposed approaches through extensive experiments using a real-world dataset. The results demonstrate that both QCS-MAB and QCS-DNN achieve near-optimal latency and significantly reduced handover times, significantly enhancing the 5G-MEC content delivery experience.

Index Terms—Multi-access Edge Computing, 5G, Mobility, Content Delivery, Online sequential decision-making, Deep Learning

1 Introduction

The proliferation of connected devices is projected to surpass 500 billion by 2030, causing a massive surge in data traffic [1]. The global data traffic is expected to grow 3.5 times between 2022 and 2028, reaching a monthly volume of 472 exabytes (EB) by the end of 2028. Content delivery constitutes a significant portion of this traffic. According to Ericsson [2], video traffic is projected to make up 80 percent of total mobile data traffic by 2028, necessitating strict low-latency and high-speed data transmission to enhance Quality of Service (QoS). This growing data traffic also translates into tangible economic impacts and significantly affects the user experience. For example, a 100ms increase in latency can reduce Amazon's sales by 1%, and Google has reported a 20% decrease in traffic due to an additional 0.5-second delay in search page generation. Similarly, if a trading platform is 5ms slower than its competitors, a broker could lose $4 million in revenues per millisecond [3]. These examples show the crucial need for innovations in network efficiency and the urgency for the improvement of latency issues.

To meet the unprecedented content delivery demand requiring low-latency, Fifth-Generation (5G) networks unfold new opportunities for faster content delivery, promising speeds nearly a hundred times faster than 4G. However, the limited transmission capacity of wireless backhaul links may hinder 5G's ability to cope with the explosive growth of traffic [4]. Traditionally, to cope with the rapidly increasing traffic, frequently accessed content has been collected and cached on the cloud. The stored content can be fetched from the cloud when requested by User Equipments (UEs). However, this solution is not practical for real-time services as it results in high latency when delivering data from the cloud. Therefore, there is a need for a more efficient solution that can provide low-latency content delivery in 5G networks.

Multi-access Edge Computing (MEC) offers a decentralization of the conventional cloud paradigm by positioning computing, storage, and caching closer to UEs at the edge of the network [5], [6]. This shift is instrumental in achieving extremely low latency and high throughput, essential for meeting the rigorous QoS requirements. Content caching and delivery at the MEC helps alleviate the heavy burden on data transmission. MEC significantly degrades duplicated content transmissions from backhaul links, improving experienced latency for users. This leads to a decrease in backhaul capacity demands by up to 35% [7]. Therefore, the European 5G Infrastructure Public Private Partnership (5G PPP) recognized MEC as one of the main technologies for 5G networks [8].

The 5G and MEC integration, known as 5G-enabled MEC or 5G-MEC, combines the advanced capabilities of 5G network components and MEC infrastructure for efficient content delivery. MEC consists of Edge Application Servers

- *E. Farhangi Maleki, W. Ma and L. Mashayekhy are with the Department of Computer and Information Sciences, University of Delaware, Newark, DE, USA 19716*
Email: erfanf@udel.edu, weibinma@udel.edu, mlena@udel.edu
- *H. La Roche is with Cisco Systems, USA*
Email: hlaroche@cisco.com

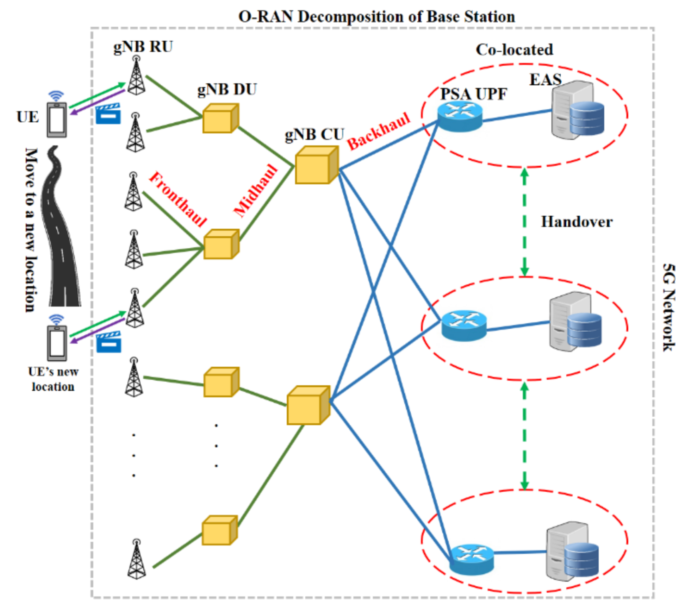


Fig. 1: Framework: 5G-MEC component selection for content delivery.

(EASs), which are localized, small-scale data centers or clusters closer to the users. The 5G architecture, known as 3rd Generation Partnership Project (3GPP) 5G Radio Access Network (RAN) or Next-Generation RAN (NG-RAN), features advanced components structured within the RAN and 5G Core Network (5GC). The RAN is comprised of next-generation base stations, or gNodeBs (gNBs), which are segmented into three integral units: the Radio Unit (RU), which interfaces with mobile devices; the Distributed Unit (DU), which processes data at or near the cell site; and the Centralized Unit (CU), which serves as the control center for multiple DUs. This tri-partite structure can be deployed in various configurations (either co-located or distributed across different locations), accommodating the diverse needs of network operators. The Protocol Data Unit (PDU) Session Anchor (PSA) User Plane Function (UPF), called PSA UPF, is a core network function that plays a key role in building the content routing path from the EAS, where the content is hosted, to a UE requesting the content [9]. Each gNB's connection to a PSA UPF facilitates UE access to cached content on an EAS.

The main challenge in 5G-MEC lies in the *component selection problem*, which involves choosing the most suitable gNB components, PSA UPF, and EAS that ensures the optimal path for UEs to retrieve content rapidly. This is crucial for real-time services and applications. The goal is to determine the most efficient route that minimizes latency from the gNB RU to the EAS hosting the desired content, as illustrated in Fig. 1.

The mobility of users introduces additional complexity in optimizing the content delivery path in 5G-MEC, challenging service continuity [10], [11]. Even though caching at 5G-MEC can offer fast content delivery to nearby users, unpredictable user mobility can negatively impact the caching strategies and the content delivery process, leading to increased experienced latency for users. Therefore, it is important to consider user mobility when selecting the optimal 5G-MEC components. This ensures to determine the efficient routing paths for content delivery, from where the content is cached to users. Selecting the optimal combination of the 5G-MEC components to accommodate such mobility is a complex problem and requires a deep understanding of the network architecture, user mobility, and application requirements. A significant obstacle in this optimization process is the lack of complete information. The current state-of-the-art approaches often result in sub-optimal decisions, leading to poor QoS and system performance.

This paper addresses the 5G-MEC component selection problem for content delivery, considering uncertain user mobility and incomplete information. We first devise a novel integer programming (IP) model to formulate this problem aiming to obtain the optimal solution when we have complete information. Then, we propose two learning-based approaches that aim to learn the optimal components for minimizing the experienced content delivery latency for users throughout their network interaction. Our first proposed approach, QCS-MAB, is an online sequential decision-making approach formulated as Multi-Armed Bandit (MAB) to learn the optimal content routing paths for UEs considering their uncertain mobility. QCS-MAB uses currently observed information to dynamically select a proper routing path for delivering requested contents to UEs at each time slot and reduce experienced latency. Online sequential decision-making is especially ideal when complete information is not available initially and data is received in a sequential manner, making it an effective decision-making technique. However, online sequential decision-making may take time for exploration to reach near-optimal results in large-scale environments. Therefore, we propose a second approach using deep learning, called QCS-DNN, employing a fully connected deep neural network to provide near-optimal solutions in real-time utilizing historical data for massive-scale problems when complete information is unavailable. We perform extensive experiments to evaluate the effectiveness of the proposed approaches using a real-world dataset.

The rest of the paper is organized as follows. In the next section, we present an overview of existing studies in this scope. We formulate the problem in Section 3. We then describe the proposed online sequential decision-making approach, QCS-MAB, in Section 4. We describe our deep learning approach, QCS-DNN, in Section 5). The experimental results are described in Section 6. Finally, we summarize our results and introduce potential future research directions.

2 Related Work

We provide a summary of the most relevant studies, categorize them by their research focus, and identify existing research gaps.

Latency. Several studies have focused on optimizing latency. Solozabal *et al.* [12] introduced a 5G-MEC-based architecture for Mission-Critical Push-to-Talk (MCPTT) services, emphasizing the importance of meeting delay requirements. Guo *et al.* [13] proposed a cross-stratum resource protection approach in fog-computing-based radio over fiber networks (F-RoFN) for 5G services, leveraging software-defined networking for control. Sharghivand *et al.* [14] addressed time-constrained service handoffs in 5G-MEC to maintain QoS, by introducing a path planning approach and payment function. Rahimi *et al.* [15] proposed a hybrid architecture combining technologies such as Device-to-Device (D2D) communication, Massive MIMO, SDN, and NFV, and consisting of MEC and Edge Cloud, to support scalability, reliability, and ultra-low latency. However, it is important to note that the research in this area has been limited, particularly in developing efficient and adaptable solutions for dynamic network environments. Our work expands upon this by focusing on a broader application of latency optimization in 5G-MEC by introducing novel learning-based approaches that address the challenges of rapidly changing network conditions and user mobility.

Other optimization objectives. Beyond addressing latency, the literature has considered various optimization criteria, such as energy consumption, spectrum utilization, and cost-efficiency. Kiani *et al.* [16] tackled the issue of energy consumption by introducing an edge computing-aware NOMA technique. Nadeem *et al.* [17] integrated D2D, MEC, and network slicing to enhance spectrum utilization, performance and scalability. Ning *et al.* [18] focused on a 5G-MEC health monitoring system for the Internet of Medical Things (IoMT) to minimize the system-wide cost. Bishoyi *et al.* [19] considered the interaction between the MEC server and users in a Stackelberg game, addressing joint cost and energy-efficient task offloading for the MEC-enabled healthcare system. Huang *et al.* [4] focused on joint optimization of caching, transcoding, and wireless resource allocation to enhance adaptive video streaming in MEC. Ei *et al.* [20] studied a UAV-enabled MEC system, formulating a joint resource allocation and offloading problem to minimize total energy consumption. Zhong *et al.* [21] introduced a multi-user cost-efficient crowd-assisted delivery and computing framework, utilizing MEC for efficient virtual reality video processing and content delivery. While these studies address various optimization objectives, a significant research gap exists in aligning these objectives with latency minimization and user perspectives on efficiency, especially in highly dynamic and competitive environments characterized by factors such as high mobility, fluctuating network conditions, and competition over resources.

Caching. Content caching is a well-known method to improve content delivery performance in 5G-MEC. Zhang *et al.* [22] proposed a mobility-aware cooperative edge caching architecture for content-centric 5G networks, utilizing edge resources to enhance caching capability. Tang *et al.* [23] introduced a cooperative caching scheme to extend the virtual cache capacity and to minimize delivery delays in user-centric delivery schemes in 5G CDNs. Markakis *et al.* [24] proposed a unified communication, computing, and caching solution for 5G, bringing various functions, services, and contents closer to UEs. Hou *et al.* [25] designed a proactive caching mechanism that leverages transfer learning for predicting content popularity, potentially optimizing cache resources. Huang *et al.* [26] studied the spatio-fine-grained and generalized consequent content delivery services hotspots prediction in ultra-dense 5G networks. However, these models should provide practical solutions for real-time content delivery while dynamically adapt caching

Study	**QoS**	**Content Delivery**	**Edge Computing**	**Hybrid Computing**	**Mobility**	**Learning**	**Multi-User**	**Real Dataset**
Huang *et al.* [4]	✓	✓	✓					
Zhong *et al.* [21]	✓	✓	✓	✓			✓	✓
Zhang *et al.* [22]	✓	✓	✓	✓	✓		✓	
Tang *et al.* [23]	✓	✓	✓				✓	
Markakis *et al.* [24]	✓	✓	✓				✓	
Hou *et al.* [25]	✓	✓	✓				✓	
Huang *et al.* [26]		✓	✓		✓	✓	✓	✓
Blasco *et al.* [27]	✓	✓				✓	✓	
Yu *et al.* [28]	✓	✓	✓	✓		✓	✓	
Ren *et al.* [29]	✓	✓	✓			✓	✓	
Zhai *et al.* [30]	✓		✓	✓		✓	✓	✓
Pang *et al.* [31]		✓	✓			✓	✓	✓
Dong *et al.* [32]	✓		✓			✓	✓	
Chen *et al.* [33]	✓		✓	✓		✓	✓	✓
Qiao *et al.* [34]		✓	✓				✓	✓
Our Study	✓	✓	✓		✓	✓	✓	✓

TABLE 1: Comparison with existing research

strategies to uncertain changes based on user mobility and network changes. Further research is necessary to develop learning-based approaches that can effectively determine the best paths for routing cached content. Our approaches contribute to this domain.

Learning. Machine learning approaches have gained popularity for caching content on the edge under conditions of partial or incomplete information. Blasco *et al.* [27] applied a multi-arm bandit (MAB) approach to learn the popularity distribution of content. Yu *et al.* [28] proposed a deep reinforcement learning approach within an intelligent ultradense edge computing environment for real-time and low-overhead computation offloading and resource allocation. Ren *et al.* [29] utilized deep reinforcement learning agents to select the suitable collaborative computing nodes and a double deep Q-learning approach to guarantee load balancing. Zhai *et al.* [30] studied service deployment in 5G-MEC using deep reinforcement learning, considering user request patterns and resource constraints. Pang *et al.* [31] designed a cooperative edge caching framework with a deep-learning-based caching approach. Dong *et al.* [32] proposed a deep learning framework for improving energy efficiency of ultra-reliable and low-latency communications (URLLC) and delay-tolerant services in MEC. Chen *et al.* [33] developed a URLLC mobile-traffic flow prediction algorithm using an LSTM-based deep-learning algorithm. Qiao *et al.* [34] proposed a distributed resources-efficient federated learning for a proactive content caching policy to enhance content caching efficiency and reduce resource consumption. However, it is important to note that these studies neglect to consider the spatial-temporal dynamics and uncertainties introduced by user mobility, which can significantly impact caching performance and network optimization.

Summary. A detailed comparison of the most relevant studies and our work is presented in Table 1. Note that hybrid computing refers to the use of edge computing in combination with cloud computing. While existing studies provide valuable insights into various aspects of 5G-MEC, there remains a gap in research focusing on the optimization of component selection from EASs to UEs, considering latency, user mobility, and network dynamics. We tackle this challenge by developing learning-based approaches that can efficiently learn component selection for UEs, optimizing content delivery performance. The advantage of our approaches lies in their ability to adaptively learn and optimize routing paths in real-time with partial information considering the complex interplay of network dynamics, user mobility, and QoS requirements.

3 System Model

We consider a time-slotted format, where the time horizon is defined as a series of time slots $\mathcal{T} = \{1, \ldots, T\}$. This temporal structure is vital for analyzing the dynamics of user mobility within the network. We denote a set of UEs by $\mathcal{N} = \{1, \ldots, N\}$. The location of each UE $n \in \mathcal{N}$, at each time is defined by l_n^t. As shown in Fig. 1, the network area under consideration is served by several gNBs denoted by $\mathcal{G} = \{1, \ldots, G\}$, where each gNB is connected to a PSA UPF. This connection allows a UE to acquire content from an EAS, where the content is cached [9]. In other words, the PSA UPF builds the content delivery routing path between each UE and the content on the EAS. Each PSA UPF is co-located with an EAS. We denote a set of co-located pairs of PSA UPFs and EASs as $\mathcal{E} = \{1, \ldots, E\}$, where each pair is indexed by i and collectively referred to as EAS E_i for simplicity in notation.

Communication between UEs and gNBs occurs wirelessly through the RAN, while communication between gNBs and PSA UPFs is facilitated through a wired network, typically a Local Area Network (LAN). This distinction introduces four different types of transmission speeds as indicated below. We define d_{ng}^t as the 5G wireless transmission speed between UE $n \in \mathcal{N}$ and its corresponding gNB $g \in \mathcal{G}$ at time slot t. Additionally, d_{gd} and d_{gc} represent the transmission speed in Fronthaul and Midhaul links, respectively. The speed of transmitting content between gNB g and E_i (Backhaul) is denoted by d_{gi}.

Each UE $n \in \mathcal{N}$, located at l_n^t, requests (requires) a content of size c_n^t at time slot t. To fulfill this request, a set of 5G-MEC components needs to be selected to route the content from an EAS to the UE, taking into account the movement of the UE. As UE moves, the initially chosen route and components may no longer be optimal, and thus necessitating a new set of components to be selected to ensure desired QoS. More specifically, when a UE moves far away from its gNB, a reselection of 5G components is needed to avoid a QoS violation. Dealing with UE mobility can be supported in two ways: through path rerouting to the same EAS or by migrating the content to a new EAS. If the targeted content for UE n is migrated from E_i to E_j at time slot t, the UE will experience a migration latency of s_{ij}^t. Inefficient routing and migration can lead to increased data movement over the network, which increases operation costs of the system and deteriorates the quality of experience of users significantly. Our objective is to develop 5G-MEC component selection approaches that meet the ultra-low latency requirements of UEs.

3.1 Optimization Model (Full Knowledge)

When a priori knowledge of all components of 5G-MEC is available at all time slots, the optimization problem can be formulated in an offline configuration. This serves as a benchmark to evaluate the obtained results and also helps to explain the system model. To formulate the optimization problem mathematically, we first define the decision variables as follows:

- x_{ng}^t is 1 if gNB g (composed of gNB RU, gNB DU, and gNB CU) is allocated to UE n at time slot t, and 0, otherwise.
- y_{nij}^t is 1 if the allocated content for UE n is migrated from E_i to E_j at time slot t, and 0, otherwise.
- z_{ni}^t is 1 if the targeted content for UE n is located at E_i at time slot t, and 0, otherwise.
- k_{ngi}^t is 1 if gNB g and E_i are allocated to UE n at time slot t, and 0, otherwise.

We formulate the 5G-MEC component selection problem as an Integer Program (IP) as follows:

$$\text{Minimize } \mathbf{D} = \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{E}} y_{nij}^t s_{ij}^t + \sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}\sum_{g\in\mathcal{G}}\sum_{i\in\mathcal{E}} k_{ngi}^t c_n^t \left(\frac{1}{d_{ng}^t} + \frac{1}{d_{gd}} + \frac{1}{d_{gc}} + \frac{1}{d_{gi}}\right) \quad (1)$$

Subject to:

$$\sum_{g\in\mathcal{G}} x_{ng}^t = 1 \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \quad (2)$$

$$\sum_{i\in\mathcal{E}} z_{ni}^t = 1 \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \quad (3)$$

$$\sum_{i\in\mathcal{E}}\sum_{j\in\mathcal{E}} y_{nij}^t \leq 1 \qquad \forall n \in \mathcal{N}, t \in \mathcal{T} \quad (4)$$

$$z_{ni}^t + z_{nj}^{t+1} - 1 \leq y_{nij}^{t+1} \qquad \forall n \in \mathcal{N}, i, j \in \mathcal{E}, t \in \mathcal{T} \quad (5)$$

$$x_{ng}^t + z_{ni}^t - 1 \leq k_{ngi}^t \qquad \forall n \in \mathcal{N}, g \in \mathcal{G}, i \in \mathcal{E}, t \in \mathcal{T} \quad (6)$$

$$k_{ngi}^t \leq x_{ng}^t \qquad \forall n \in \mathcal{N}, g \in \mathcal{G}, i \in \mathcal{E}, t \in \mathcal{T} \quad (7)$$

$$k_{ngi}^t \leq z_{ni}^t \qquad \forall n \in \mathcal{N}, g \in \mathcal{G}, i \in \mathcal{E}, t \in \mathcal{T} \quad (8)$$

$$x_{ng}^t \in \{0, 1\} \qquad \forall n \in \mathcal{N}, g \in \mathcal{G}, t \in \mathcal{T} \quad (9)$$

$$y_{nij}^t \in \{0, 1\} \qquad \forall n \in \mathcal{N}, i, j \in \mathcal{E}, t \in \mathcal{T} \quad (10)$$

$$z_{ni}^t \in \{0, 1\} \qquad \forall n \in \mathcal{N}, i \in \mathcal{E}, t \in \mathcal{T} \quad (11)$$

$$k_{ngi}^t \in \{0, 1\} \qquad \forall n \in \mathcal{N}, i \in \mathcal{E}, g \in \mathcal{G}, t \in \mathcal{T} \quad (12)$$

The objective function in Eq. (1) minimizes the total sum of handover latency and content delivery latency over the entire time horizon. The first term calculates the handover latency. The second term calculates the content delivery latency, as indicated by k_{ngi}^t, when a specific routing path is chosen. Content delivery latency is the total sum of latencies incurred while delivering content through different components of the 5G network. For example, $\frac{c_n^t}{d_{ng}^t}$ represents the content delivery latency when a requested

TABLE 2: Notations

Symbol	Description
$\mathcal{T}$	Set of time slots, indexed by t
$\mathcal{N}$	Set of UEs, indexed by n
$\mathcal{G}$	Set of gNBs, indexed by g
$\mathcal{E}$	Set of co-located pairs of PSA UPFs and EASs, index by E_i
d_{ng}^t	Wireless transmission speed between UE n and its corresponding gNB g at time slot t
d_{gd}	Transmission speed in Fronthaul
d_{gc}	Transmission speed in Midhaul
d_{gi}	Transmission speed between gNB g and E_i
l_n^t	Location of UE n at time slot t
c_n^t	Content size requested by UE n at time slot t
s_{ij}^t	Handover latency from E_i to E_j at time slot t
x_{ng}^t	Decision variable indicating if gNB g is allocated to UE n at time slot t
z_{ni}^t	Decision variable indicating if targeted content for UE n is located at E_i at time slot t
y_{nij}^t	Decision variable indicating if content for UE n is migrated from E_i to E_j at time slot t
k_{ngi}^t	Decision variable indicating if gNB g and E_i are allocated to UE n at time slot t
$\mathcal{R}_n^t$	Set of routing paths available for UE n at time t
$\mathcal{H}_n$	History of selected routing paths for UE n
q_n^t	Routing path chosen for exploration
p_n^t	Routing path for exploitation (optimal path among explored ones)
β	Weight of exploration parameter
$\mathcal{D}_{n,p_n^t,t}$	Observed latency for UE n on routing path p_n^t at time t
$\bar{p}_n^t$	Selected routing path for UE n at time slot t
$\chi_{n,p_n^t,t}$	Count of times routing path p_n^t has been selected for UE n until time slot t
$R_{n,T}$	Learning Regret

content is delivered from gNB g to UE n. Constraints (2) ensure each UE connects to only one gNB at any time slot for accessing edge content. Constraints (3) ensure that at each time there is an EAS at which the UE's targeted content is located. Constraints (4) guarantee that the handover between EASs happens at most once between two consecutive time slots for each UE. Constraints (5) are to set the handover decision variables according to whether a content migration/handover happens or not. Constraints (6) set the routing path decision variables based on whether a routing path composed of gNB and EAS is selected or not. Constraints (7)-(8) ensure that routing decision variables are not set to one if the corresponding gNB and EAS are not selected. Finally, constraints (9)-(12) ensure that all decision variables are binary. Table 2 presents a summary of symbols and notations in the paper.

3.2 Learning (Partially Known Information)

In a dynamic 5G-MEC environment, several key pieces of information are typically unknown or partially known, significantly influencing decision-making processes. This uncertainty primarily pertains to dynamic network conditions (e.g., transmission speed or handover latency), user mobility patterns (e.g., future location of a user), and real-time content demands, which are crucial for optimizing content delivery routes.

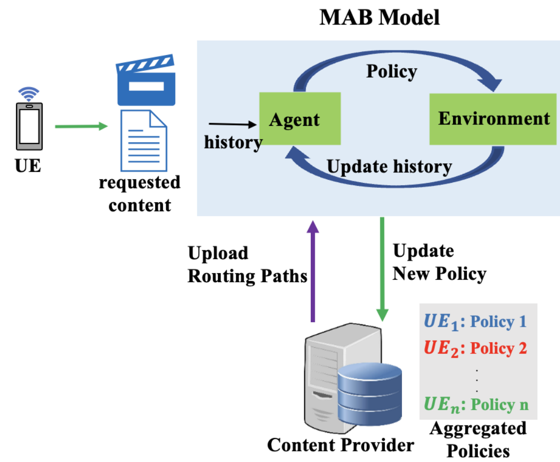


Fig. 2: MAB Model

Online sequential decision-making is a promising technique for handling dynamic and uncertain information as data arrives sequentially. This approach is ideal when complete information is not initially available for making decisions, allowing for adaptive responses to constant changes in data. Unlike traditional batch learning methods which require extensive memory or data storage, online sequential decision-making processes small data portions at a time, offering time and cost efficiency, making it well-suited for real-time MEC requirements. A central challenge in this context is the exploration-exploitation trade-off, which can be effectively addressed through methods such as Multi-Armed Bandits (MAB). MAB is particularly useful when faced with scenarios involving multiple available actions or decisions with incomplete information about the rewards associated with each action. However, it is important to note that this approach has limitations in handling large-scale datasets.

The advent of neural networks has made it possible to analyze any data. Neural networks are effective in solving complex problems and can discover and model nonlinear and complex relationships. They are suitable for environments where not all information and patterns are known beforehand. Neural networks can uncover hidden patterns, make predictions, and generalize to unseen data. As they gather more information, the performance of neural networks improves, while traditional machine learning algorithms eventually reach a point where more data does not improve performance. After learning from the initial inputs and their relationships, neural networks can also infer unseen relationships on unseen data, thus making the model generalize and predict unseen data. A Deep Neural Network (DNN) is a neural network with multiple hidden layers and nodes in each layer, which can be used to train and predict outputs from complex data.

In this regard, we design two QoS-Aware 5G-MEC Component Selection (QCS) approaches based on the above

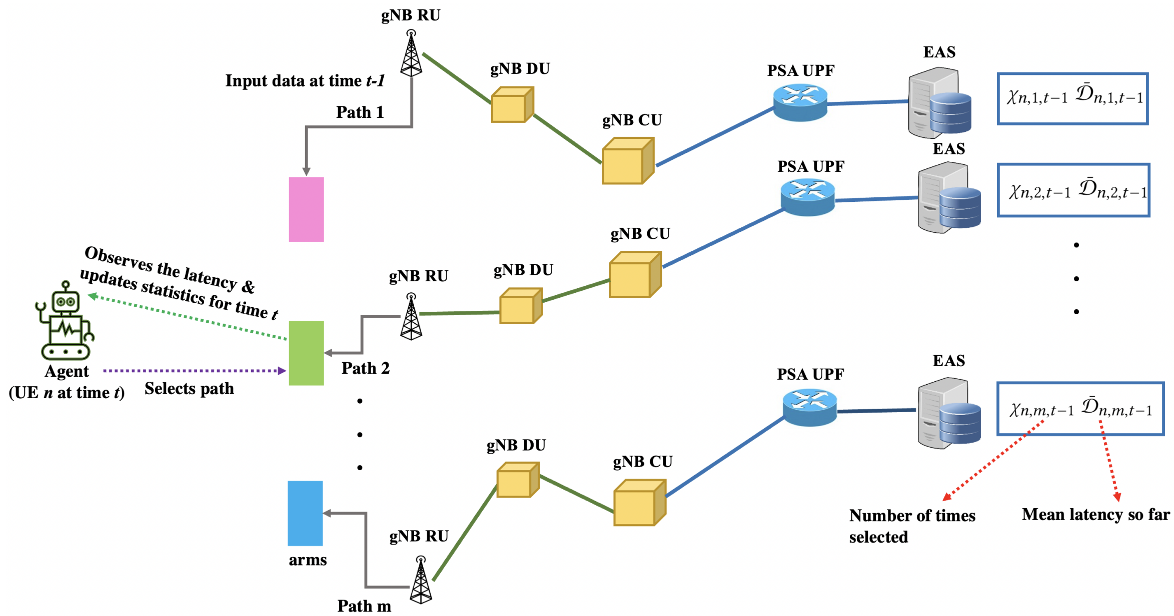


Fig. 3: Component selection using multi-armed bandit learning

learning approaches for content delivery to efficiently solve the component selection problem, both in small and large cases, when there is limited information. We will describe these approaches in the next two sections.

4 QoS-aware 5G-MEC Component Selection Based on Multi-Armed Bandit

Multi-Armed Bandit (MAB) [35] is a mathematical framework for modeling decision-making problems where an agent interacts with an environment to maximize the total reward over time. It is inspired by a whimsical scenario of a gambler who must choose which slot machine to play to maximize their winnings. It refers to determining which arm of a K-slot machine to pull to maximize the total reward in a series of attempts. In MAB, the agent chooses an action from a finite set of actions and collects a non-deterministic reward depending on the action taken. The goal of the agent is to maximize the total collected reward over time. The agent must make decisions based on incomplete information, resulting in a dilemma known as Exploration vs. Exploitation. Exploration refers to collecting new information leading to better decisions in the future; while Exploitation refers to choosing the best option given current information. MAB is ideal for modeling many real-world learning and optimization problems with uncertain information about the actions and their rewards.

We propose a QoS-aware 5G-MEC component selection approach based on the MAB algorithm for content delivery. This approach, called QCS-MAB, works by considering each potential routing path as an arm. It then selects the path (arm) with the highest expected reward, which is based on the observed QoS. QCS-MAB uses Bayesian techniques to dynamically update its beliefs about the expected reward for each path and make decisions in real-time. QCS-MAB is designed as an online sequential decision-making solution for each UE to determine the optimal routing path by selecting gNB and EAS that minimizes latency for the content delivery over time. The obtained routing path, denoted by $\mathcal{Z}$ is selected in real-time and independently from other UEs.

QoS-MAB balances the tradeoff between exploration and exploitation by considering potential new routing paths to discover a range of possible new latencies while considering previously known optimal routing paths. QoS-MAB continuously updates its estimates of the value of each routing path over time. The core idea is that QCS-MAB chooses an unselected routing path for each UE for every time slot and observes its average latency per unit of received content. During time slots with no new unselected routing path for exploration, the UE selects a routing path that has the highest preference over others so far. QCS-MAB also reduces the complexity of learning by only considering routes that can deliver the UE's requested content and are in closer proximity. This approach is expected to provide near real-time solutions for highly mobile users and is lightweight, making it suitable for real-time decisioning in MEC.

QCS-MAB is primarily based on the UCB1 (Upper Confidence Bound) strategy [35], which sets a balance between exploration and exploitation in the selection of routing paths. UCB1 allocates a counter for each routing path to specify the number of times that route has been selected. It decides the priority of every route based on its obtained mean latency and the number of times that route has been selected, taking into account both exploitation and exploration. Figures 2-3 provide further details on our proposed QCS-MAB approach.

4.1 QCS-MAB Approach

The pseudo-code of our proposed approach is presented in Algorithm 1. For every UE n, the information about the current location l_n^t, the routing paths $\mathcal{R}_n^t$ in close proximity of the UE, the amount of received content c_n^t, and the weight of exploration parameter β are available at each time slot (Line 1). The history of the selected routing paths for every UE n is maintained and denoted by $\mathcal{H}_n$, which is initially set to an empty set (Line 2).

QCS-MAB first checks if there is an unselected routing path $\in \mathcal{R}_n^t$ (Line 4). If such a route exists, QCS-MAB selects it once as the path chosen, denoted by q_n^t, for exploration for UE n at time slot t(Line 5) and appends it to the history set (Line 6). QCS-MAB then observes the perceived latency by UE n on this new routing path q_n^t at time slot t (Line 7). The sample mean latency per unit of the received content, $\bar{\mathcal{D}}_{n,q_n^t,t}$, for routing path q_n^t is then updated (Line 8). This allows for comparison of the average experienced latency per unit of content for each routing path. The parameter $\chi_{n,q_n^t,t}$ represents the number of times that the newly selected routing path q_n^t has been selected for UE n until time slot t. It is initially is set to 1 for each newly selected routing path and used in the calculation of the average latency per unit of content by that routing path (Line 9).

If there is no unselected routing path, QCS-MAB selects the optimal routing path p_n^t among the already explored routing paths, representing the up to this point optimal path (Line 11). The parameter β balances exploitation and exploration and assigns a value to the less selected routing paths. The first term of the formula gives higher value to the already best routing paths with the lowest average latency, which represents exploitation. As the routing path, p, is selected less often over time, the second term becomes more significant, leading to exploration of less observed routing paths. The value of β adjusts the weight between the first term (exploitation) and the second term (exploitation). When a new routing path is selected, QCS-MAB observes the new latency $\mathcal{D}_{n,p_n^t,t}$ (Line 12) and updates the mean latency per unit of content $\bar{\mathcal{D}}_{n,p_n^t,t}$ by calculating the average of past and newly observed latencies, then increments the count of times the path has been selected $\chi_{n,p_n^t,t}$ (Lines 13-14). This adaptive iterative process enables QCS-MAB to continuously refine the selection of routing paths for each UE at every time slot. The algorithm outputs the optimal routing path $\bar{p}_n^t$ for each UE at each time slot, based on the accumulated data and learning. QCS-MAB progressively improves the selected routing paths as more knowledge about different routing paths and network conditions is gained over time, leading to improved QoS.

4.2 Regret Analysis

We assess the performance loss of each UE during the learning process of QCS-MAB by measuring the *learning regret* to quantify the efficiency and effectiveness of the obtained routing paths selected over time. The learning regret is a combination of sampling regret and handover regret, as described in [36]. The sampling regret refers to the expected loss in reward due to the lack of perfect knowledge about the optimal strategy. It is calculated as the expected difference between the total rewards if an optimal routing path was used and the actual total rewards that was realized using selected routing path. Therefore, it captures the regret incurred due to not knowing the best routing path a priori. On the other hand, the handover regret indicates the expected handover latency for a UE due to handovers between different routing paths. Handover regret is a critical component in dynamic environments like 5G-MEC, where UEs frequently switch between different routing paths.

Algorithm 1 QoS-Aware 5G-MEC Component Selection based on Multi-Armed Bandit (QCS-MAB)

1: **Input:** $l_n^t, \mathcal{R}_n^t, c_n^t, \beta$ at the beginning of each time slot for UE n, $\forall t \in T$
2: $\mathcal{H}_n = \emptyset$
3: **for all** $t \in T$ **do**
4:   **if** $\exists\, q \in \mathcal{R}_n^t$ such that $q \notin \mathcal{H}_n$ **then**
5:     Select $q_n^t = q$
6:     $\mathcal{H}_n = \mathcal{H}_n \cup q_n^t$
7:     Observe latency $\mathcal{D}_{n,q_n^t,t}$ on path q_n^t
8:     Update mean latency $\bar{\mathcal{D}}_{n,q_n^t,t} = \frac{\mathcal{D}_{n,q_n^t,t}}{c_n^t}$
9:     $\chi_{n,q_n^t,t} = 1$
10:   **else**
11:     Select $p_n^t = \arg\min_{p \in \mathcal{H}_n} \left(\bar{\mathcal{D}}_{n,p,t} - \beta \sqrt{\frac{2 \ln t}{\chi_{n,p,t}}}\right)$
12:     Observe new latency $\mathcal{D}_{n,p_n^t,t}$ on path p_n^t
13:     Update $\bar{\mathcal{D}}_{n,p_n^t,t} = \frac{\chi_{n,p_n^t,t} \times \bar{\mathcal{D}}_{n,p_n^t,t} + \frac{\mathcal{D}_{n,p_n^t,t}}{c_n^t}}{\chi_{n,p_n^t,t} + 1}$
14:     $\chi_{n,p_n^t,t} = \chi_{n,p_n^t,t} + 1$
15:   **end if**
16:   Update and store the optimal path $\bar{p}_n^t$ for UE n at t
17: **end for**

The learning regret for UE n until time T is represented by $R_{n,T}$ and is defined as follows:

$$R_{n,T} = \underbrace{\mathbb{E}[\sum_{t=1}^{T} \mathcal{D}_{n,\bar{p}_n^t,t} - \mathcal{D}_{n,p_n^*,t}]}_{\text{sampling regret}} + \underbrace{\mathbb{E}[h_{n,p_n^t}]}_{\substack{\text{handover} \\ \text{regret}}}$$

where $\mathcal{D}_{n,\bar{p}_n^t,t}$ and $\mathcal{D}_{n,p_n^*,t}$ represent the latencies experienced by UE n using the chosen routing path $\bar{p}_n^t$ and the optimal routing path p_n^*, respectively. Furthermore, $\mathbb{E}[h_{n,p_n^t}]$ indicates the expected handover regret for UE n when using the path p_n^t. In particular, h_{n,p_n^t} captures the latency when handover happens. This is the amount of regret QCS-MAB experiences for not knowing the best arm (path) in advance. This regret analysis provides a comprehensive measure of the algorithm's performance over time, taking into account both the quality of routing decisions and the impact of handovers.

5 QoS-aware 5G-MEC Component Selection based on Deep Neural Network

Deep Neural Networks (DNNs) have emerged as a powerful approach in Machine Learning (ML) with an ability to uncover complex patterns and deliver precise predictions from data across diverse domains such as computer vision, speech recognition, and big data analysis. Deep learning (DL) algorithms, including DNNs, have capability to efficiently process large and complex datasets, making them particularly suitable for dynamic and multifaceted problems like 5G-MEC component selection. In contrast to traditional optimization algorithms that are typically designed to methodically search for the optimal solution, DNNs are adept at quickly finding near-optimal solutions in real-time. This distinction is critical in large-scale or complex scenarios

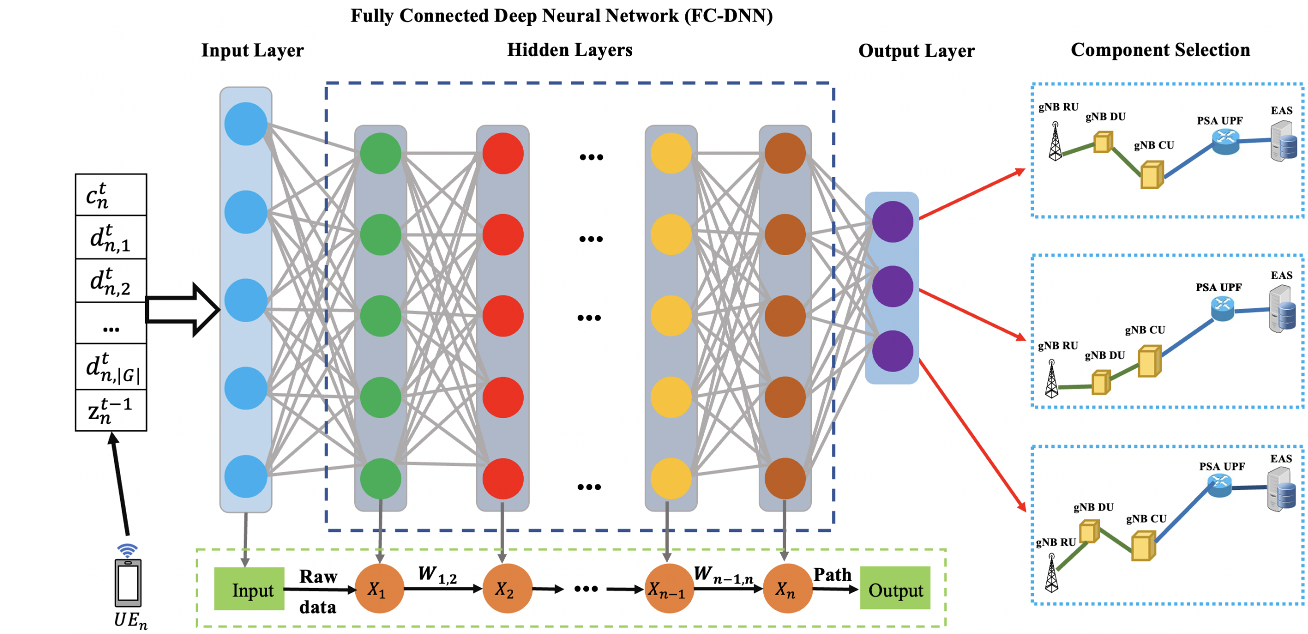


Fig. 4: A general framework for 5G-MEC component selection using fully connected DNN

where exhaustive searches for the absolute optimal solution might be computationally prohibitive or time-consuming. While MAB methods, such as QCS-MAB, are valuable for their ability to adaptively learn optimal strategies through sequential decision-making, they have their limitations. In the initial stages of the learning process, MAB approaches can lead to the selection of suboptimal routing paths, resulting in higher handover frequencies and content migrations between EASs. MAB approaches also require a well-defined reward system that often involve more complex implementation and training procedures. Given these considerations, DNNs present a valuable alternative for 5G-MEC component selection.

According to the universal approximation theorem of DNNs [37], a DNN with at least one hidden layer can approximate any continuous function to a high degree of accuracy, given sufficient data and computational resources. This theoretical foundation assures that DNNs can effectively be suitable for the 5G-MEC component selection, where the underlying system dynamics are too complex for traditional modeling approaches. Moreover, choosing DNN over other ML approaches such as Graph Neural Networks (GNNs) [38] or Support Vector Machines (SVMs) [39] in component selection is justified by its computational efficiency, simplified model complexity, ease of implementation, focus on node-level features, and potential benefits with larger labeled datasets.

To solve the 5G-MEC component selection using a DNN (QCS-DNN), we model the network as a fully connected neural network. The 5G-MEC architecture consists of three layers (UEs, gNBs, and EASs), with fixed structures for gNBs and EASs, while the location of UEs changes over time.

5.1 Training and Evaluation of the QCS-DNN Model

QCS-DNN begins with data preprocessing, involving the import of a dataset containing the selected features relevant to the 5G-MEC component selection problem. To ensure stability and convergence during training, we normalize the input data using the StandardScaler, a crucial step that standardizes feature values and prevents any particular feature from disproportionately influencing the learning process due to variance in scale.

The dataset is then divided into training and testing sets, with 80% of the data allocated for training and the remaining 20% for testing, allowing the model to be evaluated on unseen data. The target variable undergoes one-hot encoding to make it suitable for multi-class classification. Through numerical analysis, we establish a neural network architecture for QCS-DNN consisting of four layers: three hidden layers and one output layer. We choose the ReLU activation function for the hidden layers due to its efficiency in converging to optimal results and the Sigmoid activation function for the output layer, which is particularly suitable for multi-class classification tasks. Following the architectural design, we compile the model, specifying the categorical cross-entropy loss function and the Adam optimizer. This compilation step readies the model for training by defining the necessary parameters for gradient descent and error computation. The training process is conducted over 200 epochs, with a batch size of 8000, carefully chosen through experimentation to balance model performance and computational efficiency. The optimal 5G-MEC component selection policies obtained from running the IP (per different UEs) are used as labeled samples to train the DNN. These policies, serving as ground truth labels, guide the model during training. Once the training of the DNN is completed, we evaluate its performance on the test samples. This evaluation assesses how well the model generalizes to unseen 20% samples we considered separately for the testing phase.

5.2 QCS-DNN Model Architecture

The QCS-DNN model's architecture is an integral part of its functionality, designed specifically for the 5G-MEC component selection problem. The QCS-DNN framework is illustrated in Fig. 4, comprises several layers, each contributing to processing the input data and extracting meaningful patterns. The inputs to the QCS-DNN include the content size of each UE $n \in N$, the wireless transmission speed between each UE $n \in N$ and all the corresponding gNB $g \in G$ at different time slots, and the assigned routing path from the previous time slot. We do not feed explicit information about the UEs locations and 5G-MEC to QCS-DNN as the impact of the UE locations is inherently reflected by the wireless transmission speed and while other parameters of the 5G-MEC are fixed. In addition, the hidden layers of the model, denoted as X_1, X_2, ..., X_n, consists of neurons that process these inputs. The weights $W_{u,v}$ and $W_{v,w}$ for neuron X_v refer to the input and output data size (e.g., weights of the DNN model) of the neuron, respectively. This determines the flow and transformation of data through the network. After the input data is processed in these layers, the extracted features generated by the last layer will be processed by a classifier and recognized as the output, denoting the path formed from the selected 5G-MEC components.

QCS-DNN model aims to offer a reliable and efficient solution for 5G-MEC component selection, tailored to adapt to the dynamic and complex nature of the environment.

6 Experimental Results

This section compares the performance of our proposed approaches, QCS-MAB and QCS-DNN, with other approaches.

6.1 Experimental Setup

We consider the following benchmarks:

- IP: We use IBM ILOG Concert Technology API for C++ [40] to implement our IP model, presented in equations (1-12). The IP model serves as a benchmark, providing the optimal solution,
- Epsilon Greedy (ϵ-Greedy): This is an MAB approach that balances exploration and exploitation based on the ϵ probability,
- Nearest Neighbor (NN): This is a heuristic approach that leverages the geographical proximity of the content delivery routing path and chooses the physically closest path, and
- Round Robin (RR): This is a heuristic approach that cycles through available content delivery routing paths in a predetermined order, ensuring that each path gets an equal share of content delivery requests.

All approaches are implemented in C++, and the experiments are conducted on a desktop PC with 2.80 GHz, 11th Gen Intel(R) Core(TM) i7-1165G7 and 16 GB RAM.

We use different datasets to represent various components of a 5G-MEC system. To determine the coordinates of UEs at each time slot, we use three datasets with different mobility patterns, including subway trains, buses, walking, rode trolleys, cars, and trucks, to evaluate the performance of our proposed approaches. These datasets are: 1) RioBuses [41], a dataset of mobility traces of buses in Rio de Janeiro, Brazil; 2) Oviedo/asturies-er [42], which contains mobility and connectivity traces extracted from

TABLE 3: Experiment Scenarios

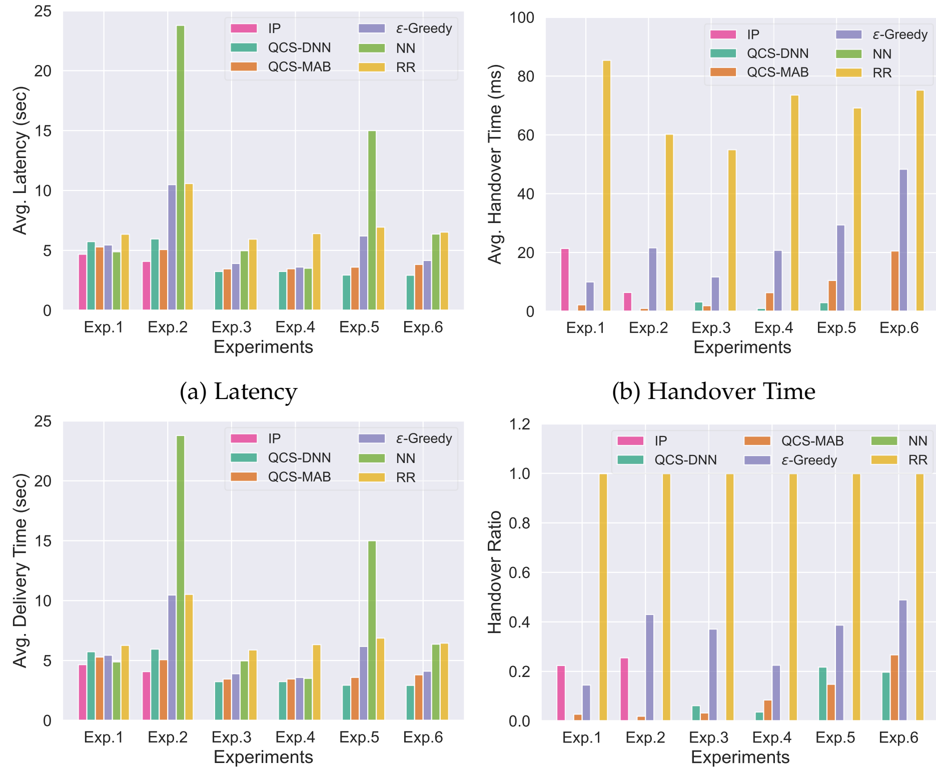
Exp.	# UEs	# gNB	# PSA UPF	# EAS
1	12	2	2	2
2	25	2	2	2
3	50	6	2	2
4	100	8	3	3
5	150	8	4	4
6	200	10	5	5

GPS traces collected from the regional Fire Department of Asturias, Spain; 3) mobilitymodels [43] a collection of people mobility traces from five different sites - two university campuses (NCSU and KAIST), New York City, Disney World (Orlando), and North Carolina state fair. Each dataset comprises a third of all coordinates of the applications, and the datasets are mapped to the same geographic area for consistency (coordinates of Oviedo/asturies-er and mobilitymodels datasets are mapped to the same geographic area as RioBuses dataset). Some coordinates were in Geodetic Coordinate system and were converted to the Cartesian for consistency in the dataset. The contents requested by the UEs at each time slot are obtained from [44] and scaled ranging from 3-5GB. The time slot duration is set to 60 seconds.

ThousandEyes [45], a network intelligence company acquired by Cisco, utilizes a dynamic monitoring technique to gather important network metrics such as loss, latency, jitter, and detailed path metrics with layer-3 hops. To model the transmission speeds between links in a 5G network, we use the latency between hops obtained from ThousandEyes. The 5G links are divided into three parts: Fronthaul, Midhaul, and Backhaul. Fronthaul refers to the connectivity between gNB RUs and gNB DUs; Midhaul to the communication link between gNB DUs and gNB CUs; and Backhaul to the connection between gNB CUs (with PSA UPFs) and EASs. We place three hops between the source and target nodes to model the transmission speeds in these links, resulting in a path with five nodes from source to target. These nodes are characterized as the gNB RU (source node), gNB DU (second node), gNB CU (third node), PSA UPF (fourth node), and EAS (fifth node), and together they form the 5G-MEC framework depicted in Fig. 1.

Based on the latency values obtained from ThousandEyes and data transmission size, we calculate the corresponding data transmission speeds in the links and scale them up to characterize the speeds in 5G-MEC. Then, we determine the minimum and maximum speeds in each hop using these samples and use a uniform distribution to simulate additional data for the experiments. Similarly, we use the method described in [46] to model the wireless transmission speed of the UEs connecting to the gNB RUs at each time slot. Then, these values are scaled to be appropriate for 5G-MEC.

As shown in Table 3, six scenarios are designed for the experiments. The parameter β in QCS-MAB and ϵ in ϵ-Greedy are set to 10^{-4} and 0.3, respectively. We conduct a sensitivity analysis on the value of β ranging between 10^{-5} to 10^{-2} to learn the best β value that results in minimizing latency (see Fig. 7a).



(a) Latency

(b) Handover Time

(c) Content Delivery Time

(d) Handover Ratio

Fig. 5: Performance Analysis (* no bars for IP in Exp. 3-6 since IP was unable to determine any solution in feasible time)

6.2 Comparative Analysis

We use different metrics such as experienced latency, handover time, content delivery time, handover ratio, and regret analysis to compare QCS-DNN and QCS-MAB with IP, ϵ-Greedy, NN, and RR algorithms. The time horizon T is set to 1000 for Figs. 5a-5d. Due to the NP-hardness of the problem, obtaining optimal results through IP is intractable. In the experiments, the maximum feasible time for the solver to find a solution is set to 120 minutes. However, the solver was unable to obtain optimal solutions within this time frame for most scenarios.

Comparative Analysis on Latency. Fig. 5a shows the average latency per time slot for UEs, measured in seconds. The results indicate that IP could not find a solution for Exp. 3-6 due to the NP-hardness of the problem. QCS-MAB performs close to optimal results in Exp. 1-2 and has consistently lower latency time in all experiments compared to ϵ-Greedy. ϵ-Greedy uses a simple random technique to balance exploration and exploitation. However, QCS-MAB intentionally forms this tradeoff by considering the repetition of the least observed routing paths and utilizes a more in-depth formula to minimize latency. QCS-DNN outperforms QCS-MAB in larger-scale experiments as it has more information about the best routing paths for UEs under different circumstances, enabling it to make better decisions considering the dynamic and changing behavior of UEs and 5G-MEC. In general, our approaches outperform the NN and RR algorithms significantly. These algorithms rely on static content delivery decisions and lack adaptability to changing conditions, resulting in suboptimal performance in dynamic environments.

Comparative Analysis on Handover Time. Fig. 5b shows the average handover time of UEs per time slot in milliseconds. QCS-MAB demonstrates comparable results with QCS-DNN in Exp. 1-3 and significantly reduced handover time compared to ϵ-Greedy. This improvement is because of the selection of the best β value for QCS-MAB, which effectively balances exploration and exploitation, leading to a reduction in both handover time and latency. Likewise, QCS-DNN outperforms ϵ-Greedy as it can make better decisions by gathering extensive data on the most suitable routing paths. ϵ-Greedy results in a higher handover time due to its random nature caused by the value of ϵ. However, QCS-DNN outperforms QCS-MAB significantly in larger-scale experiments by making more informed decisions, thus reducing the need for frequent handover. IP does not achieve the best handover time in Exp. 1-2, as the minimization of handover is not its direct objective (but the total latency is). The NN algorithm consistently selects the closest route for content delivery, effectively eliminating handovers. In contrast, the RR algorithm leads to a large number of handovers due to its cyclic allocation of requests to different content delivery routes.

Comparative Analysis on Content Delivery Time. Fig. 5c shows the results of the average content delivery time of UEs per time slot. UEs experience better content delivery time with QCS-MAB compared to ϵ-Greedy in all experiments. In Exp. 1-2, QCS-MAB shows close to optimal results. This is due to the fact that QCS-MAB effectively selects the best routing paths or explores less selected ones. QCS-DNN

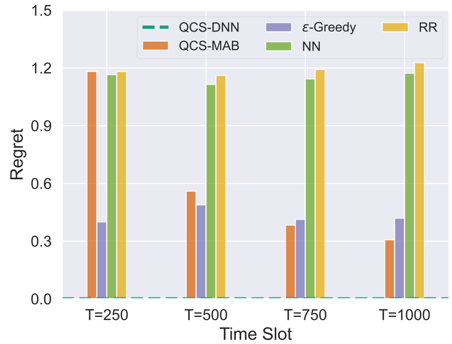


Fig. 6: Regret Analysis

achieves the best results in larger experiments (Exp. 3-6) as it can better determine the best routing paths for UEs based on the additional information obtained. QCS-DNN and QCS-MAB significantly outperform NN and RR algorithms because they better adapt to changing conditions, enabling them to make more efficient content delivery decisions.

Comparative Analysis on Handover Ratio. We measure the handover ratio as the percentage of handover that occurs for each UE per time slot. Fig.5d shows the handover ratio for each approach. Although IP can only solve Exp. 1-2 within a feasible timeframe, it does not consistently achieve the minimum handover ratio. QCS-DNN and QCS-MAB show comparable handover ratios, both approaches tend to experience higher ratios in larger experiments due to the increased availability of routing path options, which are leveraged to minimize latency. In comparison, ϵ-Greedy consistently results in significantly higher handover ratios across all experiments. NN has a handover ratio close to zero because it consistently selects the closest content delivery route, eliminating the need for frequent handovers. In contrast, RR has the worst handover ratio, close to one, due to cyclically distributing the UE requests to different routes, causing frequent connection switches and high handover rates.

Comparative Analysis on Regret. The learning regret, which represents the deviation percentage from the optimal result, is compared in Fig. 6. The results from Exp. 6 were used to assess the regret of QCS-DNN and QCS-MAB approaches. QCS-DNN results in minimum regret, close to 0, in all time slots (close to the actual values of the labels), significantly outperforming other approaches. QCS-MAB shows a decreasing regret over time as there are more samples available for learning. This indicates QCS-MAB's ability to gradually approximate optimal paths, despite initial deviations and a higher frequency of handovers. However, ϵ-Greedy shows a random and not-linear pattern. NN and RR algorithms exhibit the highest regret and lack any reduction in regret over time, primarily due to lack of adaptability and learning capabilities.

6.3 Sensitivity Analysis

We conduct sensitivity analysis to evaluate the impact of two parameters, β value and number of time slots T, on the results while keeping the other parameters fixed to investigate the sensitivity over each distinct parameter.

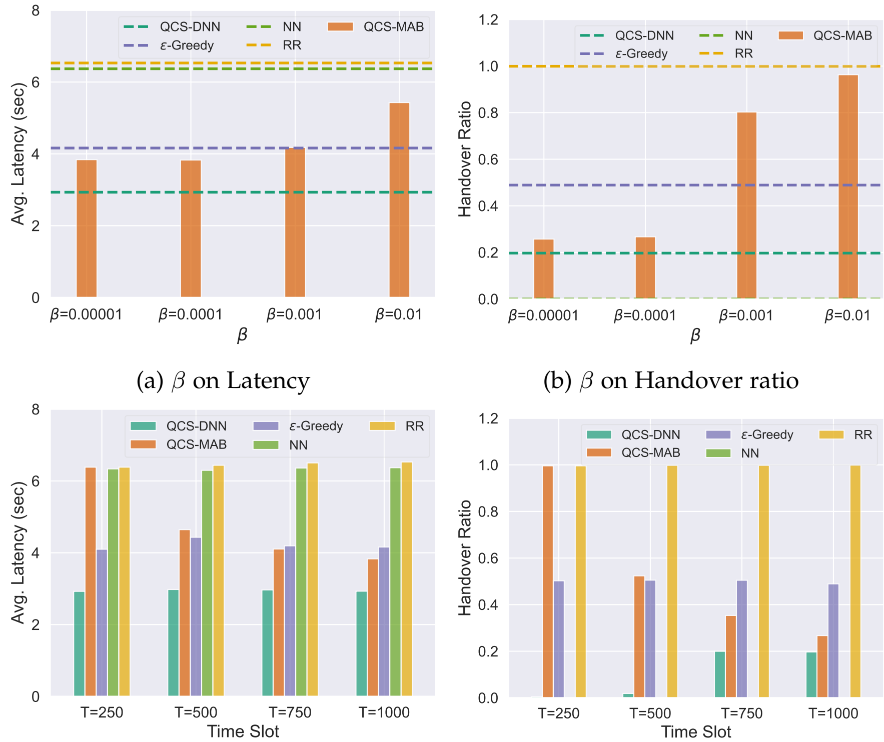
Sensitivity Analysis on Beta. In QCS-MAB, β determines the weight of exploration versus exploitation. For Exp. 6 with $T = 1000$, we perform two sensitivity analyses on the β value to investigate its impact on latency and handover ratio. The results are presented in Figs. 7a and 7b, respectively. It should be noted that the other approaches are not sensitive to the value of β, we only present their results. The results show that the best β for the latency is the one that strikes the best balance between exploration and exploitation, which in our case, is 0.0001. The results also indicate that as β increases, the handover ratio also increases. This is because a higher value of β causes a higher chance for exploration, which in turn leads to more handovers.

Sensitivity Analysis on Time Horizon. We carry out a sensitivity analysis on the number of time slots (T) for Exp. 6 to evaluate its impact on latency and handover ratio. The results are presented in Figs. 7c and 7d, respectively. As the number of iterations (time slots) increases, QCS-MAB is able to make better decisions and learn which routing paths result in optimal latency and lower handover ratio. QCS-DNN maintains consistent latency across different time slots due to sufficient initial information in the large-scale experiment (Exp. 6), although its handover ratio increases slightly. As expected, ϵ-Greedy does not exhibit meaningful sensitivity when the number of time slots increases. Both the NN and RR algorithms show insensitivity to the number of time slots and consistently result in higher latency across different numbers of time slots. NN consistently achieves a handover ratio of 0, while RR consistently results in a handover ratio of close to 1 for various time slots.

To sum up, the results demonstrate the effectiveness of our proposed approaches, QCS-MAB and QCS-DNN, in finding the optimal routing paths along with the corresponding 5G-MEC components for UEs, ensuring efficient high-speed content delivery with low latency, minimal handover time, and optimal content delivery times. While QCS-MAB performs well in smaller experiments, QCS-DNN exhibits better results for larger experiments.

7 Conclusion and Future Work

This paper tackled the critical challenge of component selection in 5G-MEC, which plays a pivotal role in alleviating backhaul congestion and enhancing QoS for UEs. We proposed two learning-based approaches to learn the optimal components that result in minimum latency for UEs. First, we proposed QCS-MAB, a multi-armed bandit-based approach that learns the optimal routing paths for UEs over time based on a tradeoff between exploration and exploitation. Then, we proposed QCS-DNN, a fully-connected neural network that makes the optimal routing path decisions based on historical data analysis. The experimental results indicate that our approaches outperform traditional methods in terms of latency reduction and handover ratio, showcasing their effectiveness in a 5G-MEC context. Future research directions include investigating methods to encourage cooperation among UEs to facilitate content sharing among adjacent UEs, ultimately reducing



(a) β on Latency

(b) β on Handover ratio

(c) T on Latency

(d) T on Handover ratio

Fig. 7: Sensitivity Analysis

routing path length and enhancing QoS. We intend to explore GNNs to model complex interconnections within the network infrastructure (virtual networks and network slicing), potentially enhancing the efficiency of resource allocation, network traffic management, and personalized QoS in dynamic 5G-MEC environments.

ACKNOWLEDGMENTS

This paper is a revised and extended version of [47]. We gratefully acknowledge the support of Cisco grant CG#1935382 and NSF grant CNS-2145268.

REFERENCES

[1] Cisco, "Internet of things." Available: https://www.cisco.com/c/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf, 2016.

[2] Ericsson, "Ericsson mobility report: Data and forecasts," 2023. Accessed: 2023-11-10.

[3] "Latency is everywhere and it costs you sales - how to crush it." Available: http://highscalability.com/latency-everywhere-and-it-costs-you-sales-how-crush-it, 2009.

[4] X. Huang, L. He, L. Wang, and F. Li, "Towards 5G: Joint optimization of video segment caching, transcoding and resource allocation for adaptive video streaming in a multi-access edge computing network," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10909–10924, 2021.

[5] D. Bhatta and L. Mashayekhy, "A bifactor approximation algorithm for cloudlet placement in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1787–1798, 2022.

[6] W. Ma and L. Mashayekhy, "Quality-aware video offloading in mobile edge computing: A data-driven two-stage stochastic optimization," in *Proc. of the 14th IEEE International Conference on Cloud Computing*, pp. 594–599, 2021.

[7] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal, *et al.*, "Mobile-edge computing introductory technical white paper," *White paper, mobile-edge computing industry initiative*, vol. 29, pp. 854–864, 2014.

[8] "5G Infrastructure Public Private Partnership." Available: https://5g-ppp.eu/. Accessed: 2022-08-29.

[9] T. Yoshizawa, S. B. M. Baskaran, and A. Kunz, "Overview of 5G URLLC system and security aspects in 3GPP," in *Proc. of the IEEE Conference on Standards for Comms. and Networking*, pp. 1–5, 2019.

[10] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Comms. Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[11] E. F. Maleki, L. Mashayekhy, and S. M. Nabavinejad, "Mobility-aware computation offloading in edge computing using machine learning," *IEEE Transactions on Mobile Computing*, 2021.

[12] R. Solozabal, A. Sanchoyerto, E. Atxutegi, B. Blanco, J. O. Fajardo, and F. Liberal, "Exploitation of mobile edge computing in 5G distributed mission-critical push-to-talk service deployment," *IEEE Access*, vol. 6, pp. 37665–37675, 2018.

[13] S. Guo, S. Shao, Y. Wang, and H. Yang, "Cross stratum resources protection in fog-computing-based radio over fiber networks for 5G services," *Optical Fiber Technology*, vol. 37, pp. 61–68, 2017.

[14] N. Sharghivand, L. Mashayekhy, W. Ma, and S. Dustdar, "Time-constrained service handoff for mobile edge computing in 5G," *IEEE Transactions on Services Computing (in press)*, pp. 1–13, 2022.

[15] H. Rahimi, Y. Picaud, K. D. Singh, G. Madhusudan, S. Costanzo, and O. Boissier, "Design and simulation of a hybrid architecture for edge computing in 5G and beyond," *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1213–1224, 2021.

[16] A. Kiani and N. Ansari, "Edge computing aware NOMA for 5G networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1299–1306, 2018.

[17] L. Nadeem, Y. Amin, J. Loo, M. A. Azam, and K. K. Chai, "Efficient resource allocation using distributed edge computing in D2D based 5G-HCN with network slicing," *IEEE Access*, vol. 9, pp. 134148–134162, 2021.

[18] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Y. Kwok, "Mobile edge computing enabled 5G health monitoring for internet of medical things: A decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2020.

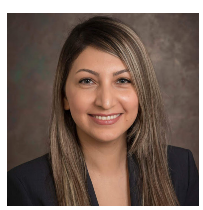
[19] P. K. Bishoyi and S. Misra, "Enabling green mobile-edge computing for 5G-based healthcare applications," *IEEE Transactions on Green Comms. and Networking*, vol. 5, no. 3, pp. 1623–1631, 2021.

[20] N. N. Ei, M. Alsenwi, Y. K. Tun, Z. Han, and C. S. Hong, "Energy-efficient resource allocation in multi-UAV-assisted two-stage edge computing for beyond 5G networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.
[21] L. Zhong, X. Chen, C. Xu, Y. Ma, M. Wang, Y. Zhao, and G.-M. Muntean, "A multi-user cost-efficient crowd-assisted VR content delivery solution in 5G-and-beyond heterogeneous networks," *IEEE Transactions on Mobile Computing*, 2022.
[22] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Cooperative content caching in 5G networks with mobile edge computing," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 80–87, 2018.
[23] S. Tang, A. Alnoman, A. Anpalagan, and I. Woungang, "A user-centric cooperative edge caching scheme for minimizing delay in 5G content delivery networks," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 8, p. e3461, 2018.
[24] E. K. Markakis, K. Karras, A. Sideris, G. Alexiou, and E. Pallis, "Computing, caching, and communication at the edge: The cornerstone for building a versatile 5G ecosystem," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 152–157, 2017.
[25] T. Hou, G. Feng, S. Qin, and W. Jiang, "Proactive content caching by exploiting transfer learning for mobile edge computing," *Intl. Journal of Communication Systems*, vol. 31, no. 11, p. e3706, 2018.
[26] S. Huang, H. Zhang, X. Wang, M. Chen, J. Li, and V. C. Leung, "Fine-grained spatio-temporal distribution prediction of mobile content delivery in 5G ultra-dense networks," *IEEE Transactions on Mobile Computing*, 2022.
[27] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *Proc. of the IEEE International Conference on Communications*, pp. 1897–1903, 2014.
[28] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, "When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2238–2251, 2020.
[29] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Collaborative edge computing and caching with deep reinforcement learning decision agents," *IEEE Access*, vol. 8, pp. 120604–120612, 2020.
[30] Y. Zhai, T. Bao, L. Zhu, M. Shen, X. Du, and M. Guizani, "Toward reinforcement-learning-based service deployment of 5G mobile edge computing with request-aware scheduling," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 84–91, 2020.
[31] H. Pang, J. Liu, X. Fan, and L. Sun, "Toward smart and cooperative edge caching for 5G networks: A deep learning based approach," in *Proc. of the IEEE/ACM 26th International Symposium on Quality of Service*, pp. 1–6, 2018.
[32] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, 2019.
[33] M. Chen, Y. Miao, H. Gharavi, L. Hu, and I. Humar, "Intelligent traffic adaptive resource allocation for edge computing-based 5G networks," *IEEE transactions on cognitive communications and networking*, vol. 6, no. 2, pp. 499–508, 2019.
[34] D. Qiao, S. Guo, D. Liu, S. Long, P. Zhou, and Z. Li, "Adaptive federated deep reinforcement learning for proactive content caching in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4767–4782, 2022.
[35] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.
[36] Y. Sun, S. Zhou, and J. Xu, "EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637–2646, 2017.
[37] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
[38] H. Wang, Y. Wu, G. Min, and W. Miao, "A graph neural network-based digital twin for network slicing management," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1367–1376, 2020.
[39] T. Saha, N. Aaraj, and N. K. Jha, "Machine learning assisted security analysis of 5G-network-connected systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 4, pp. 2006–2024, 2022.
[40] IBM, "Concert Technology version 12.1 C++ API Reference Manual." Available: https://public.dhe.ibm.com/software/websphere/ilog/docs/optimization/cplex/refcppcplex.pdf, 2009. Accessed: 2022-08-29.
[41] D. Dias and L. H. M. K. Costa, "CRAWDAD dataset coppe-ufrj/riobuses (v. 2018-03-19)." Available: https://crawdad.org/coppe-ufrj/RioBuses/20180319, Mar. 2018.
[42] S. Cabrero, R. Garcia, X. G. García, and D. Melendi, "CRAWDAD dataset oviedo/asturies-er (v. 2016-08-08)." Available: https://crawdad.org/oviedo/asturies-er/20160808, Aug. 2016.
[43] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "CRAWDAD dataset ncsu/mobilitymodels (v. 2009-07-23)." Available: https://crawdad.org/ncsu/mobilitymodels/20090723, July 2009.
[44] S. Farthofer, M. Herlich, C. Maier, S. Pochaba, J. Lackner, and P. Dorfinger, "CRAWDAD dataset srfg/lte-4g-highway-drive-tests-salzburg (v. 2022-01-18)." Available: https://crawdad.org/srfg/lte-4g-highway-drive-tests-salzburg/2022-01-18, Jan. 2022.
[45] "Thousandeyes." Available: https://www.thousandeyes.com/. Accessed: 2022-08-29.
[46] J. van der Hooft, S. Petrangeli, T. Wauters, R. Huysegems, P. R. Alface, T. Bostoen, and F. De Turck, "HTTP/2-Based Adaptive Streaming of HEVC Video Over 4G/LTE Networks," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2177–2180, 2016.
[47] E. F. Maleki, W. Ma, L. Mashayekhy, and H. La Roche, "Qos-aware 5G component selection for content delivery in multi-access edge computing," in *Proc. of the 14th IEEE/ACM International Conference on Utility and Cloud Computing*, pp. 1–10, 2021.

BIOGRAPHIES

Erfan Farhangi Maleki is currently a Ph.D. Candidate and a Doctoral Fellow in the Department of Computer and Information Sciences at the University of Delaware. His research interests include edge/cloud computing, 5G, and distributed systems.

Weibin Ma received his M.Sc. degree in Engineering Science from State University of New York at Buffalo, USA, in 2018. He is currently pursuing the Ph.D. degree in Computer Science at the University of Delaware, USA. His research interests include edge computing, game theory, and optimization.



Lena Mashayekhy is an associate professor in the Department of Computer and Information Sciences at the University of Delaware. Her research interests include edge computing, cloud computing, Internet of Things, and game theory. She is a recipient of the 2022 NSF CAREER Award and several best paper awards.

Humberto J. La Roche is Principal Engineer in Cisco's Mass-scale Infrastructure Group where he is responsible for 5G solution architecture. In his career in telecom, he has been responsible for LTE systems engineering, product management of IMS and wireless softswitch products, product planning of optical networking and broadband access products, and transport network design. Humberto holds a PhD degree in Physics from the University of Texas at Austin.