TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 07/31 pp100–111
DOI: 10.26599/TST.2024.9010033
Volume 30, Number 1, February 2025

Scheduling of Low-Latency Medical Services in Healthcare Cloud with Deep Reinforcement Learning

Hongfei Du, Ming Liu, Nianbo Liu, Deying Li, Wenzhong Li, and Lifeng Xu*

Abstract: In the current landscape of online data services, data transmission and cloud computing are often controlled separately by Internet Service Providers (ISPs) and cloud providers, resulting in significant cooperation challenges and suboptimal global data service optimization. In this study, we propose an end-to-end scheduling method aimed at supporting low-latency and computation-intensive medical services within local wireless networks and healthcare clouds. This approach serves as a practical paradigm for achieving low-latency data services in local private cloud environments. To meet the low-latency requirement while minimizing communication and computation resource usage, we leverage Deep Reinforcement Learning (DRL) algorithms to learn a policy for automatically regulating the transmission rate of medical services and the computation speed of cloud servers. Additionally, we utilize a two-stage tandem queue to address this problem effectively. Extensive experiments are conducted to validate the effectiveness for our proposed method under various arrival rates of medical services.

Key words: medical service; tandem queue; cloud computing; Deep Reinforcement Learning (DRL); resource allocation

1 Introduction

In the current landscape of online data services, the control of data transmission and cloud computing is often fragmented between ISPs and cloud providers. This separation leads to substantial cooperation challenges and suboptimal optimization for global data services. ISPs and cloud providers operate distinct infrastructures and pursue different profit patterns, resulting in diverse approaches to support online data services with varying service requirements. An illustrative example is the decline of Quality of Service (QoS) and the rise of Quality of Experience (QoE). ISPs perceive QoS as the ability to assign different priorities to applications, users, or data flows, or to ensure a certain level of performance for a given data flow. This approach treats network conditions as a black box and necessitates explicit and precise service requirements from end users, which may prove challenging to implement across numerous applications. On the other hand, cloud providers prioritize QoE, which gauges the satisfaction or dissatisfaction of customers' experiences with a service. They proactively assess network conditions without active involvement from ISPs to deliver a

• Hongfei Du, Ming Liu, and Nianbo Liu are with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China. E-mail: duhongfei@ydnkj.com; csmliu@uestc.edu.cn; liunb@uestc.edu.cn.
• Deying Li is with School of Information, Renmin University of China, Beijing 100872, China. E-mail: deyingli@ruc.edu.cn.
• Wenzhong Li is with Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China. E-mail: lwz@nju.edu.cn.
• Lifeng Xu is with People's Hospital of Quzhou City, Wenzhou Medical University, Quzhou 324000, China. E-mail: qz1109@wmu.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2023-10-26; revised: 2024-01-16; accepted: 2024-02-03

holistic service experience. Consequently, any scheduling of data transmission and cloud computing must navigate the opaque nature of information and cooperation challenges between ISPs and cloud providers.

Simultaneously, the rapid growth of the Internet of Things (IoTs) has facilitated the emergence of numerous low-latency and computation-intensive services, necessitating support through edge computing or cloud computing at the user's end. It brings a pressing need for a practical paradigm that effectively coordinates data transmission and processing to enable low-latency data services within local networks. In the healthcare domain, the global IoT in healthcare market is projected to reach a value of 188.2 billion USD by 2025[1]. This growth can be attributed to the increasing public awareness regarding personal health, which has led to a rising demand for various types of smart medical wearable devices, such as ECG monitors, heart rate monitors, and blood pressure monitors. These devices enable the acquisition of real-time health and fitness information. In addition to personal medical devices, hospitals also deploy a multitude of dedicated IoT medical devices, including smart electrocardiographs and smart call devices. However, despite these advancements, it is important to note that certain critical medical services, such as real-time patient monitoring, virtual consultations, and Augmented Reality/Virtual Reality (AR/VR)-based surgeries[2, 3], which require low latency and intensive computation, may still face challenges in meeting their requirements.

In recent years, numerous researchers have explored the potential of cloud computing in the IoT and cloud-based healthcare domains to provide intelligent medical services[4–9]. However, healthcare cloud platforms face a unique challenge with high latency in many scenarios, which is particularly unacceptable due to the potential risks it poses to patients' health and even their lives. Some studies propose the use of fog computing-based healthcare systems to minimize latency between medical IoT devices and cloud servers. Fog computing servers, being geographically closer to the devices, offer a potential solution for reducing latency[10–16]. However the deployment of fog nodes incurs substantial costs, and the communication and collaboration between fog nodes pose non-trivial challenges, making them less suitable for small or medium-sized hospitals.

As mentioned earlier, all traditional methods have focused on optimizing the computational resources and computational latency at the server side, without considering the latency during the transmission processes. One of unique advantages of healthcare cloud is that both the transmission network and cloud services are controlled by the hospital's IT department, which enables effective joint optimization of end-to-end latency and computational resources while ensuring statistical latency guarantees within local IoT and private cloud networks, as shown in Fig. 1. To bridge this gap, we propose a two-stage tandem queue consisting of a communication queue and a computation queue[17–19]. In order to satisfy latency constraints without wasting communication and computation resources, a well-designed resource allocation policy is required to effectively utilize cloud computing. In this paper, we introduce a central controller that dynamically adjusts bandwidth and computation speed on a global scale. Specifically, allocating more bandwidth resources corresponds to lower transmission latency, while faster computation speed leads to reduced processing latency. We model the process of IoT devices accessing cloud computing services as a tandem queue. The primary objective of this paper is to ensure that the service latency remains below the specified latency requirement for medical services, while minimizing the utilization of communication and computation resources. To achieve this, we employ a DRL algorithm, which is a natural choice for addressing sequential decision-making problems. Through the DRL algorithm, we develop an intelligent policy for the controller that automatically regulates the transmission rate and computation speed in an optimal manner.

The structure of this paper is organized as follows: Section 2 presents the related work in this field. In Section 3, we describe the system model and define the problem. Section 4 provides an overview of the related materials and presents our proposed methods. The

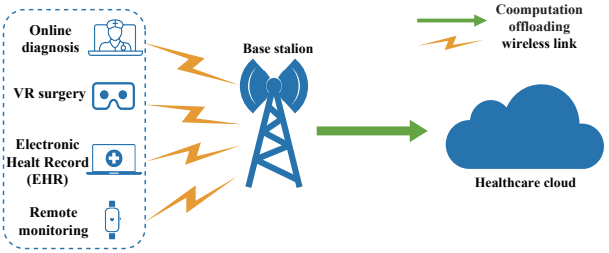

Fig. 1 Medical services in healthcare cloud.

training of the model, simulation details, and evaluation results are discussed in Section 5. Finally, in Section 6, we summarize the conclusions of this paper.

2 Related Work

Due to the increasing adoption of IoT and cloud-based healthcare systems, numerous researchers have focused on leveraging IoT technologies and cloud computing to enable intelligent medical services, thereby enhancing the experiences of patients and improving the efficiency of healthcare providers. For example, Zgheib et al.[4] developed a human activity detection application capable of handling data from diverse IoT devices in a monitoring environment, providing semantic detection of activities. Tariq et al.[5] introduced a technique and an augmented dataset to improve the detection of daily assistive activities, with a particular emphasis on sitting posture, in IoT-driven environments. In Ref. [6], the authors created an integrated medical platform for remote health monitoring, collecting vital signs and living environment information from patients and transmitting it to the cloud for processing and analysis. Marques and Pitarma[7] developed an IoT-based indoor air quality system that offers a practical assessment of indoor air quality for subsequent interventions to enhance air quality.

Furthermore, the field of Internet of m-Health things, which combines mobile computing, medical sensors, and cloud computing, has gained significant attention. Erdeniz et al.[20] proposed a novel recommender system that suggests healthcare devices, new applications, and physical activity plans for patients in IoT-enabled mobile health applications. Xu et al.[21] presented a framework based on cloud computing and mobile computing for pervasive health monitoring, incorporating three layers: a data storage layer for ensuring patient privacy, a data annotation layer for enhancing health data interoperability, and a data analysis layer for executing mining algorithms. Wearable devices have also made remarkable progress, enabling the monitoring of various patient biosignals to benefit both patients and doctors. Kelati et al.[22] introduced a battery-powered wearable IoT system with a microcontroller for data processing and wireless transmission of patient biosignals.

However, addressing latency requirements remains a challenging problem that, if not effectively resolved, can result in serious medical negligence. Consequently, researchers have engaged in extensive research to find solutions for reducing latency in IoT and cloud-enabled medical services. Mahmud et al.[12] proposed a cloud-fog-based IoT healthcare structure with interoperability and coordination to support low-latency services. To mitigate latency issues introduced by the cloud paradigm, Ref. [13] proposes an architecture for IoT-based health monitoring that leverages fog computing's proximity to IoT medical devices. Awaisi et al.[14] introduced a fog-based healthcare architecture for IoT, utilizing virtual machine partitioning technology to significantly reduce latency and improve network usage. They also proposed a user authentication method to protect patient privacy through an identity management system. In Ref. [15], the authors proposed a novel framework called healthfog, which combines edge computing and ensemble deep learning algorithms for time-critical automatic heart disease analysis. To address pressing challenges such as resource limitations, low-latency provision, and massive data processing, Ren et al.[16] proposed a task offloading strategy with a centralized decision-making algorithm that combines software-defined networking and blockchain technologies in a fog-assisted healthcare system.

Generally, the aforementioned literature primarily focuses on fog computing to support low-latency service provision without considering the excessive use of communication and computation resources. We consider a local IoT-cloud healthcare system in our early work[23], and develop a DRL algorithm to automatically regulate the transmission rate and computation speed in this study.

3 Problem Formulation

3.1 System model

Figure 1 illustrates the architecture of our IoT and cloud-based healthcare system, which comprises multiple IoT medical devices, a Base Station (BS), and a healthcare cloud platform. All these devices can connect to the cloud computing platform through a wireless link. We assume the existence of homogeneous relay servers in the BS and computation servers in the healthcare cloud. Initially, a medical task is transmitted to the BS and then forwarded to the healthcare cloud if there are available idle relay servers. Otherwise, it is buffered in a queue. Similarly, the medical task is processed if there are idle

computation servers, and buffered in a queue otherwise[24].

To model the system, we abstract it as a two-stage tandem queue, comprising a communication queue and a computation queue. This queuing model follows the first-come-first-serve discipline to ensure efficient and reliable provision of medical services, as depicted in Fig. 2. Notably, we do not assume any specific arrival rate distribution for medical services, making the arrivals completely random and thus more realistic. Consequently, it is crucial to dynamically adjust resource allocation to prevent wastage of communication and computation resources.

To address this challenge, we propose the concept of a central controller that sets the transmission rate and computation speed at the beginning of each time slot. These values remain constant throughout the time slot in this environment, as shown in Fig. 3. At the start of the time slot, we have two queue lengths as q_1 and q_2 and two adjust actions a_1 and a_2 on the two queues. Moregenerally, the queue length is denoted as q_n, and the service latency, denoted as d, represents the duration from when a task enters the communication queue to when it leaves the computation queue. We also define $d_{\max}$ as the maximum tolerable latency for medical services. The objective of the proposed controller is to determine appropriate transmission rates and computation speeds for relay servers and computation servers, respectively, that satisfy the latency requirement while minimizing resource usage.

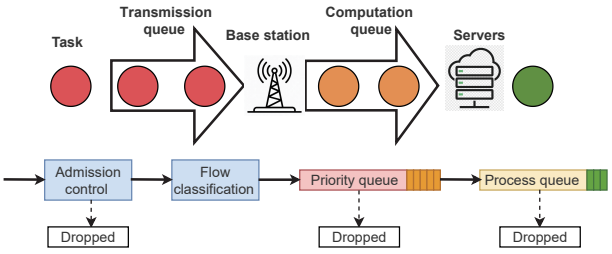

Fig. 2 Two-stage tandem queue.

3.2 Problem formulation

We formulate our problem within the context of a two-stage tandem queue system, which consists of a communication queue and a computation queue, both equipped with multiple servers. The tandem queue model is commonly employed when there are multiple queues involved, each with one or more servers. This model is particularly suitable for scenarios where tasks or services can be divided into independent procedures that need to be executed in a specific order[25, 26]. Given its applicability to our problem, it serves as an excellent candidate for our study.

Our objective is to design a central controller that minimizes the latency of each medical service, ensuring it remains below the threshold of $d_{\max}$. However, due to the limited availability of communication and computation resources, there exists a tradeoff between service latency and resource utilization. Hence, the problem is formulated to seek the minimun expected value of the total resources as follows:

$$\max_{\theta} -E\left[\rho_1 m_1 c + \rho_2 m_2 s\right] \tag{1}$$

$$\text{s.t., } P\left(d < d_{\max}\right) > \epsilon \tag{2}$$

$$c_{\min} \leqslant c \leqslant c_{\max} \tag{3}$$

$$s_{\min} \leqslant s \leqslant s_{\max} \tag{4}$$

$$m_n \geqslant 1,\ n \in \{1, 2\} \tag{5}$$

In our problem formulation, we introduce the variables m_n to represent the number of relay servers or

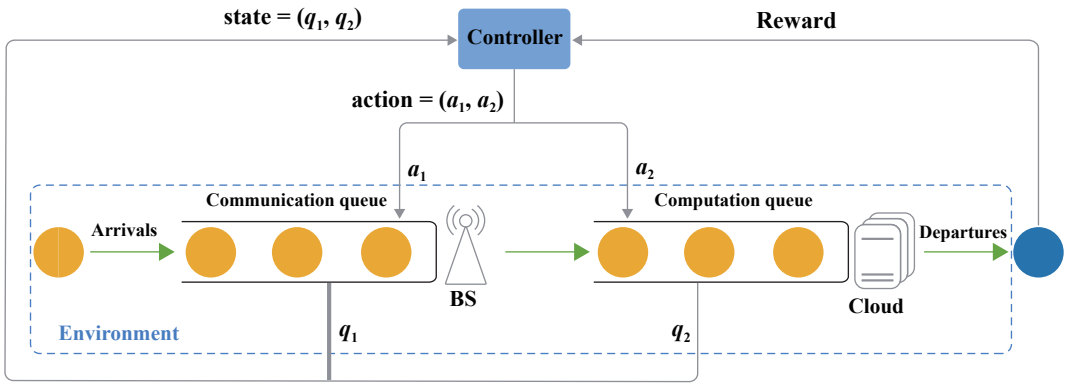

Fig. 3 Our proposed RL framework.

computation servers, and ρ_n as the fixed cost coefficient associated with communication and computation. The transmission rate is denoted as c, the task latency denoted as d, while the computation speed of cloud servers is represented by s. Furthermore, the values of $c_{\min}$, $c_{\max}$, $s_{\min}$, and $s_{\max}$ are determined based on the available communication and computation resources.

Our primary objective is to minimize the total utilization of resources, taking into consideration both communication and computation aspects. To achieve this, we aim to optimize the allocation of relay servers, computation servers, transmission rates, and computation speeds. Additionally, we introduce the parameter ϵ, a small positive integer, which allows us to control the number of medical services that exceed the maximum allowable latency. By minimizing the utilization of resources while maintaining service latency within acceptable limits, we aspire to minimize the number of medical services that experience latency violations.

In the given problem context, the selection of transmission rate (c) and computation speed (p) directly influences task latency (d). These two factors assume a crucial role in determining the overall system performance. Our problem involves a two-stage serial queue system for tasks, namely the communication queue and the computation queue. The following elucidates the impact of transmission rate and computation speed choices on task latency.

Selection of transmission rate (c) affects latency in the communication queue and potential resource contention.

Latency in the communication queue: The transmission rate governs the waiting time of tasks in the communication queue. A higher transfer rate generally facilitates faster movement of tasks through the communication queue, thereby reducing communication delays.

Potential resource contention: However, higher transfer rates may lead to contention with other tasks or service resources, which could increase overall latency in certain cases.

Selection of computation speed (s) influences latency in the computation queue and resource utilization and contention.

Latency in the computation queue: Computation speed directly correlates to the processing time of tasks in the computation queue. Higher computing speed typically diminishes computing latency.

Resource utilization and contention: Nevertheless, opting for a higher computation speed might entail increased resource consumption, possibly resulting in less efficient resource utilization and subsequently augmenting the overall latency between communication and computation.

In conclusion, the selection of transmission rate and computation speed necessitates a trade-off process that involves comprehensive consideration of the communication queue and computation queue characteristics to minimize overall system delay. The optimal balance between these two factors depends on specific problem requirements and the availability of resources.

4 Method

4.1 Materials of reinforcement learning

In this section, we begin by introducing the fundamental concepts of Reinforcement Learning (RL) as one of the three fundamental machine learning paradigms. RL involves an agent and an environment as its core components. The agent perceives the environment through sensors and takes actions that can influence the environment. Unlike supervised learning, which heavily relies on labeled datasets, RL learns through interaction with the environment[27]. At each time slot, the agent receives observations from the environment and selects an action a_t based on a learned policy $\mu(a|s)$, representing the probability of taking action a_t in state s_t. Subsequently, the state transitions from s_t to s_{t+1}, and the agent receives a reward r_t or a reward with a mapping relationship $R(s_t, a_t, s_{t+1})$ from the environment. The total discounted reward from time slot t to infinity is defined as $\sum_{k=t}^{\infty}\gamma^{k-t}r_k$, where $0<\gamma<1$. The factor γ determines the importance of past rewards, with smaller values assigning less weight to earlier rewards. Therefore, the agent's goal is to search for a policy that maximizes the total rewards. One approach is to store state-action pairs in a table and update it iteratively until convergence[28]. However, this method becomes impractical when dealing with large state spaces. Another approach involves using artificial neural networks as function approximators to address this limitation[29].

To solve RL problems with continuous actions, policy gradient methods are commonly employed. These methods directly model and optimize the policy

to find the optimal solution. The policy function is denoted as $\pi_\theta(a|s)$, parameterized by θ. According to the policy gradient theorem[30], the gradient is defined as

$$\nabla_\theta J(\theta) = E_\pi [Q^\pi(s, a) \nabla\theta \ln \pi_\theta(a|s)] \quad (6)$$

However, evaluating the state-action value function $Q^\pi(s, a)$ can be computationally complex. Therefore, researchers often approximate the expectation using Monte Carlo methods[31]. Additionally, the actor-critic algorithm, another popular policy gradient approach, is used to address this challenge. The actor-critic method consists of two components, the actor and the critic, which work together to improve policy updates. The actor updates the parameters θ of the policy function $\pi_\theta(a|s)$ based on feedback from the critic, while the critic updates the parameters w of the value function $Q_w(a|s)$. Finally, we adopt the Deep Deterministic Policy Gradient (DDPG) algorithm, an offline-policy algorithm that combines deep neural networks with Deterministic Policy Gradient (DPG). DDPG improves upon DPG, especially in scenarios with large state spaces, by leveraging neural networks. Additionally, it incorporates a replay buffer mechanism to address the challenges of correlated data and non-stationary distribution. Furthermore, to enhance training stability, DDPG employs target networks that are periodically updated based on the online network.

4.2 Design of DRL model

Our proposed framework comprises three modules: a central controller responsible for determining the transmission rate and computation speed, a communication queue for storing data to be transmitted, and a computation queue for processing tasks. Figure 3 illustrates the architecture. Initially, both the communication queue and computation queue are empty, and the parameters of the actor and critic networks are randomly initialized. Medical tasks are continuously generated according to an arbitrary distribution. These tasks are first placed in the communication queue, and then forwarded to the healthcare cloud for processing. At the beginning of each time step, the controller outputs the transmission rate and computation speed, which remain constant throughout the time step. At the end of the time step, feedback is obtained from all completed medical tasks. The controller then updates its network parameters, and the queue lengths are also updated. This iterative process continues until the controller converges to a policy that ensures low latency.

We will begin to definie the three fundamental components of the RL problem: state, action, and reward. Furthermore, we will present more detailed design aspects of our proposed approach.

State is represented by a vector containing the lengths of the communication queue and computation queue, denoted as $s = (q_1, q_2)$.

Action is obtained by evaluating the deterministic policy $a = \mu_\theta(s)$. This action is defined as a vector that contains the values outputted by the controller, including the transmission rate c and computation speed p, which remain constant during the time step.

Reward is often the most challenging aspect of an RL problem, as it greatly influences the algorithm's convergence, correctness, and robustness. Therefore, it is crucial to set the reward function carefully to reflect the controller's performance after taking an action in one time slot. Let T represent the duration of one time step, and let $\mathcal{A}_t$ denote the set of medical task arrivals at time slot t. The immediate reward r' for each medical task within the set $\mathcal{A}_t$ at time slot t is defined as two punishment values β_1 and β_2 as follows:

$$r' = \begin{cases} \beta_1, & d \leqslant d_{\max}; \\ \beta_2, & d > d_{\max} \end{cases} \quad (7)$$

In addition to considering latency, we also need to account for resource costs, such as communication and computation resources. Hence, we calculate the total cost s_t^{sum} at time slot t in the following:

$$s_t^{\text{sum}} = \rho_1 m_1 c^t + \rho_2 m_2 p^t \quad (8)$$

where ρ_1 and ρ_2 represent the cost coefficients for communication and computation, respectively, c^t and p^t represent the transmission rate and computation speed during the time step, respectively. Consequently, the immediate reward r_t at time slot t can be calculated as the sum of r' values for all medical tasks within $\mathcal{A}_t$ minus s_t^{sum},

$$r_t = r(s_t, \mu_\theta(s_t)) = \sum_{\mathcal{A}_t} r' - s_t^{\text{sum}} \quad (9)$$

Then, according to Eqs. (6) and (9), the average reward is defined as follows:

$$J(\mu_\theta) = E_{s\sim\rho^\mu(s)} [r(s, \mu_\theta(s))] \quad (10)$$

where $\rho^\mu(s)$ stands for distribution of the states following policy μ, and we can get the gradient of Eq. (10),

$$\nabla_\theta J(\mu_\theta) = E_{s\sim\rho^\mu(s)}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right] \quad (11)$$

In Eq. (11), the expectation is taken only with respective to state space and thus avoiding the problem of state space explosion.

The design specifics of the central controller are outlined in the subsequent section. Figure 4 illustrates the four components: environment, actor network, critic network, and experience replay buffer. In the actor network, the input consists of a sequence of vectors that undergoes two hidden layers with ReLU activation, each comprising 32 neurons. The output layer of the actor network is a fully-connected neural network activated by the hyperbolic tangent function, ranging from −1 to 1. However, since the transmission rate and computation speed must be positive, we rescale the output range to obtain valid values. Additionally, to explore potentially high-reward actions, we introduce random noise to the final output and apply a clipping operation to satisfy conditions of Eqs. (3) and (4).

In contrast, the critic network takes the input and passes it through a fully-connected network. The output is then concatenated with the output of another linear layer, which takes the action outputted by the actor network as input. The resulting concatenated output is fed into another hidden layer with ReLU activation. Unlike the actor network, the final output of the critic network is activated by a linear function. The training process is described in the following paragraph.

It is important to note that the target network has the same structure as the online network for both the actor and the critic. As depicted in Fig. 4, the online network overlaps with the target network. The optimization of network parameters for the actor and critic networks continues until the model converges. At each time slot, the final action a_t is generated by the actor network $\mu(s_t)$, with the addition of a noise value. Subsequently, the environment transitions from state s_t to s_{t+1} and provides an immediate reward. The experience replay buffer stores tuples (s_t, a_t, r_t, s_{t+1}) for later sampling and learning from a randomly-selected batch of quadruples $N^*(s_i, a_i, r_i, s_{i+1})$. Moreover, the buffer has a fixed capacity, meaning it only retains the most recent experiences, requiring careful selection of the capacity value. the critic network, actor network, and their corresponding target networks are updated by sampling from the experience replay buffer. To stabilize the update process, we employ a soft update approach to gradually update both the actor target network and the critic target network.

5 Experiment

In this section, we present the experimental settings, train the DRL model in a predefined environment, and run up a series of comparison experiments to demonstrate the rationality and validity of our proposed model.

5.1 Experiment settings

At first, we employ Python to simulate a two-stage tandem queue environment. In this simulation, we utilize the gamma distribution to model both the inter-arrival times and service times. Given the absence of a terminal state in our problem, we introduce a variable Maxsteps to halt the learning process within an episode if the number of time steps exceeds this threshold. At the onset of each episode, we initialize the environment to encompass a diverse range of states. Specifically, we set $d_{\max}$ to 5 and T to 15 ms.

To address the limited applicability of static datasets, we adopt dynamic datasets by generating the arrival

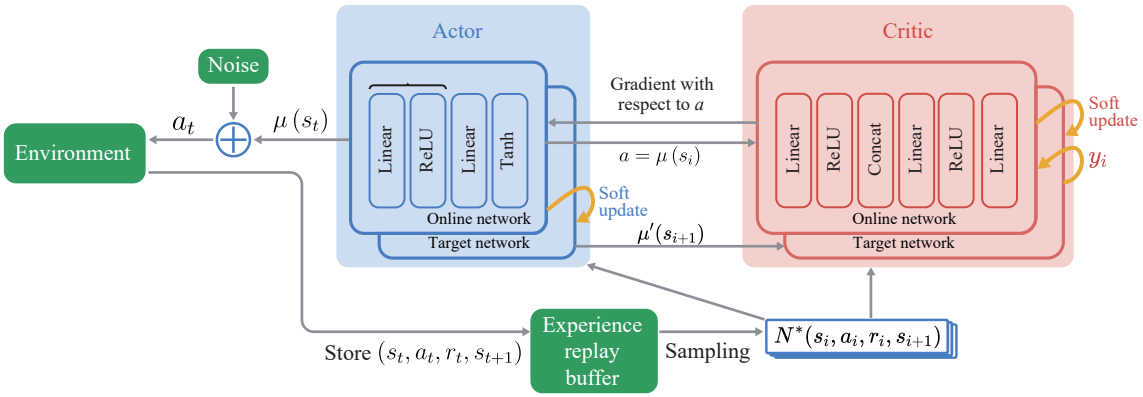

Fig. 4 Implementation of our proposed model.

rates and service times of medical services using arbitrary distributions, as discussed in previous works[32, 33]. This approach allows us to tackle the issue of a narrow range of application for static datasets and ensures a more realistic representation of the healthcare system dynamics.

The DRL algorithm is implemented using PyTorch v1.0. All experiments are conducted on an Ubuntu server equipped with an Intel Xeon E5-2640v4 CPU, 128 GB of memory, and a Tesla M40 8 G GPU. More training parameters can be found in Table 1. At the sametime, we set $\gamma = 0.99$, $\tau = 0.01$, number of episodes = 1000, and adopt Ornstein-Uhlenbeck noise with the settings $\theta = 0.15$, $\mu = 0$, $\sigma = 0.5 - 0.05$, and annealing steps = 53 200.

In our subsequent experiments, we define four metrics to evaluate the performance of our proposed model: average resource score, average service latency, latency violation probability, and average reward. These metrics provide comprehensive insights into the resource utilization, latency management, and overall system performance achieved by our model.

5.2 Experiment results

We conduct training for our DRL model within the predefined environment, and the results are depicted in Fig. 5, which illustrates the convergence of average service latency, average resource score, and average reward per time step as a function of episodes. To ensure robustness, we train the DRL model multiple times with random initialization and select the best-performing result. To promote exploration, we employ the Ornstein-Uhlenbeck random process, which generates noise added to the output of the actor network. The core parameter of this process, σ, gradually decreases from 0.5 to 0.05. As a result, the blue curve exhibits significant fluctuations in the initial 350 episodes and gradually stabilizes thereafter. Notably, our proposed model demonstrates fast convergence, and the average service latency remains below the threshold of $d_{\max}$.

To validate the efficacy of our proposed model, we compare it with two baselines. The first baseline, called Adjusting with Queue Length (AQL), adjusts the transmission rate and computation speed based solely on queue length. Specifically, when the queue lengths of communication and computation queues increase, the controller increases the transmission rate and computation speed to expedite the processing of medical services, and vice versa. The second baseline, called Mean Value with Noise (MVN), instructs the controller to output the mean value of available resources with the addition of random noise. We conduct performance comparison experiments using test data generated from the predefined environment with identical parameters.

As depicted in Fig. 6, our proposed model achieves a satisfactory resource allocation, meeting the latency requirements of medical services while minimizing resource usage. It is evident that the Mean Value with Noise (MVN) controller wastes a significantly larger amount of resources compared to the other two methods. However, lower resource usage brings higher violation probability, therefore, we evaluate the performance of all methods under increased arrival rates ($\eta_a = 1.5$) and decreased arrival rates ($\eta_a = 0.5$) of

Table 1 Training parameters of DRL model.

Parameter	Actor	Critic
Numberof linear nodes	32	32
Activation function	ReLU	ReLu
Output function	Tanh	Linear
Learning rate	0.0001	0.001
Batch size	128	128

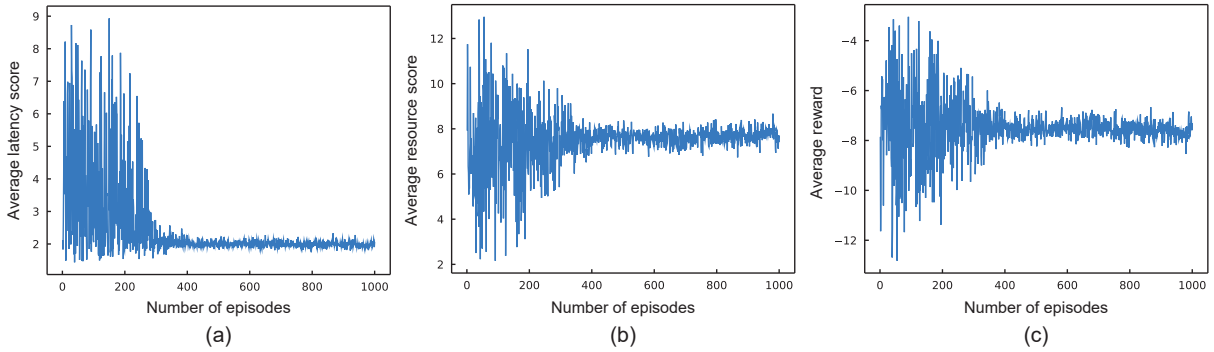

Fig. 5 Results of average service latency, average resource score, and average reward per time step.

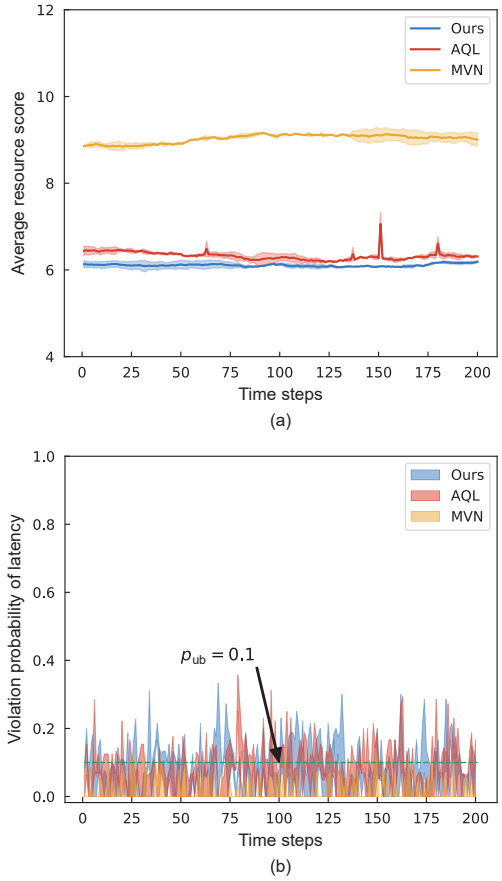

Fig. 6 Resource score of each method and short-term violation probability of latency requirement with $\eta_a = 1.0$ (p_{ub} is the probability that the end-to-end task latency exceeds d_{max}).

medical services.

Figure 7 presents the results, with Figs. 7a and 7b depicting the outcome for $\eta_a = 1.5$ and Figs. 7c and 7d for $\eta_a = 0.5$. In order to ensure reliability, we conduct five rounds of testing for each controller, and the pale region in the figure represents the standard error band. Clearly, our proposed model consistently satisfies the latency requirements of medical services while consuming fewer resources, regardless of whether the arrival rate increases or decreases. However, we also observe that in Fig. 7a, the other two methods consume more resources, particularly the AQL controller, even with a slight increase in arrival rate. In Fig. 7d, we notice that the AQL controller exhibits significantly larger fluctuations compared to the other two controllers.

In conclusion, based on all experimental results, our proposed model proves to be effective and efficient once training is completed, even in the face of changing environments. The MVN controller satisfies

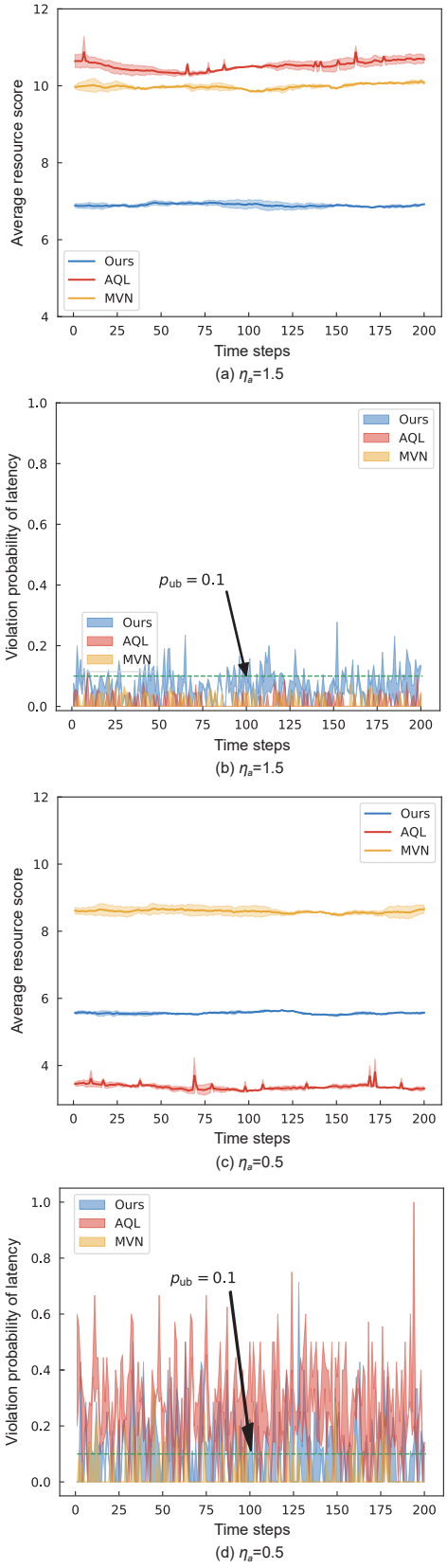

Fig. 7 Average resource score of each method and short-term violation probability of latency requirement with increased/decreased arrival rate.

latency requirements but consumes excessive resources, while the AQL controller struggles to adapt to environmental variations and may become unstable.

6 Conclusion

In the context of an IoT-cloud based healthcare system that offers real-time and computation-intensive medical services, a significant challenge arises in achieving the dual objectives of meeting medical service latency requirements while minimizing resource utilization. In this study, we address this challenge by formulating the problem as a two-stage tandem queue and employing a DRL algorithm to learn a policy. This learned policy enables automatic adjustment of the transmission rate and computation speed, thereby ensuring the fulfillment of latency requirements while conserving resources.

To evaluate the effectiveness and efficiency of our proposed model, we conduct extensive comparison experiments. These experiments aim to validate the model's ability to appropriately allocate resources and demonstrate its superiority over alternative approaches. Furthermore, as part of our future work, we intend to explore the feasibility of applying our model to a cloud-fog based healthcare system. This investigation will enable us to assess the adaptability and performance of our proposed model in a different system architecture. A more stringent medical cloud configuration encompasses factors such as strong real-time computing requests, on-demand computing for workload variations, energy-efficient computing, and others. We intend to further explore these aspects in our future research endeavors.

Acknowledgment

This work was supported by the National Key Research and Development Program (No. 2022YFB3104600), the Medico-Engineering Cooperation Funds from University of Electronic Science and Technology of China (Nos. ZYGX2021YGLH213 and ZYGX2022YGRH016), the Interdisciplinary Crossing and Integration of Medicine and Engineering for Talent Training Fund, West China Hospital, Sichuan University (No. HXDZ22010), the Yuxi Normal University (No. 202105AG070010), the Municipal Government of Quzhou (Nos. 2022D018, 2022D029, 2023D007, 2023D033, 2023D034, and 2023D035), the Quzhou Municipal Science and Technology Bureau Key Technology Research and Development Project (No. 2022K50), as well as the Zhejiang Provincial Natural Science Foundation of China (No. LGF22G010009).

References

[1] Marketsandmarkets, IoT in healthcare market by component, application, end user, and region global forecast to 2025, https://www.marketsandmarkets.com/Market-Reports/iot-healthcaremarket-160082804.html, 2020.

[2] S. Dananjayan and G. M. Raj, 5G in healthcare: How fast will be the transformation? *Ir. J. Med. Sci. 1971*, vol. 190, no. 2, pp. 497–501, 2021.

[3] X. Wang and Z. Jin, An overview of mobile cloud computing for pervasive healthcare, *IEEE Access*, vol. 7, pp. 66774–66791, 2019.

[4] R. Zgheib, A. De Nicola, M. L. Villani, E. Conchon, and R. Bastide, A flexible architecture for cognitive sensing of activities in ambient assisted living, in *Proc. IEEE 26th Int. Conf. Enabling Technologies*: *Infrastructure for Collaborative Enterprises* (*WETICE*), Poznan, Poland, 2017, pp. 284–289.

[5] M. Tariq, H. Majeed, M. O. Beg, F. A. Khan, and A. Derhab, Accurate detection of sitting posture activities in a secure IoT based assisted living environment, *Future Gener. Comput. Syst.*, vol. 92, pp. 745–757, 2019.

[6] A. Rashed, A. Ibrahim, A. Adel, B. Mourad, A. Hatem, M. Magdy, N. Elgaml, and A. Khattab, Integrated IoT medical platform for remote healthcare and assisted living, in *Proc. Japan-Africa Conf. Electronics, Communications and Computers* (*JAC-ECC*), Alexandria, Egypt, 2017, pp. 160–163.

[7] G. Marques and R. Pitarma, An indoor monitoring system for ambient assisted living based on Internet of Things architecture, *Int. J. Environ. Res. Public Health*, vol. 13, no. 11, p. 1152, 2016.

[8] C. Ma, X. Dai, J. Zhu, N. Liu, H. Sun, and M. Liu, DrivingSense: dangerous driving behavior identification based on smartphone autocalibration, *Mob. Inf. Syst.*, vol. 2017, p. 9075653, 2017.

[9] Y. Liu, L. Zhang, Y. Yang, L. Zhou, L. Ren, F. Wang, R. Liu, Z. Pang, and M. J. Deen, A novel cloud-based framework for the elderly healthcare services using digital twin, *IEEE Access*, vol. 7, pp. 49088–49101, 2019.

[10] H. Mora, F. J. Mora Gimeno, M. T. Signes-Pont, and B. Volckaert, Multilayer architecture model for mobile cloud computing paradigm, *Complexity*, vol. 2019, p. 3951495, 2019.

[11] S. Shukla, M. F. Hassan, D. C. Tran, R. Akbar, I. V. Paputungan, and M. K. Khan, Improving latency in Internet-of-Things and cloud computing for real-time data transmission: A systematic literature review (SLR), *Clust. Comput.*, vol. 26, no. 5, pp. 2657–2680, 2023.

[12] R. Mahmud, F. L. Koch, and R. Buyya, Cloud-fog interoperability in IoT-enabled healthcare solutions, in *Proc. 19th Int. Conf. Distributed Computing and Networking*, Varanasi, India, 2018, pp. 1–10.

[13] C. S. Nandyala and H.-K. Kim, From cloud to fog and IoT-based real-time U-healthcare monitoring for smart homes and hospitals, *Int. J. Smart Home*, vol. 10, no. 2, pp. 187–196, 2016.

[14] K. S. Awaisi, S. Hussain, M. Ahmed, A. Ali Khan, and G.

Ahmed, Leveraging IoT and fog computing in healthcare systems, *IEEE Internet Things Mag.*, vol. 3, no. 2, pp. 52–56, 2020.
[15] S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, and R. Buyya, HealthFog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments, *Future Gener. Comput. Syst.*, vol. 104, pp. 187–200, 2020.
[16] J. Ren, J. Li, H. Liu, and T. Qin, Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT, *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 760–776, 2022.
[17] Y. Wang, X. Tao, Y. T. Hou, and P. Zhang, Effective capacity-based resource allocation in mobile edge computing with two-stage tandem queues, *IEEE Trans. Commun.*, vol. 67, no. 9, pp. 6221–6233, 2019.
[18] C. Ma, J. Zhu, M. Liu, H. Zhao, N. Liu, and X. Zou, Parking edge computing: Parked-vehicle-assisted task offloading for urban VANETs, *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9344–9358, 2021.
[19] M. Yang, N. Liu, L. Zuo, Y. Feng, M. Liu, H. Gong, and M. Liu, Dynamic charging scheme problem with actor-critic reinforcement learning, *IEEE Internet Things J.*, vol. 8, no. 1, pp. 370–380, 2021.
[20] S. P. Erdeniz, I. Maglogiannis, A. Menychtas, A. Felfernig, and T. N. T. Tran, Recommender systems for iot enabled m-health applications, *in Proc. IFIP International Conference on Artificial Intelligence Applications and Innovations*, Cham, Switzerland: Springer, 2018: 227–237.
[21] B. Xu, L. Xu, H. Cai, L. Jiang, Y. Luo, and Y. Gu, The design of an m-health monitoring system based on a cloud computing platform, *Enterp. Inf. Syst.*, vol. 11, no. 1, pp. 17–36, 2017.
[22] A. Kelati, I. B. Dhaou, and H. Tenhunen, Biosignal monitoring platform using wearable IoT, in *Proceedings of the 22st Conference of Open Innovations Association FRUCT*, Petrozavodsk, Russia, 2018, pp. 9–13.
[23] H. Xu, L. Zuo, F. Sun, M. Yang, and N. Liu, Low-latency patient monitoring service for cloud computing based healthcare system by applying reinforcement learning, in *Proc. IEEE 8th Int. Conf. Computer and Communications* (*ICCC*), Chengdu, China, 2022, pp. 1373–1377.
[24] M. Raeis, A. Tizghadam, and A. Leon-Garcia, Queue-learning: A reinforcement learning approach for providing quality of service, *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 1, pp. 461–468, 2021.
[25] A. M. Law, *Simulation Modeling & Analysis*, 5th ed. New York, NY, USA: McGraw-Hill, 2015.
[26] M. Raeis, A. Tizghadam, and A. Leon-Garcia, Reinforcement learning-based admission control in delay-sensitive service systems, in *Proc. GLOBECOM 2020 - 2020 IEEE Global Communications Conf.*, Taipei, China, 2020, pp. 1–6.
[27] P. R. Montague Reinforcement learning: An introduction, by Sutton, RS, and Barto, AG, *Trends Cogn. Sci.*, vol. 3, no. 9, p. 360, 1999.
[28] C. J. Watkins and P. Dayan, Q-learning, *Machine learning*, vol. 8, nos. 3&4, pp. 279–292, 1992.
[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[30] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in *Proc. of 12th Int Conf. on Neural Information Processing Systems*, Denver, CO, USA, 2000, pp. 1057–1063.
[31] N. Metropolis and S. Ulam, The Monte Carlo method, *J. Am. Stat. Assoc.*, vol. 44, no. 247, pp. 335–341, 1949.
[32] T. Xie, X. Cheng, X. Wang, M. Liu, J. Deng, T. Zhou, and M. Liu, Cut-thumbnail: A novel data augmentation for convolutional neural network, in *Proc. 29th ACM Int. Conf. Multimedia. Virtual Event China*, 2021, pp. 1627–1635.
[33] L. Wu, M. Liu, X.-M. Wang, G.-H. Chen, and H.-G. Gong, Mobile distribution-aware data dissemination for vehicular ad hoc networks, *J. Softw.*, vol. 22, no. 7, pp. 1580–1596, 2011.

Hongfei Du received the BEng and MEng degrees from National University of Defense Technology and University of Electronic Science and Technology of China (UESTC), China in 2012 and 2018, respectively. He is the founder of Chengdu Athena Technology Co. Ltd. and Sichuan Gangtai Construction Engineering Co. Ltd. His work and research have focused on video communication software engineering computer science and technology for more than 20 years. He is currently a PhD candidate at Future Media Research Center, School of Computer Science and Engineering, UESTC. As an electronic technology senior engineer, his current research interests include artificial intelligence, big data, and multi-agent confrontation and collaboration. He is the author of 4 papers and more than 13 patents. Now he is in charge of the sub-project of the Key Research and Development Project of the Ministry of Science and Technology.

Ming Liu received the PhD degree from Nanjing University, China in 2006, and received Excellent Doctoral Dissertation of Nanjing University Outstanding Doctoral Thesis in Jiangsu Province. He is currently a professor at School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include deep learning, parallel and distributed computing, wireless sensing and networks, big data, medical image processing, etc. He has published over 100 papers in major international journals and conference proceedings.

Deying Li is a professor at Renmin University of China. She received the BS and MS degrees in mathematics from Central China Normal University, China in 1985 and 1988, respectively, and the PhD degree in computer science from City University of Hong Kong, China in 2004. Her research interests include wireless networks, ad hoc and sensor networks, mobile computing, distributed network system, social networks, algorithm design, etc.

Wenzhong Li receives the BEng and PhD degrees from Nanjing University, China in 2002 and 2007, respectively, both in computer science. He was an Alexander von Humboldt Scholar Fellow at University of Goettingen, Germany. He is now a professor at Department of Computer Science and Technology, Nanjing University, China. His research interests include distributed computing, big data mining, and social networks. He served as program co-chair of MobiArch 2013 and registration chair of ICNP 2013. He was the TPC member of several international conferences. He is the reviewer of many journals, the principle investigator of four fundings from NSFC, and the co-principle investigator of a China-Europe international research staff exchange program. He is a member of IEEE, ACM, and China Computer Federation (CCF). He was also the winners of the Best Paper Award of ICC 2009 and APNet 2018. He was featured on Elsevier's Most Cited Chinese Researchers in 2022. His research interests include AI-empowered networking systems and applications, edge computing/mobile cloud computing, big data processing and mining, social networks analysis, etc.

Nianbo Liu received the PhD degree from University of Electronic Science and Technology of China in 2011. He was a research assistant at Hong Kong Polytechnic University, China in 2010, and now is a research associate and master's supervisor at School of Computer Science and Engineering, University of Electronic Science and Technology of China. His research interests include computer networks, algorithm design, artificial intelligence, etc.

Lifeng Xu received the bachelor degree of medicine from Wenzhou Medical University, China in 1995. He serves as the Discipline Inspection Secretary and the Chief Technician of Clinical Laboratory Diagnostics at People's Hospital of Quzhou City Wenzhou Medical University, China. In addition to his clinical work, he is actively involved in academia as a master's supervisor and adjunct professor at both Wenzhou Medical University and Zhejiang Chinese Medical University, China. His contributions to the field extend to his role as the committee member of the Clinical Transfusion Branch of Zhejiang Medical Association, vice chairman of the Quzhou City Anti-Cancer Association, Nan Kong Elite in medical and health sciences in Quzhou City, and the director of the Key Laboratory of Medical Artificial Intelligence Diagnosis and Prognosis Technology Research and Development in Quzhou City, China. His research interests include AI-assisted medical network, AI-assisted disease diagnosis, AI-assisted logistics management, etc.