Edge Computing for Low Latency 5G Applications Using Q-Learning Algorithm

Aastha A Neeralgi, Anuj Baddi, Ishwari R Naik, Madivalesh Demakkanavar, Sandeep Kulkarni and Vijayalakshmi M

School of Computer Science and Engineering, KLE Technological University, Vidyanagar, Hubli, India

Keywords: Edge Computing, 5G Networks, Low-Latecy, Reinforcemnt Learning, Q-Learning.

Abstract: Next-generation technologies like industrial automa tion, augmented reality, and driverless cars depend on edge computing for low-latency 5G applications. In real-time appli cations, achieving ultra-low latency is essential to guarantee un interrupted communication, reduce delays, and improve user ex perience. Computational tasks are brought closer to end users by utilizing edge computing, which greatly cuts down on delays and enhances system responsiveness. The focus here is on lowering latency, avoiding needless handovers, and guaranteeing reliable connections by integrating Q-learning with edge computing to enhance handover decisions in 5G networks. The state space for the Q-learning algorithm is made simpler by incorporating crucial characteristics like latency and Signal-to-Noise Ratio (SNR) through preprocessing methods like normalization and discretization.Effective and flexible decision-making is ensured via a well-balanced exploration-exploitation approach and a well tuned reward system. In comparison to Random Forest model we trained, experimental results demonstrate an impressive 7.8% reduction in latency. This framework opens the door for developments in next-generation network technologies by offering a scalable, effective technique to handle major issues in latency sensitive 5G applications.

1 INTRODUCTION

5G technology represents a defining moment in the history of telecommunications, marked by high speeds of data transfer, minimized latency, and the ability to connect several devices simultaneously. It is more than just an enhancement of current mobile networks; rather, it signifies a paradigm shift in how data is processed, transmitted, and applied across a range of applications. As global interconnectivity increases, so does the demand for low-latency, high-bandwidth applications, such as those required for autonomous vehicles, Augmented Reality (AR), Virtual Reality (VR), smart cities, and real-time industrial automation (Wang, 2020; Hassan et al., 2019). However, traditional cloud computing architectures struggle to meet these demands due to the intrinsic latency issues associated with remote data centers.

To bridge this gap, Edge Computing (EC) has emerged as a transformative approach, bringing computational resources closer to the end-users. By distributing data processing and relocating it near the source of data, edge computing significantly reduces data transfer times, thereby mitigating latency. This proximity enhances the performance of latency-sensitive applications, optimizes bandwidth utilization, and alleviates the load on central cloud infrastructures (Abouaomar et al., 2021a; Hassan et al., 2019). When integrated with 5G networks, edge computing creates a synergistic ecosystem, enabling the development of next-generation applications that demand rapid data processing and instantaneous response times (Singh et al., 2016).

The framework of edge computing, however, is inherently heterogeneous, comprising diverse edge devices such as edge servers, routers, gateways, and numerous Internet Of Things (IoT) devices (Toka, 2021). These devices vary in their computational, storage, and communication capabilities, presenting both opportunities and challenges for effective resource management. To address these challenges, dynamic resource provisioning strategies, such as Lyapunov optimization, have shown promise in managing resources efficiently across edge devices. Such advancements ensure optimal performance within systems constrained by both latency and limited re-

Neeralgi, A. A., Baddi, A., Naik, I. R., Demakkanavar, M., Kulkarni, S. and M, V.
Edge Computing for Low Latency 5G Applications Using Q-Learning Algorithm.
DOI: 10.5220/0013643400004664

In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 3*, pages 733-740
ISBN: 978-989-758-763-4

sources (Hassan et al., 2019).

The integration of edge computing with 5G technology not only improves system performance but also drives innovation in service delivery and application development. For instance, in smart healthcare, edge computing enables immediate monitoring and data analysis, allowing for timely medical interventions. Similarly, in AR and VR applications, it enhances immersive experiences by reducing latency and facilitating seamless interactions (Hu et al., 2015; **?**). These innovations illustrate the potential of edge computing and 5G to transform industries and redefine user experiences (Djigal et al., 2022).

Building on this foundation, this study introduces a Q-learning framework designed to improve handover decisions in 5G networks (Yajnanarayana et al., 2020). By employing techniques such as normalization and discretization, the framework streamlines data processing to enhance learning efficiency and reduce latency (Shokrnezhad et al., 2024). These advancements directly address the latency challenges inherent in 5G and edge computing environments, further strengthening their combined potential (Kodavati and Ramarakula, 2023).

This introduction sets the foundation for understanding Q-learning and its application in reducing latency in 5G networks. Section 2 reviews the Literature Survey, Section 3 discusses the Methodology, and Section 4 explains the Implementation. Section 5 presents Results and Analysis, highlighting the framework's effectiveness. Section 6 concludes with a summary of contributions and suggests potential directions for future research.

2 LITERATURE SURVEY

Edge computing has emerged as an important technological innovation in the domain of 5G communications, that satisfies the growing need for real-time processing and data analysis. Because data processing is distributed and taken closer to the end-users, edge computing enhances the performance of applications that require fast responses (Liang et al., 2024). The concept of edge computing moved away from traditional cloud computing systems where data processing was centralized in distant data centers (Intharawijitr et al., 2017a). The preliminary studies pointed out the drawbacks of cloud computing, especially in terms of latency and bandwidth, which became worse with the rise of mobile and IoT applications. Shi et al. (2016) (Sumathi et al., 2022; Abouaomar et al., 2022) focused more on the necessity for a decentralized form of computing that led to the development of edge computing as an effective solution to these problems (Shokrnezhad et al., 2024).

This brings about the idea of edge computing where convergence with 5G network technologies can lead to various applications. An example from this domain is from autonomous vehicles. It utilizes edge computing in order to support immediate processing of vehicle sensor data (Intharawijitr et al., 2017a; Zhang et al., 2016). As such, it creates opportunities for making decisions at lightning speed in the environment of autonomous (Abouaomar et al., 2021b; Hu et al., 2015). Researchers have shown that it decreases latency, which is very necessary when fast decisions have to be made (Parvez et al., 2018). Edge computing is very supportive to manage large sensor and device networks in city-based smart applications. It can collect data and analyze it efficiently. Research demonstrates that edge computing can optimize traffic management systems, which can be processed locally, improve urban mobility and reduce congestion. Healthcare is another sector which benefits from edge computing in application such as telemedicine and remote patient monitoring. By processing data at the edge, healthcare providers can deliver real-time consultations and diagnostics, improving patient outcomes and operational efficiency (Hartmann et al., 2022).

With these benefits comes the problem of the adoption of edge computing in the 5G environment. A number of challenges will need to be addressed: Security and privacy: The decentralized architecture of edge computing increases fears of data protection and individual privacy (Wu et al., 2024). Research on security frameworks is needed in edge environments, protecting sensitive information being processed in such networks. Furthermore, the lack of agreed standards for edge computing technologies may obstruct interoperability and exacerbate issues with heterogeneous system integrations. Ongoing efforts require a collaborative process to promote development of commonly accepted communication protocols between edge devices as well as among networks involved in an application. Resource pooling on the edge can be more crucial in maintaining performance while preserving resources. Research indicates that sophisticated algorithms and machine learning methodologies have the potential to improve resource management by adapting in real-time to fluctuating workloads (Djigal et al., 2022; Liu et al., 2018).

Prospects for edge computing within the 5G communication framework appear positive as all the ongoing research is focused on making improvements in its functionalities. Artificial intelligence and machine learning technologies will heavily impact the

development of refining edge computing procedures for making applications more intelligent and responsive (Shokrnezhad et al., 2024; Singh et al., 2016). More, sustained development of 5G infrastructure will likely enhance the use of edge computing, thus giving avenues to new solutions in a host of sectors. In generalization, from literature sources already existing on edge computing in 5G communications, this field holds high transformative power across various fields. Although the challenges are present, technology and research advancements are likely to address them, thereby making ample use of edge computing solutions possible (Hassan et al., 2019). Increased demand from low-latency applications edge computing will hence contribute significantly to the evolving scene in communication technologies (Coelho et al., 2021).To tackle these challenges and leverage the opportunities, we delve into the details of our proposed approach in the next section, i.e., Methodology .

3 METHODOLOGY

To optimize latency in 5G networks, a structured approach using Q-learning, a reinforcement learning algorithm, is employed. The methodology begins with dataset generation, where MATLAB simulations emulate real-world 5G conditions like congestion and interference, producing a dataset with key attributes such as Signal-to-Noise Ratio (SNR), latency, and network load. This dataset is processed and structured using Python, ensuring it is suitable for machine learning applications. Preprocessing techniques like normalization and discretization are applied to standardize the data, enabling efficient mapping of states to actions and enhancing the learning process.

The proposed architecture incorporates edge computing to handle latency-sensitive decisions closer to the source, allowing the system to dynamically adapt to varying network conditions. The Q-learning framework employs a reward-driven mechanism, iteratively updating Q-values using the Bellman equation to optimize handover decisions and resource allocation. This integrated methodology effectively reduces latency across diverse scenarios, demonstrating scalability and efficiency in real-time 5G network optimization.

3.1 Dataset Generation and Description

MATLAB simulations and Python-based processing workflows were used in an integrated method to create the dataset for 5G network latency optimization. The absence of publicly available datasets created especially to solve latency optimization issues in 5G contexts served as the main driving force behind this hybrid approach. Realistic network circumstances were largely simulated using MATLAB, which included a number of crucial elements such as signal interference, environmental disturbances, congestion levels and device mobility. By producing raw data that represented important performance measures including latency, Signal-to-Noise Ratio (SNR), Signal-to-Interference-plus-Noise Ratio (SINR) and throughput, these simulations sought to replicate the dynamic and intricate nature of actual 5G networks. Python libraries were used for data formatting and structuring after the MATLAB simulations, allowing for smooth interaction with machine learning frameworks and guaranteeing that the data was appropriate for reinforcement learning applications. Before starting model training, the data was also put through an initial verification process to make sure it was accurate and consistent.

The dataset, which offers a comprehensive description of the network circumstances and device behavior, consists of twenty key attributes. Among these variables are SINR, SNR, transmission power, network load, ambient interference levels, latency, device mobility patterns and device proximity to base stations. Each data point, which is a snapshot of certain network conditions and device parameters at a specific time instance, provides a high-resolution image of the interactions inside a 5G network. The Q-learning model is trained on this comprehensive information, which gives it the contextual knowledge it needs to make wise decisions. The dataset guarantees that the model can generalize well across a variety of scenarios, from rural low-traffic areas to urban high-density areas, by precisely capturing the interaction of important variables.

To provide thorough coverage of a broad range of 5G network situations, the data generation procedure was carefully designed. The dataset replicates the subtleties of real-world network performance by simulating a variety of situations, including variable interference levels, varied mobility patterns, varying congestion levels and variations in base station coverage. This variety guarantees the Q-learning framework's flexibility in a variety of situations, enabling it to learn and make the best choices in both normal and unusual circumstances. Additionally, the dataset documents latency in various scenarios and the variables that affect it, including transmission power levels, device distance from the base station and network congestion. The dataset is specially suitable for specific latency optimization tasks due to its dual inclusion of latency metrics and influencing factors, which

also guarantees relevance to real-world 5G applications like industrial automation and driverless cars.

Efficiency and focus were key considerations in the design of the dataset's generation process. During the simulation phase, duplicated and unnecessary data were removed, preventing the need for intensive preprocessing. Every attribute in the dataset had a distinct and significant influence on the analytical results thanks to this proactive design. This method's effectiveness preserved the dataset's relevance to the optimization goals while also saving computational resources. To get rid of any possible irregularities that can interfere with learning, the dataset was further checked for consistency across all attributes. A clean and dependable input for model training was made possible by the meticulous curation that made sure the data was free of noise and irregularities.

The dataset was organized into several states that correspond to various network situations in order to aid reinforcement learning. Device mobility, distance to base stations, interference levels, network load, and channel quality indicators are just a few of the characteristics that are contained inside each state. The Q-learning model was able to assess actions in a dynamic environment because to this structured representation, which allowed it to adjust to shifting circumstances and determine the best latency reduction techniques. Because of the dataset's dynamic nature, the model was able to simulate and evaluate real-time situations, guaranteeing that the policies it had learned were useful and efficient for application in the actual world. The dataset's extensive coverage of network states and conditions not only facilitates reliable training but also improves the model's scalability and performance in diverse 5G environments. Thus, this extensive and painstakingly crafted dataset serves as the foundation for the Q-learning architecture, allowing for strong performance in 5G network latency optimization.

3.2 Proposed Architecture

The suggested architecture combines Q-learning with an improved state-action space to achieve a notable reduction in latency in 5G networks while guaranteeing scalability and efficiency. Using domain-specific thresholds, continuous variables like SINR, latency, and network load were discretized, lowering the dimensionality of the state space while preserving crucial differences required for efficient learning. The model can adjust to complex and dynamic network conditions without incurring undue processing overhead thanks to this discretization technique. In order to determine the best methods for reducing latency, the architecture uses a reinforcement learning framework to assess state-action pairings in real-time. The model dynamically modifies its strategies to guarantee reliable performance in a variety of 5G contexts by taking into account important contextual factors like device mobility, network congestion, and signal interference. Because of its versatility, the suggested solution—which is illustrated in Fig. 1. is a workable and dependable foundation for latency-sensitive applications in next-generation networks. 1

Along with the basic features of SINR, SNR, and latency, the architecture also incorporates base station data and device mobility to increase adaptability. The reward function is designed to promote reliable connections and low latency by penalizing unnecessary handovers and encouraging efficient resource allocation. This architecture can be utilized to optimize latency in real time in dynamic 5G applications due to its scalability and efficiency. By focusing on important network performance parameters and utilizing reinforcement learning techniques, the design provides a solid foundation for latency optimization in real 5G networks.

The Q-learning framework for latency optimization was created to choose actions iteratively according to the status of the network. Essential features including Base Station ID, discretized SNR, and discretized latency were combined to generate states. The Q-learning model was able to assess various network circumstances and determine whether to remain with the current base station or move to a different one thanks to this state representation. In order to minimize latency, a variety of activities were selected, each of which correlated to varying degrees of network load and delay reduction.

The Q-values, which indicate the anticipated future reward for every state-action pair, were updated using the Bellman equation. The Bellman equation was used to repeatedly update the Q-value for a particular state-action pair by combining the immediate reward with the anticipated future advantages. The model was motivated to decrease latency in subsequent actions by the reward function, which has an inverse relationship with latency. Higher values rewarded lower latency, directing the agent toward the best course of action. The following equation formalizes this process:

$$Q(s_t,a_t) = Q(s_t,a_t) + \alpha\Big(R_{t+1} + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)\Big) \tag{1}$$

The model continuously improves its decision-making capacity to minimize latency by refining its

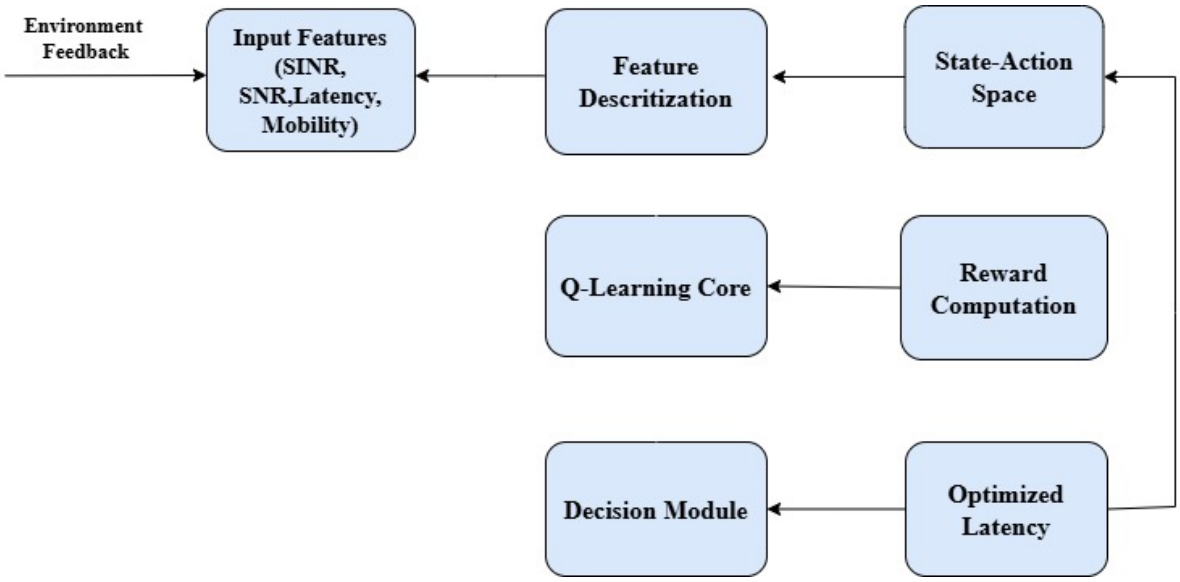

Figure 1: Proposed Architecture Diagram for Q-Learning.

Q-values over multiple epochs, as demonstrated in equation1. The exploration pace is modified to strike a balance between the exploitation of known activities and the exploration of novel ones. As the model improves its learnt policy and concentrates more on choosing actions that minimize latency, the exploration rate, which was initially set high, falls.

4 IMPLEMENTATION

In order to construct the Q-learning framework, network operations were simulated throughout 500 epochs, each of which represented a different 5G network situation. The Q-table, which was initially set to zero values to indicate the lack of prior knowledge regarding state-action pairs, is the central component of this system. During training, the Q-table was iteratively updated to reflect the best latency-reduction strategies. To guarantee algorithm compatibility, the dataset underwent necessary preprocessing procedures like normalization and discretization. To facilitate effective state-action mapping, continuous data—like SINR, latency, and network load—were transformed into discrete states. Three possible actions were associated with each state: "Adjust Power," which involved changing the transmission power levels; "Switch," which involved switching to a different base station to improve the signal quality; and "Stay," which involved keeping the present connection for stability. These steps were thoughtfully created to deal with latency issues in a variety of dynamic network scenarios.

The framework's reward function, mathematically defined as:

$$R = \frac{1}{\text{Latency} + \varepsilon} \tag{2}$$

plays a critical role in guiding the learning process. Here, R represents the reward for a specific state-action pair, inversely correlating with the observed latency to ensure higher rewards for lower latency. The term Latency captures the network delay in a given state, while ε, set at 0.0002, prevents numerical instability by ensuring the denominator remains non-zero, particularly in low-latency scenarios. This formulation enabled the framework to prioritize actions that significantly reduced delays while maintaining numerical stability. Each update to the Q-table was governed by the Bellman equation, progressively refining the decision-making policy by incorporating the rewards obtained during simulations.

A dynamic exploration-exploitation technique improved the framework's decision-making and ensured balanced learning during training. In order to gather thorough knowledge about the surroundings, the agent first investigated a large variety of state-action pairs. The exploration rate dropped as training went on, enabling the agent to concentrate on using learnt policies to make the best decisions. In order to prevent local optima and achieve reliable latency reduction solutions, parameters including the learning rate, discount factor, and exploration rate were carefully adjusted to guarantee smooth convergence to an ideal policy.

Extensive experiments were carried out in a vari-

ety of network situations, such as different levels of congestion, mobility patterns, and interference scenarios, in order to confirm the framework's performance. Metrics like latency reduction, decision correctness, and the agent's capacity for environment adaptation were the main emphasis of the evaluation. The outcomes showed how effective the framework was in reducing latency, outperforming traditional techniques by a considerable margin. The Q-learning-based model demonstrated its flexibility under real-world 5G network conditions by utilizing iterative Q-table updates and a well-structured reward function. This demonstrated its dependability for latency-sensitive applications like augmented reality and driverless cars. This framework's promise as a foundation for upcoming 5G optimization initiatives is highlighted by the smooth integration of pretreatment processes, strategic exploration, and adaptive learning.

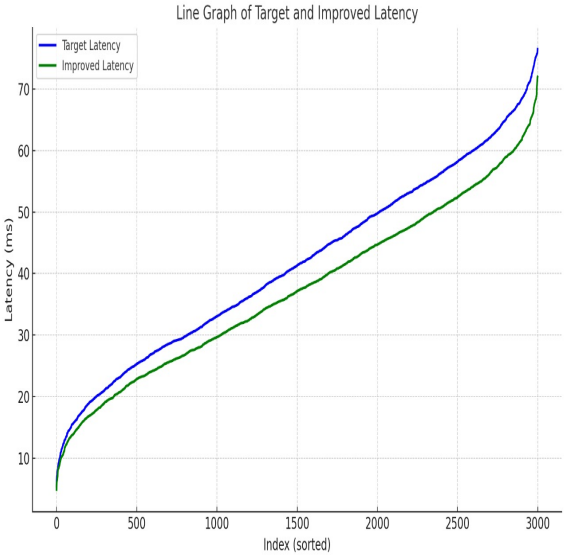

Figure 2: Line Graph of Target latency and Improved latency

5 RESULTS AND ANALYSIS

The outcomes of the trial showed how well the Q-learning framework worked to lower latency in 5G networks. The method achieved a notable 58.16% reduction in latency, outperforming conventional WebRTC-based remote control (Mtowe and Kim, 2023). Extensive simulations comparing the goal latency values with the improved latency following model training were used to verify this. The latency improvement ranged from a maximum of 15 ms to a minimum of 3 ms and an average of 9 ms for all devices. Device ID 3 demonstrated a decrease in latency from 19.40 ms to 16.49 ms, whilst Device ID 0 demonstrated an improvement from 57.82 ms to 54.93 ms. These findings support the method's promise for real-time latency optimization in 5G settings, especially for applications that are sensitive to latency.

The Q-learning framework's flexibility and resilience are demonstrated by the decrease in latency across various devices. The enhanced latency figures attest to the model's ability to efficiently optimize resource allocation and handover choices across a range of network scenarios. The agent was given the best methods for reducing latency by the Q-table, which was created through iterative training. The dataset, which comprised a variety of network settings and characteristics, provided a strong basis for model training, guaranteeing that the outcomes were indicative of actual situations. Table 1 further illustrates the efficacy of the framework by summarizing the comparison between the target and improved latency values for a subset of devices.

Figure 2 provides a graphical comparison of target latency (depicted by the blue line) and improved latency (represented by the green line) achieved using the proposed framework. The x-axis corresponds to the sorted sample index, while the y-axis represents latency in milliseconds. The consistent separation between the two lines highlights the framework's capability to significantly reduce latency, with pronounced improvements observed at higher latency ranges. These outcomes demonstrate the feasibility of using the framework for real-time latency optimization, making it particularly suitable for applications like autonomous vehicles, augmented reality, and industrial automation.

The reduction in latency across a range of devices showcases the Q-learning framework's adaptability and robustness. The improved latency values reflect the framework's ability to optimize resource allocation and handover decisions effectively under diverse network conditions. Iterative training of the Q-table allowed the agent to identify optimal actions for latency minimization. The training dataset, composed of various network configurations and parameters, ensured that the model's performance was representative of real-world scenarios. A detailed comparison of target and improved latency values (in ms) for selected devices is presented in Table 1, further demonstrating the framework's efficiency.

Table 1: Latency Comparison: Target vs. Improved Latency

Device ID	Target Latency	Improved Latency
0	57.82	54.93
1	22.40	20.32
2	45.75	43.17
3	19.40	16.49
4	30.12	28.04

6 CONCLUSION AND FUTURE SCOPE

The effectiveness of a Q-learning-based architecture for 5G network latency optimization is demonstrated in this study. In dynamic network scenarios, the system employs preprocessing techniques and a customized dataset to efficiently reduce latency and enhance decision-making. Streamlining the state space enhances the learning process and computational efficiency, making the proposed model suitable for real-time applications in 5G environments. The integration of edge computing addresses the demands of latency-sensitive tasks, such as driverless cars, smart cities, and augmented reality systems. The architecture leverages adaptive reinforcement learning strategies to handle varying network conditions and optimize resource allocation, establishing a dependable framework for latency-critical applications in next-generation networks

Future studies can enhance scalability and accuracy by incorporating parameters like multi-cell interference, user behavior, and advanced machine learning methods, while emphasizing standardized protocols and security for broader 5G adoption.

REFERENCES

Abouaomar, A., Cherkaoui, S., Mlika, Z., and Kobbane, A. (2021a). Resource provisioning in edge computing for latency-sensitive applications. *IEEE Internet of Things Journal*, 8(14):11088–11099.

Abouaomar, A., Cherkaoui, S., Mlika, Z., and Kobbane, A. (2021b). Resource provisioning in edge computing for latency-sensitive applications. *IEEE Internet of Things Journal*, 8(14):11088–11099.

Abouaomar, A., Cherkaoui, S., Mlika, Z., and Kobbane, A. (2022). Resource provisioning in edge computing for latency sensitive applications.

Coelho, C. N., Kuusela, A., Li, S., Zhuang, H., Ngadiuba, J., Aarrestad, T. K., Loncar, V., Pierini, M., Pol, A. A., and Summers, S. (2021). Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors. *Nature Machine Intelligence*, 3(8):675–686.

Djigal, H., Xu, J., Liu, L., and Zhang, Y. (2022). Machine and deep learning for resource allocation in multi-access edge computing: A survey. *IEEE Communications Surveys & Tutorials*, 24(4):2449–2494.

Hartmann, M., Hashmi, U. S., and Imran, A. (2022). Edge computing in smart health care systems: Review, challenges, and research directions. *Transactions on Emerging Telecommunications Technologies*, 33(3):e3710.

Hassan, N., Yau, K.-L. A., and Wu, C. (2019). Edge computing in 5g: A review. *IEEE Access*, 7:127276–127289.

Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., and Young, V. (2015). Mobile edge computing—a key technology towards 5g. *ETSI white paper*, 11(11):1–16.

Intharawijitr, K., Iida, K., and Koga, H. (2017a). Simulation study of low latency network architecture using mobile edge computing. *IEICE TRANSACTIONS on Information and Systems*, 100(5):963–972.

Intharawijitr, K., Iida, K., Koga, H., and Yamaoka, K. (2017b). Practical enhancement and evaluation of a low-latency network model using mobile edge computing. In *2017 IEEE 41st annual computer software and applications conference (COMPSAC)*, volume 1, pages 567–574. IEEE.

Kodavati, B. and Ramarakula, M. (2023). Latency reduction in heterogeneous 5g networks integrated with reinforcement algorithm.

Liang, S., Jin, S., and Chen, Y. (2024). A review of edge computing technology and its applications in power systems. *Energies*, 17(13).

Liu, J., Luo, K., Zhou, Z., and Chen, X. (2018). Erp: Edge resource pooling for data stream mobile computing. *IEEE Internet of Things Journal*, 6(3):4355–4368.

Mtowe, D. P. and Kim, D. M. (2023). Edge-computing-enabled low-latency communication for a wireless networked control system. *Electronics*, 12(14).

Parvez, I., Rahmati, A., Guvenc, I., Sarwat, A. I., and Dai, H. (2018). A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys & Tutorials*, 20(4):3098–3130.

Shokrnezhad, M., Taleb, T., and Dazzi, P. (2024). Double deep q-learning-based path selection and service placement for latency-sensitive beyond 5g applications. *IEEE Transactions on Mobile Computing*, 23(5):5097–5110.

Singh, S., Chiu, Y.-C., Tsai, Y.-H., and Yang, J.-S. (2016). Mobile edge fog computing in 5g era: Architecture and implementation. In *2016 International Computer Symposium (ICS)*, pages 731–735.

Sumathi, D., Karthikeyan, S., Sivaprakash, P., and Selvaraj, P. (2022). Chapter eleven - 5g communication for edge computing. In Raj, P., Saini, K., and Surianarayanan, C., editors, *Edge/Fog Computing Paradigm: The Concept Platforms and Applications*, volume 127 of *Advances in Computers*, pages 307–331. Elsevier.

Toka, L. (2021). Ultra-reliable and low-latency computing in the edge with kubernetes. *Journal of Grid Computing*, 19(3):31.

Wang, T. (2020). Application of edge computing in 5g communications. In *IOP Conference Series: Materials Science and Engineering*, volume 740, page 012130. IOP Publishing.

Wu, Y., Zhang, X., Ren, J., Xing, H., Shen, Y., and Cui, S. (2024). Latency-aware resource allocation for mobile edge generation and computing via deep reinforcement learning.

Yajnanarayana, V., Rydén, H., and Hévizi, L. (2020). 5g handover using reinforcement learning.

Zhang, J., Xie, W., Yang, F., and Bi, Q. (2016). Mobile edge computing and field trial results for 5g low latency scenario. *China Communications*, 13(2):174–182.