A deep reinforcement learning-based wireless body area network offloading optimization strategy for healthcare services

A Deep Reinforcement Learning-Based Wireless Body Area Network Offloading Optimization Strategy for Healthcare Services

Yingqun Chen[1], Shaodong Han[1], Guihong Chen[1,2*], Jiao Yin[3], Kate Nana Wang[4] and Jinli Cao[5]

[1]School of Cyber Security, Guangdong Polytechnic Normal University, Guangzhou, 510000, China.

[2]School of Automation Science and Engineering, South China University of Technology, Guangzhou, 510641, China.

[3]Institute for Sustainable Industries and Liveable Cities, Victoria University, Melbourne, 3011, Australia.

[4]School of Health and Biomedical Sciences, RMIT University, Melbourne, 3082, Australia.

[5]Department of Computer Science and Information Technology, La Trobe University, Melbourne, 3086, Australia.

*Corresponding author(s). E-mail(s): chenguihong@gpnu.edu.cn;

Abstract

Wireless body area network (WBAN) is widely adopted in healthcare services, providing remote real-time and continuous healthcare monitoring. With the massive increase of detective sensor data, WBAN is largely restricted by limited storage and computation capacity, resulting in severely decreased efficiency and reliability. Mobile edge computing (MEC) technique can be combined with WBAN to resolve this issue. This paper studies the joint optimization problem of computational offloading and resource allocation (JCORA) in MEC for healthcare service scenarios. We formulate JCORA as a Markov decision process (MDP) and propose a deep deterministic policy gradient-based WBAN offloading strategy (DDPG-WOS) to optimize time delay and energy consumption in interfered transmission channels. This scheme employs MEC to mitigate the computation pressure on a single WBAN and increase the transmission ability. Further, DDPG-WOS optimizes the offloading strategy-making process by considering the channel condition, transmission quality, computation ability and energy consumption. Simulation results verify the effectiveness of the proposed optimization schema in reducing energy consumption and computation latency and increasing the utility of WBAN compared to two competitive solutions.

Keywords: Wireless body area networks, deep reinforcement learning, offloading policy, mobile edge computing

1 Introduction

Real-time healthcare monitoring is necessary and crucial for patients and physicians to achieve the best possible healthcare. With the continuous development of the internet of things (IoT), the wireless body area network (WBAN) has also received close attention from all walks of life [1, 2]. It has become a feasible solution for human health monitoring [3–5].

Generally, a WBAN is composed of several wearable or implanted physiological sensors to sense relevant physiological parameters of the human body, such as body temperature, blood pressure, electrocardiogram (ECG) signals [6], electroencephalogram (EEG) signals, etc., as demonstrated in Fig. 1. In addition, one central sensor, in the form of a smartwatch, belt, etc., acts as a coordinator to collect data parameters from other sensor nodes and schedule the transmission and computation in WBAN [7]. The two kinds of sensors in WBANs have limited energy and capacity for intensive data transmission and computation. These constraints may cause mission completion latency, unreliability, and even failure, which is unacceptable. Because the missions (e.g., physiological data transmission, health status evaluation, etc.) in WBAN are life-related. Thus, it is crucial to design novel mechanics to improve the transmission and computation abilities of WBAN under the constraints of energy and capacity to advance the service quality [8].

The mobile edge computing (MEC) technique places distributed computing capabilities at the edge of mobile networks, using wireless access networks to provide users with information technology services and computing capabilities [9, 10]. Typical service scenarios include computation-intensive applications, telematics, and the IoT [11, 12]. Based on the MEC technique, WBAN can offload computing tasks to edge servers for computing processing, making task execution faster and more efficient while reducing the computing energy consumption of WBAN [13, 14]. As shown in Fig. 1, when offloading tasks from WBAN to MEC, there are three processes, i.e., local computing, transmission process and MEC computing. The primary concern for the MEC-based WBAN applications is how to formulate a WBAN offloading strategy to fully use the network resources under various network conditions and constraints.

To date, only a few works have focused on MEC-based WBAN applications. Most current researches focus on optimizing latency and energy consumption,

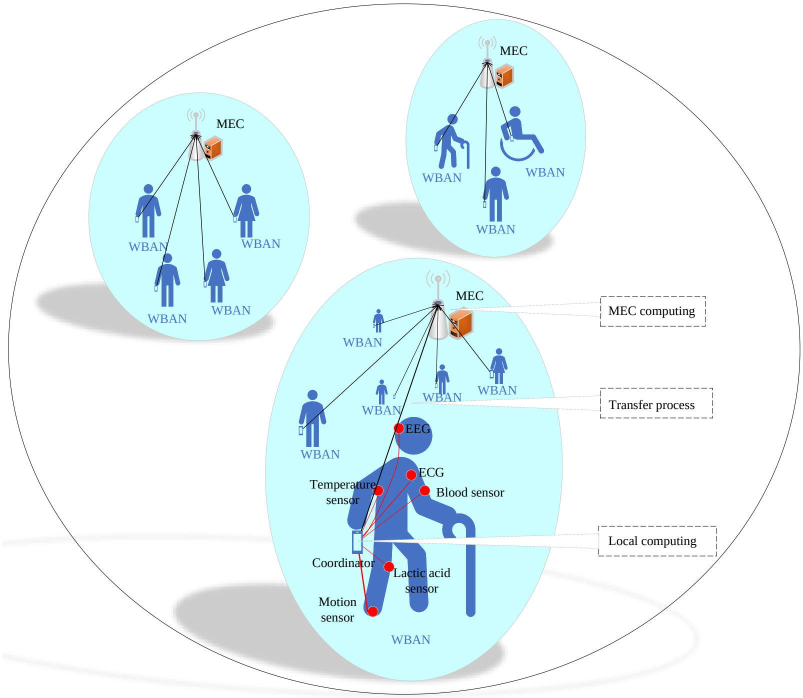


Fig. 1: Illustration of MEC enhanced WBAN networks

which is especially important for WBAN applications. For this type of application, long-term system performance is also critical. Chen, Y. et al. proposed a deep reinforcement learning algorithm to minimize the long-term average delay [15]. However, they did not consider the interference between WBAN and MEC, which is unavoidable in real WBAN application scenarios.

In these scenarios, interference brings enormous challenges. It will lead to an increase in the energy consumption of user equipment, an increase in the delay of computing tasks, and even a severe threat to the security of wireless communication in MEC networks [16]. Nevertheless, most previous works do not consider interference when studying data transmission [10, 13, 17]. Therefore, to address this gap, it is crucial to develop an effective WBAN offloading strategy in the presence of interference and to study the joint optimization problem of computational offloading and resource allocation (JCORA). The WBAN offloading strategy needs to consider the offloading object, offloading volume, channel conditions, transmission power, and other factors which increase the difficulty and complexity of the JCORA. In addition, WBAN channels are difficult to estimate in real-time, affecting the offloading policy's accuracy. The strategy formulation process can be defined as a Markov decision process (MDP) to resolve this issue. Reinforcement learning (RL) can be used to optimize the MDP, in which the channel conditions are not required [15, 18, 19]. In the process of RL, the coordinators of WBAN continuously interact with MECs to develop long-term solutions for JCORA according to the system state. Without considering the dynamics and continuity of WBAN task generation, Alnoman, Ali et al. proposed a reinforcement learning-based computational offloading algorithm to solve the resource allocation problem with low latency and energy consumption. However, it is challenging to apply the algorithm to more complex environments [13]. The authors in [20] proposed an RL-based computation offloading scheme to minimize latency and energy consumption. This scheme considers key performance indicators, including throughput, latency, resource utilization, and energy consumption, to maximize the long-term benefits of the system resource allocation algorithm. In [21], the service migration problem is modeled as a Markov model, and an efficient algorithm is proposed to find the minimum long-term cost. However, since the user equipment is extremely close to the MEC, the author only considers the case of safe offloading and does not consider the interference that brings more significant delay and energy consumption to the WBAN communication. Besides, the RL technology in solving dynamic JCORA problems usually suffers from exponentially growing search space and heavy computational burden, especially in large-scale scenarios. In the dynamic case, the state and action space of WBAN is high-dimensional continuous. Thus general RL is unsuitable for solving the JCORA problem of WBAN.

Deep reinforcement learning (DRL) can solve the high-dimensional problem of RL [22]. As one of the DRL methods, deep deterministic policy gradient (DDPG) learning is preferable in solving the JCORA issue[23]. Particularly, in the WBAN offloading scenario with signal interference, DDPG can effectively solve the dimensional disaster of system states and actions. In addition, it is more practical to consider the values of offloading actions, such as offloading rate, transmission power, etc., as continuous values. And DDPG can accurately select actions in the continuous action space. Thus, in this paper, we design a DDPG-based WBAN offloading strategy (DDPG-WOS) to solve the dynamic JCORA problem. The findings of this paper contribute to the current literature in three ways.

1. We investigate the JCORA problem arising from WBAN combined with MEC. The WBAN coordinators act as the offloading strategy learning agents. Unlike existing works, DDPG-WOS allows partial task offloading, where a portion of arbitrary tasks can be offloaded to the corresponding MEC server for processing.
2. We propose an interference model to make independent decisions based on local JCORA information to optimize delay and energy consumption while solving the interference problem for the reliable transmission of real data.
3. Experimental results show that the DDPG-WOS is more effective in reducing task latency, saving energy and improving system utility compared with the two competing schemes.

The rest of the paper is organized as follows. Section 2 reviews related works on optimization-based techniques and reinforcement learning-based techniques. Section 3 presents the network model, computational model and problem formulation of the task offloading problem. Section 4 describes the proposed DDPG-WOS scheme in detail, followed by the experiments and analysis in Section 5. Finally, Section 6 summarizes the paper.

2 Related works

With the combination of WBAN and MEC, the network environment becomes more and more complex [20]. How to properly manage various resources and optimize network performance in the combination of WBAN and MEC has become a hot research topic in recent years [8, 24, 25]. Multiple techniques are already available to solve the JCORA problem of MEC. We divide them into two research directions, one is based on optimization, and the other is based on RL algorithms.

2.1 Optimization-based techniques

Edge computing servers can offer numerous powerful computing resources to IoT devices. IoT devices can offload intensive computing tasks to edge servers, saving their computing resources and reducing energy consumption [9]. Liu, J. tried to seek an exchange between energy consumption and latency to satisfy user needs for varied IoT applications [23]. Lyu, X. et al. devised a heuristic algorithm to solve the JCORA problem in MEC networks. However, obtaining a satisfactory solution usually takes a long time[26].

Several studies have used scientific theories to resolve the JCORA. For instance, Zheng, J. et al. studied the multi-user computation offloading downside for mobile cloud computing in an exceedingly dynamic surrounding [27]. They developed the offloading invocation method for mobile users in dynamic environments as a stochastic game, trying to make it resemble a minimum Nash equilibrium (NE). Moreover, a multi-intelligent random learning algorithm program was planned to ensure the convergence speed of NE. To address the problem of mutually constraining strategies in the strategy space of the potential game model, Yuan, Xiaoming et al. proposed a two-stage optimization method to reduce the process quality and improve the practicality of the algorithmic program [17]. However, as the number of devices increases, these methods typically suffer from exponential growth in action and state spaces and a heavy computational burden.

To solve the dynamic JCORA downside, Nath, S. et al. designed an accommodating offloading calling algorithmic program that supported Lyapunov improvement to attenuate energy consumption [28]. However, Lyapunov improvement needs prior information concerning the surroundings statistics, which is not applicable in real dynamic MEC systems.

2.2 Reinforcement learning-based techniques

RL is a mathematical framework for experience-driven autonomous learning through interactions, making machines intelligent, and minimizing manual input [29]. RL is promising to improve computational performance. For example, the binary offloading-based MEC framework, as designed in [30], applies Q-learning to select the MEC server working modes for less energy consumption following the quality of service requirements. Due to scalability issues, applications of conventional RL methods mainly focus on problems with low-dimensional states and action spaces. In addition, their performance is heavily dependent on the quality of handcrafted features. Thus, they are unsuitable for large, complex problems, such as emergency control for large-scale systems.

Dynamic offloading decisions are complicated because the influencing factors are multidimensional and time-varying [31]. In recent years, more and more researches have been devoted to deep reinforcement learning-based techniques, including value-based and policy-based approaches. DRL has emerged as a class of RL schemes that uses deep learning to enable RL to scale to decision-making problems with high-dimensional state and action spaces [18, 24, 30, 32].

2.2.1 Value-based deep reinforcement learning

Huang, L. et al. proposed a deep Q-network (DQN)-based online offloading framework that adopts deep neural networks to learn binary offloading decisions from experience [33]. Wu, H. et al. proposed a DQN-based joint computational offloading and task migration optimization algorithm, where the total latency and energy consumption cost are minimized [28]. Xu, X. et al. proposed a novel heuristic algorithm to solve the task offloading problem for optimal value functions to obtain the minimized total delay [34]. Lu, H. et al. proposed a DQN-based intelligent resource allocation scheme that adaptively allocates computational and network resources, reduces mean time to service and balances the use of resources across different MEC environments [35]. However, the DQN is trained with Q-learning, which overestimates the true value. Furthermore, due to the dynamic nature of WBAN, the value-based approach cannot handle the continuous action space.

2.2.2 Policy-based deep reinforcement learning

Ale, L. et al. proposed a DDPG algorithm to solve multi-objective optimization problems, maximizing the number of tasks processed before expiration and minimizing energy costs and service delays [36]. More importantly, DDPG can efficiently handle the restricted distributed-continuous hybrid action space. The complex computation offloading problem can be solved based on the network's real-time state and the task's attributes. As a policy-based approach, DDPG is adopted by Li, Y. et al. to optimize computation offloading decisions [37]. Chen, Xutao, et al. proposed an optimal offloading decision based on the DDPG algorithm to solve the high energy consumption and total latency cost of task processing [38]. Hu, Han, et al. investigated the problem of offloading computational tasks in dynamic environments and proposed the DDPG algorithm to handle continuous actions to minimize the average long-term service cost in terms of power consumption and buffering delay [39]. Zhang, Lingling, et al. proposed a DDPG-based algorithm to perform computational offloading and resource allocation in each fog access point to achieve lower task execution latency and energy consumption[40].

Due to the characteristics of WBAN, the decentralized JCORA mechanism based on DDPG shows a significant advantage in handling the MEC-based WBAN offloading problem.

3 System model

3.1 Network model

Fig. 1 shows a general MEC-enabled WBAN network containing N WBAN devices and M MEC servers. Let $WBAN_n$, $n \in \{1, 2, ..., N\}$, presents the n-th WBAN device being managed, and MEC_m, $m \in \{1, 2, ..., M\}$, denotes the m-th MEC server available for use.

WBAN devices offload some data to MEC equipment for calculation. However, during the offloading process, communication interference exists, including superimposed interference from WBAN devices covered by the same MEC and co-tier interference from WBANs covered by other MECs. We assume that the WBAN utilizes orthogonal frequency division multiple access (OFDMA) for offloading tasks, with each WBAN utilizing a distinct channel for transmissions. When offloading data from $WBAN_n$ to MEC_m server, the signal-to-interference-to-noise ratio (SINR) of each time slot k, $k \in \{$ 1, 2, 3, ..., $T\}$, in the system can be calculated as equation (1) according to [41].

$$SINR_{m,n}^{(k)} = \frac{p_{m,n}^{tr(k)} \mid h_{m,n}^{(k)} \mid^2}{I_{m,n}^{(k)} + J_{m,n}^{(k)} + \sigma^2} \tag{1}$$

where $p_{m,n}^{tr(k)}$ is the transmission power of $WBAN_n$. $h_{m,n}^{(k)} = l_{m,n} d^{-\sigma}$ is the channel gain from $WBAN_n$ to MEC_m, in which d denotes the distance from $WBAN_n$ to MEC_m and σ denotes the route loss factor. $I_{m,n}^{(k)} = \sum_{u=1,u\neq n}^{N} p_{m,u}^{tr(k)} \mid h_{m,n}^{(k)} \mid^2$ is the superimposed interference from other $WBAN$s covered by MEC_m. $J_{m,n}^{(k)} = \sum_{m1=1,m1\neq m}^{M} \sum_{u=1,u\neq n}^{N} p_{m1,u}^{tr(k)} \mid h_{m,n}^{(k)} \mid^2$ is co-tier interference from WBANs covered by other MECs. σ^2 is the white Gaussian noise.

According to Shannon algorithm [42], the maximum transmission rate from $WBAN_n$ to MEC_m is defined as (2).

$$R_{m,n}^{(k)} = B_{m,n}^{(k)} \log_2 (1 + SINR_{m,n}^{(k)}) \tag{2}$$

where $B_{m,n}^{(k)}$ is the wireless bandwidth.

3.2 Task definition

The connections of WBANs are dynamic, and their computation tasks are generated randomly. At each time slot k, the computation task generated by $WBAN_n$ is represented as $Task_n^{(k)} = \left\{D_n^{(k)}, T_n^{(k)}, E_n^{(k)}\right\}$, where $D_n^{(k)}$ represents the data volume that needs to be transmitted and computed, $T_n^{(k)}$ and $E_n^{(k)}$ are the maximum time and energy consumption allowed to handle the task, respectively. We denote the offload rate of task $Task_n^{(k)}$ as $o_{m,n}^{(k)}$, where $o_{m,n}^{(k)} \in [0, 1]$. If $o_{m,n}^{(k)}= 0$, the task is performed locally only, and if $o_{m,n}^{(k)}= 1$, the task is totally offloaded to the MEC server for computing. When $0 < o_{m,n}^{(k)} < 1$, $(1 - o_{m,n}^{(k)})D_n^{(k)}$ is computed by the local $WBAN_n$, and $o_{m,n}^{(k)}D_n^{(k)}$ is offloaded to the MEC_m server for computation.

3.3 Computational model

3.3.1 Local computing model

If $o_{m,n}^{(k)} \neq 1$, the task $(1 - o_{m,n}^{(k)})D_n^{(k)}$ is processed at $WBAN_n$ locally. We define f_n^l as the CPU frequency of $WBAN_n$ for the local computing task, and $c_n^{l(k)}$ represents the number of CPU cycles needed to process one bit of data in $WBAN_n$. Therefore, the local task execution time is denoted as (3).

$$T_n^{l(k)} = \frac{c_n^{l(k)}(1 - o_{m,n}^{(k)})D_n^{(k)}}{f_n^l} \tag{3}$$

According to the literature [42], the local energy consumption can be represented as (4).

$$E_n^{l(k)} = p_n^{l(k)}T_n^{l(k)} \tag{4}$$

where $p_n^{l(k)}$ is the local execution power of $WBAN_n$.

3.3.2 MEC offloading model

If $o_{m,n}^{(k)} \neq 0$, it represents that MEC_m is utilized to complete task $o_{m,n}^{(k)}D_n^{(k)}$. The offloading model must consider multiple edge servers' coexisting situation and the changes in computation resources. The offloading computing process consists of three components, i.e., task transmission, offloaded task computation, and result return.

As the data volume transmitted from MEC to WBAN is tiny, the corresponding transmission latency and energy consumption can be disregarded. Thus, this work only focuses on the latency and energy consumption of the task-offloading process. The transmission time of the offloading task is denoted as (5).

$$T_{m,n}^{tr(k)} = \frac{o_{m,n}^{(k)}D_n^{(k)}}{R_{m,n}^{(k)}} \tag{5}$$

The corresponding transmission energy consumption for the offloading task is represented as (6).

$$E_{m,n}^{tr(k)} = p_{m,n}^{tr(k)} T_{m,n}^{tr(k)} \tag{6}$$

We define f_m as the CPU frequency of MEC_m. $c_m^{(k)}$ represents the number of CPU cycles needed to process one bit of data in MEC_m server. $\xi^{(k)}$ is a variable factor that can be viewed as the computation resource that MEC prefers to allocate to the offloaded task. Since the MEC server has sufficient power supply capacity, its energy consumption is ignored. Therefore, we only need to focus on the offloaded task computation time on MEC, which can be represented as (7).

$$T_m^{(k)} = \xi^{(k)} \frac{c_m^{(k)} o_{m,n}^{(k)} D_n^{(k)}}{f_m} \tag{7}$$

where higher $\xi^{(k)}$ can result in higher $T_m^{(k)}$, which means fewer CPU cycles the MEC server wants to allocate to the offloaded computation task in a period.

This paper assumes that the MEC server has sufficient computational resources. Once the server receives the offloaded data from WBAN, it can execute the task immediately, so there is no queuing delay.

3.3.3 Total computing model

Since $WBAN_n$ and MEC_m conduct operations concurrently, the total time needed to complete the task $Task_n^{(k)}$ is represented as (8).

$$T_{m,n}^{total(k)} = max\left\{T_n^{l(k)}, T_{m,n}^{tr(k)} + T_m^{(k)}\right\} \tag{8}$$

where $T_{m,n}^{total(k)} \leq T_n^{(k)}$.

The total amount of energy needed to complete the task $Task_n^{(k)}$ is represented as (9).

$$E_{m,n}^{total(k)} = E_n^{l(k)} + E_{m,n}^{tr(k)} \tag{9}$$

where $E_{m,n}^{total(k)} \leq E_n^{(k)}$.

For reference, we summarize the main symbols used in the system model in Table 1.

3.4 Problem formulation

Considering that health monitoring is extremely latency-sensitive and WBAN devices are energy-constrained, we regard the offloading and resource allocation of the MEC system as an optimization problem. Under the maximum permissible latency and computing capacity limitations, the problem is defined as (10).

$$u_n^{(k)} = -\lambda_1^{(k)} T_{m,n}^{total(k)} - \lambda_2^{(k)} E_{m,n}^{total(k)} + \lambda_3^{(k)} SINR_{m,n}^{(k)} + \lambda_4 \tag{10}$$

Table 1: Summary of symbols and notations

Symbols	Notations
N	Number of $WBAN$ devices
M	Number of MEC devices
$Task_n^{(k)}$	Computation task generated by $WBAN_n$ at time slot k
$o_{m,n}^{(k)}$	Offload rate of $Task_n^{(k)}$
$D_n^{(k)}$	Computational data volume of $Task_n^{(k)}$
$B_{m,n}^{(k)}$	Channel bandwidth from $WBAN_n$ to MEC_m
f_n^l	CPU frequency of $WBAN_n$
$p_{m,n}^{tr(k)}$	Transmission power of $WBAN_n$
$SINR_{m,n}^{(k)}$	SINR from $WBAN_n$ to MEC_m
$R_{m,n}^{(k)}$	Maximum transmission rate from $WBAN_n$ to MEC_m

where $\lambda_1^{(k)}$, $\lambda_2^{(k)}$ and $\lambda_3^{(k)}$ are the corresponding weight factors of $T_{m,n}^{total(k)}$, $E_{m,n}^{total(k)}$ and $SINR_{m,n}^{(k)}$, respectively, and λ_4 is a constant term. Various tasks might have varying weight values based on the current system state. For example, if the node battery power is low, it should adjust the energy consumption factor to conserve more energy. The latency factor should be applied for latency-sensitive operations to lessen the delay. In environments with high interference, the weight factor of SINR is used to reduce interference. λ_4 is to ensure that $u_n^{(k)}$ is always positive.

Our objective is to build a practical medical task offloading and resource allocation strategy with the lowest cost for the overall WBAN system. Thus, the optimization issue can be represented as (11).

$$P_1 : max \sum_{n=1}^{N} (-\lambda_1^{(k)} T_{m,n}^{total(k)} - \lambda_2^{(k)} E_{m,n}^{total(k)} + \lambda_3^{(k)} SINR_{m,n}^{(k)} + \lambda_4) \tag{11}$$

$s.t.$
$c_1 : \lambda_1^{(k)} + \lambda_2^{(k)} = 1$ and $\lambda_1^{(k)}, \lambda_2^{(k)} \in [0,1]$,
$c_2 : 0 \le o_{m,n}^{(k)} \le 1$,
$c_3 : T_{m,n}^{WBAN(k)} \le T_n^{(k)}$ and $T_{m,n}^{tr(k)} + T_{m,n}^{MEC(k)} \le T_n^{(k)}$,
$c_4 : E_{m,n}^{WBAN(k)} + E_{m,n}^{tr(k)} \le E_n^{(k)}$.

WBAN delay and energy consumption are controlled by constraint c_1. The constraint c_2 signifies the range of constrained offload rate. The constraint c_3 limits the computation time of $WBAN_n$ and MEC_m. The computation energy consumption of $WBAN_n$ is restricted by constraint c_4.

The solution to problem P1 will be described in detail in the next section.

4 Methodology

In the MEC-based WBAN offloading scenario, problem P1 can be viewed as an MDP. DRL has been proven to have advantages in solving MDP in the literature. This section describes the details of the offloading strategy based on DRL.

4.1 Markov decision process

An MDP consists of four sections, denoted as $MDP = \{S, A, P, R\}$, where S indicates the state space, A indicates the action space, P indicates the state transfer probability and R indicates the reward function. At time slot k, WBAN creates action $a^{(k)}$ according to the current system state $s^{(k)}$. After executing the action, the state changes to $s^{(k+1)}$. Then, a reward $r^{(k)}$ is obtained. The MDP is described in detail as follows.

State space

The state space is defined as (12).

$$s^{(k)} = \left\{ \frac{\xi^{(k)}}{f_m}, B_{m,n}^{(k)}, T_{m,n}^{total(k)}, E_{m,n}^{total(k)}, SINR_{m,n}^{(k)} \right\} \tag{12}$$

in which $\frac{\xi^{(k)}}{f_m}$ and $B_{m,n}^{(k)}$ are allocated by MEC_m. The total computation time $T_{m,n}^{total(k)}$, the total energy consumption $E_{m,n}^{total(k)}$ and the signal transmission quality $SINR_{m,n}^{(k)}$ are observed by the coordinator of $WBAN_n$.

Action space

The action space is defined as (13).

$$a^{(k)} = \left\{ MEC_m, p_{m,n}^{tr(k)}, o_{m,n}^{(k)} \right\} \tag{13}$$

in which MEC_m is denoted as the selected edge node, $p_{m,n}^{tr(k)}$ is denoted as the transmission power and $o_{m,n}^{(k)}$ is denoted as the offloading rate.

State transfer probability

If a stochastic process is an MDP, the future state relies solely on its present state and has no association with previous states. Therefore, the state transfer probability $P(s^{(k+1)} \mid s^{(k)}, a^{(k)})$ indicates the probability distribution of $s^{(k+1)}$ given $s^{(k)}$ and $a^{(k)}$. Since WBAN has no previous knowledge of $P(s^{(k+1)} \mid s^{(k)}, a^{(k)})$, state transfer probability can only be determined by the environment [43].

Reward function

The reward function represents the reward when action $a^{(k)}$ is selected in the current state $s^{(k)}$. Problem P1 aims to save energy consumption, reduce computational latency and increase offloading transmission SINR. Thus, the reward function can be defined as (14).

$$r^{(k)} = \sum_{n=1}^{N} (-\lambda_1^{(k)} T_{m,n}^{total(k)} - \lambda_2^{(k)} E_{m,n}^{total(k)} + \lambda_3^{(k)} SINR_{m,n}^{(k)} + \lambda_4) \tag{14}$$

4.2 Deep reinforcement learning

This section proposes a DDPG-based WBAN offloading strategy (DDPG-WOS) to optimize the above MDP. Specifically, DDPG is utilized to create deterministic actions via a policy network, also known as an actor network, which is trained by a specified policy gradient to approach the optimum actions under all the possible states. Meanwhile, a Q-network called the critic network is used to approximate the value function to assess the produced actions. The critic network is trained by minimizing the loss function. The detail of the DDPG-WOS algorithm is provided below.

Define a policy μ and a value function $Q^{\mu}(s^{(k)}, a^{(k)})$. This function represents the expected reward obtained from $s^{(k)}$ after $a^{(k)}$ is taken. According to the Bellman equation, $Q^{\mu}(s^{(k)}, a^{(k)})$ is defined as (15).

$$Q^{\mu}(s^{(k)}, a^{(k)}) = \mathbb{E}[r^{(k)} + \gamma Q(s^{(k+1)}, \mu s^{(k+1)})] \tag{15}$$

where the policy μ is a mapping from states to actions, (i.e., $a^{(k)} = \mu(s^{(k)})$).

The critic network in DDPG is a DNN approximating the Q-value, $Q(s^{(k)}, a^{(k)} \mid \theta^Q) \approx Q(s^{(k)}, a^{(k)})$, where θ^Q is the critic network's trainable parameter. Compared to the critic network, the actor network is a DNN with the parameter θ^{μ} that approximates the ideal policy μ, i.e., $\mu(s^{(k)} \mid \theta^{\mu}) \approx \mu^*(s^{(k)})$, where $\mu^*(s^{(k)})$ represents the optimal action policy.

To increase the approximation accuracy of the critic network for Q-values, we use Adam's algorithm [44] to update the parameters by minimizing the loss function $L(\theta^Q)$. $L(\theta^Q)$ is defined as (16).

$$L(\theta^Q) = \mathbb{E}[(Q(s^{(k)}, a^{(k)} \mid \theta^Q) - y^{(k)})^2] \tag{16}$$

where $y^{(k)}$ is calculated as (17).

$$y^{(k)} = r^{(k)} + \gamma Q'(s^{(k+1)}, \mu'(s^{(k+1)} \mid \theta^{\mu'}) \mid \theta^{Q'}) \tag{17}$$

At time slot k, $r^{(k)}$ will be obtained after $a^{(k)}$ is taken, and the state will change from $s^{(k)}$ to $s^{(k+1)}$. Then, the list $[s^{(k)}, a^{(k)}, r^{(k)}, s^{(k+1)}]$ will be stored in the replay memory buffer R_{mb}. Simultaneously, V data will be sampled randomly from R_{mb} to train the critic network. Thus, the loss function in (16) can be represented as (18).

$$L(\theta^Q) = \frac{1}{V} \sum_{i=1}^{V} [(Q(s^{(i)}, a^{(i)} \mid \theta^Q) - y^{(i)})^2] \tag{18}$$

The actor network employs the output of the critic network to evaluate the effectiveness of policy μ [44]. Random noise is introduced into the decision mechanism to explore possible strategies. The unique action η with the introduction of random noise is defined as (19).

$$\eta = \mu(s^{(k)} \mid \theta^{\mu}) + \rho^{(k)} \tag{19}$$

where $\rho^{(k)}$ is the Gaussian noise with zero mean. Let $J_{\eta}(\mu)$ denotes the target function of the actor network, which is given by (20).

$$J_{\eta}(\mu) = \mathbb{E}_{p^{\eta}}[Q^{\mu}(s^{(k)}, \mu(s^{(k)} \mid \theta^{\mu}))] \tag{20}$$

where p^{η} is the state probability distribution function based on distinct action strategies η. The optimal action is found by maximizing $J_{\eta}(\mu)$, i.e., $\mu^{*}(s^{(k)}) = argmaxJ_{\eta}(\mu)$. To maximize $J_{\eta}(\mu)$, the parameter Q^{μ} has to be trained by altering the gradient.

To make the output of the critic network more stable, two target networks are copied from the actor network and critic network, respectively. The parameters of the two target networks are updated by equation (21).

$$\begin{aligned} \theta^{Q'} &= \phi\theta^{Q} + (1-\phi)\theta^{Q'} \\ \theta^{\mu'} &= \phi\theta^{\mu} + (1-\phi)\theta^{\mu'} \end{aligned} \tag{21}$$

where ϕ is the rate of change that restricts the goal value.

As seen in Algorithm 1, the system state and the DNN parameters are initialized at the start, as shown in steps 1-3. Then, in the training process, the coordinator of WBAN uses the actor network to choose actions. It observes the state at each time slot k. Following the policy μ, it chooses a MEC_m and offloading rate $o_{m,n}^{(k)}$ to help its computation task, and uses transmission power $p_{m,n}^{tr(k)}$ to offload the task. To better explore, Gaussian noise is added to the action policy. After executing the offloading strategy $a^{(k)}$, an immediate reward $r^{(k)}$ would be obtained. And the system state transfers to $s^{(k+1)}$ that is composed of the observation value $\frac{\xi^{(k+1)}}{f_m}$ and $B_{m,n}^{(k+1)}$, the calculation value $T_{m,n}^{total(k+1)}$ and $E_{m,n}^{total(k+1)}$ and the measurement value $SINR_{m,n}^{(k+1)}$. These data composes a transition tuple $\left\{s^{(k)}, a^{(k)}, r^{(k)}, s^{(k+1)}\right\}$ and would be stored in the experience replay memory R_{mb}, shown as the steps 8-13. On the other side, the critic network works simultaneously with the actor network. $a^{(k)}$ and $s^{(k)}$ would be inputted into the critic network to get the Q-value to guide the coordinator to take optimal actions through maximizing (20) in the actor network. To make the critic network performance more stably, the target actor and critic network are trained with the mini-batch data sampled randomly from R_{mb}. The output of the target critic network $y^{(i)}$ is used to calculate the loss, as shown in (18). By minimizing the loss, the parameters in critic network would be updated. The parameter θ^{μ} in the critic-network would be updated via (20), as shown in steps 14-17. After some time, the parameters of the two target DNN networks are copied from the other two DNN networks, as stated in step 18.

Algorithm 1 DDPG-based WBAN offloading strategy (DDPG-WOS)

Input: $D_n^{(k)}$,$f_n^{(k)}$,$f_m^{(k)}$, $B_{m,n}^{(k)}$, $h_{m,n}^{(k)}$
Output: MEC_m, $o_{m,n}^{(k)}$, $p_{m,n}^{tr(k)}$

Initialize the critic network $Q(s^{(0)}, a^{(0)} \mid \theta^Q)$ and the actor network $\mu(s^{(0)} \mid \theta^\mu)$ with weights θ^Q and θ^μ;
Initialize the target-network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$ and $\theta^{\mu'} \leftarrow \theta^\mu$;
Initialize the replay memory buffer R_{mb};
for episode=1 to $max^{episode}$ **do**
  Initialize a random noise $\rho^{(k)}$ for action exploration;
  Initialize the WBAN system environment and receive the initial observation state $s^{(1)}$;
  for k = 1 to K **do**
    Choose MEC_m, $p_{m,n}^{tr(k)}$ and $o_{m,n}^{(k)}$ according to (19) to formulate $a^{(k)}$;
    Execute $a^{(k)}$ and obtain a reward $r^{(k)}$;
    Observe $\frac{\xi^{(k+1)}}{f_m}$ and $B_{m,n}^{(k+1)}$;
    Calculate $T_{m,n}^{total(k+1)}$, $E_{m,n}^{total(k+1)}$ and measure $SINR_{m,n}^{(k+1)}$;
    Formulate the state $s^{(k+1)}$ via (12);
    Store transition tuple $\left\{s^{(k)}, a^{(k)}, r^{(k)}, s^{(k+1)}\right\}$ in R_{mb} ;
    Randomly sample a mini-batch of V tuple $\left\{s^{(i)}, a^{(i)}, r^{(i)}, s^{(i+1)}\right\}$ from R_{mb};
    Calculate $y^{(i)}$ via (17);
    Update critic by minimizing the loss via (18);
    Update the actor policy using the sampled policy gradient via (20);
    Update the target networks parameters via (21);
  end for
end for

5 Experiments

We conduct a series of simulations to evaluate the performance of the DDPG-WOS scheme. First, the performance of different schemes in terms of task completion latency, energy consumption and system utility is compared in a non-interfered environment. Then, the experiment is conducted in an interfered environment. At last, the influence of different data volumes and bandwidth on the performance of the schemes is analyzed in an interfered scenario.

5.1 Experimental settings

In the simulation, we assume the following setup. We examine a 10 × 10 m square area where the base stations deploy MEC servers. The CPU cycle frequency of the MEC server is f_m=2 G cycles/s. The data volume generated at each time slot is in the range of $D_n^{(k)}$=[100, 300] kB, and the CPU cycle frequency of each WBAN device is set as f_n^l=0.5 G cycles/s.

5.2 Experimental comparison

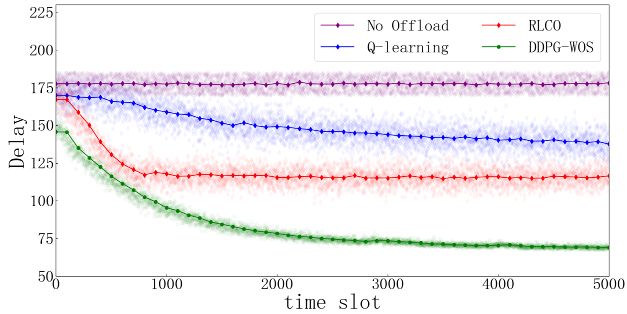
The performance of DDPG-WOS in terms of latency, energy consumption and utility is evaluated in a non-interfered and interfered environment. In addition, the influence of interference is also compared. Traditional Q-learning and the reinforcement learning-based computational offloading (RLCO) algorithm in [20] are adopted as the benchmark schemes.

Firstly, we experiment in a non-interfered environment. In this experiment, the computational data volume is set to be 200 kB per time slot, the channel gain is 0.5, and each channel bandwidth is fixed in the range of [2.5, 4] Mbps. In addition, the performance of the method without offloading is compared as a baseline. Fig.2a shows that the latency based on the DDPG-WOS algorithm can be reduced by 47.44 ms, 69.63 ms and 108.95 ms compared to the RLCO-based, Q-learning-based and no-offloading algorithms, respectively. Regarding energy consumption, it can be seen from Fig.2b that the DDPG-WOS algorithm can reduce 2.31 mJ compared to the RLCO algorithm and can reduce 2.99 mJ and 4.63 mJ compared to the Q-learning and no-offloading algorithms, respectively. Fig.2c shows that the utility based on the DDPG-WOS algorithm is improved by 230 and 330 compared to the RLCO-based and Q-learning-based algorithms.

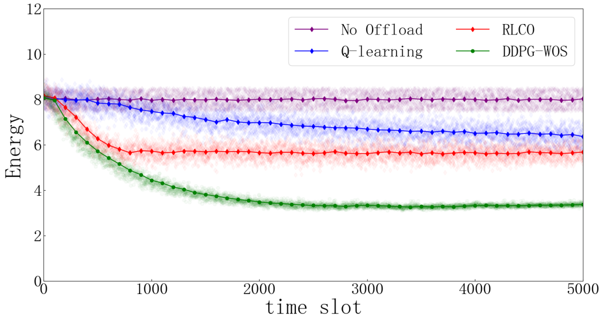
In an interfered environment, the experiment settings are the same as the former experiment. Besides, the interference power is randomly chosen from [30, 40, 50]. Fig. 3 compares the performance of the algorithms without offloading, Q-learning, RLCO, and DDPG-WOS. Regarding latency, Fig. 3a shows that the DDPG-WOS algorithm can decrease 46.79 ms compared to the RLCO algorithm and can decrease 58.85 ms and 118.15 ms compared to the Q-learning and no-offloading algorithms, respectively. Fig. 3b shows that the energy consumption based on the DDPG-WOS algorithm can be reduced by 7.09 mJ compared to the RLCO-based algorithm and can be reduced by 8.25 mJ and 14.31 mJ compared to the Q-learning-based and no-offloading algorithms. Fig. 3c illustrates that the utility based on the DDPG-WOS algorithm is improved by 396 compared to the RLCO-based algorithm and 488 compared to the Q-learning-based algorithm.

From Fig. 2 and Fig. 3, it can be seen that the DDPG-WOS algorithm has better performance no matter with or without interference compared with the other algorithms.

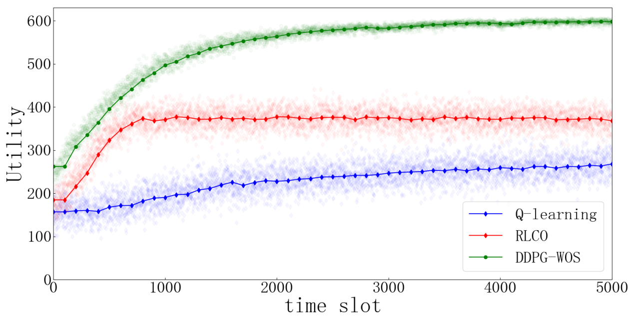
The impact of changing single and multiple parameters on the algorithms is compared in the presence of interference. Firstly, the data volume is set in the range of [100, 300] kB and the bandwidth is fixed at 2.5 Mbps. Then the bandwidth is set in the range of [2.5, 4] Mbps and the data volume is fixed at 200 kB. Finally, the data volume and bandwidth are set in the range of [100, 300] kB and [2.5, 4] Mbps, respectively. In Fig. 4, each value represents the average value of the particular algorithm after convergence. The results show that the DDPG-WOS algorithm outperforms the RLCO and Q-learning algorithms.



(a) Delay

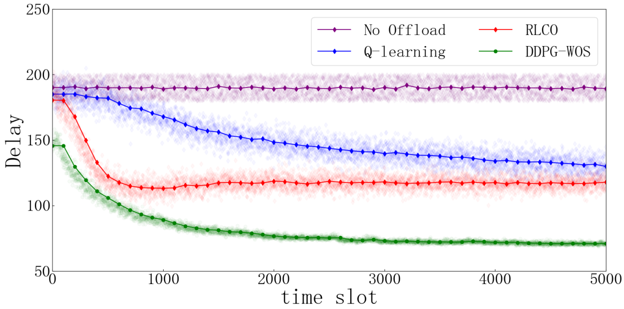


(b) Energy

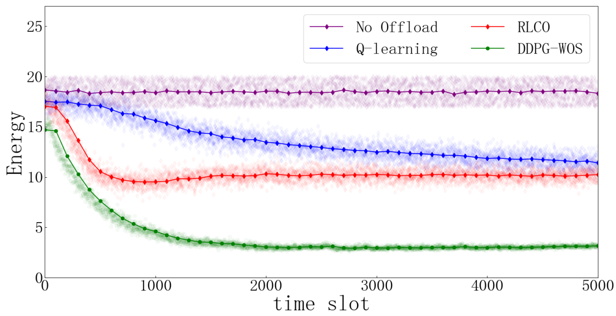


(c) Utility

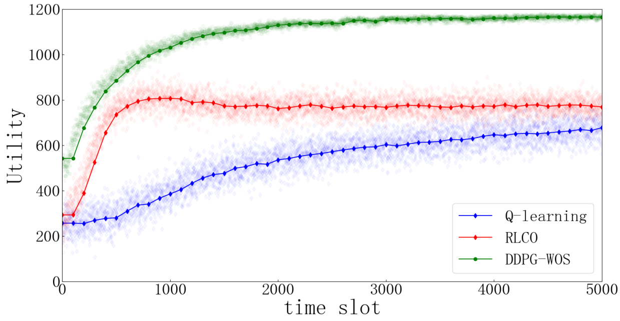
Fig. 2: The performance of the algorithms without channel interference



(a) Delay

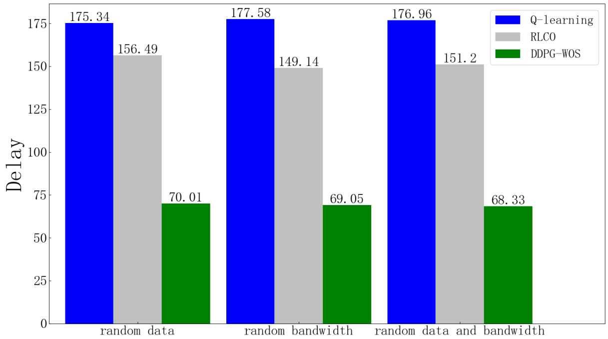


(b) Energy

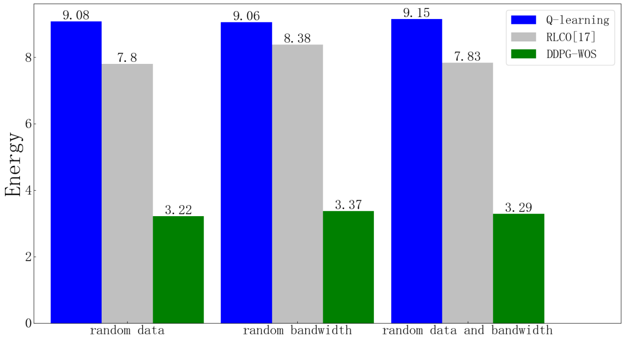


(c) Utility

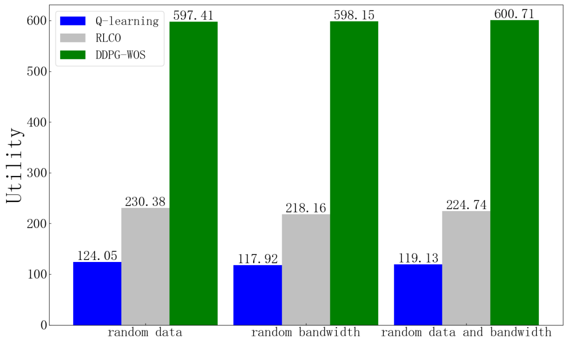
Fig. 3: The performance of the algorithms under channel interference



(a) Delay



(b) Energy



(c) Utility

Fig. 4: The influence of different data volumes and bandwidth on the performance of the algorithms

6 Conclusion

This paper developed a DDPG-based WBAN offloading strategy (DDPG-WOS) to solve the joint optimization problem of computational offloading and resource allocation (JCORA) in MEC for healthcare service scenarios. In this scheme, MEC is adopted to mitigate the computation pressure on a single WBAN and increase the transmission ability. The influence of interference in transmission channels on the performance of the DDPG-WOS algorithm was investigated. Experimental results show that the performance of DDPG-WOS outperforms the benchmark algorithms in energy consumption, compute delay, and utility, with or without interference. Future studies should further investigate the security and privacy problems in WBANs to provide more thorough, patient-centered healthcare services.

Acknowledgement

This work was supported in part by the Science and Technology Program of Guangzhou, China (Grant No. 202102080279).

Declarations

Conflict of interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

[1] Tang, C., Yin, J.: A localization algorithm of weighted maximum likelihood estimation for wireless sensor network. Journal of Information &Computational Science **8**(16), 4293–4300 (2011)

[2] Du, J., Michalska, S., Subramani, S., Wang, H., Zhang, Y.: Neural attention with character embeddings for hay fever detection from twitter. Health information science and systems **7**(1), 1–7 (2019)

[3] Qadri, Y.A., Nauman, A., Zikria, Y.B., Vasilakos, A.V., Kim, S.W.: The future of healthcare internet of things: a survey of emerging technologies. IEEE Communications Surveys & Tutorials **22**(2), 1121–1167 (2020)

[4] Teshome, A.K., Kibret, B., Lai, D.T.: A review of implant communication technology in WBAN: Progress and challenges. IEEE reviews in biomedical engineering **12**, 88–99 (2018)

[5] Hammood, D., Alkhayyat, A.: An overview of the survey/review studies in wireless body area network, 18–23 (2020). IEEE

[6] Yin, J., Cao, J., Siuly, S., Wang, H.: An integrated mci detection framework based on spectral-temporal analysis. International Journal of Automation and Computing **16**(6), 786–799 (2019)

[7] Wang, W., Qin, T., Wang, Y.: Encryption-free data transmission and hand-over in two-tier body area networks. Computer Methods and Programs in Biomedicine **192**, 105411 (2020)

[8] Tang, C., Cheng, Y., Yin, J.: An optimized algorithm of grid calibration in wsn node deployment based on the energy consumption distribution model. JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE **9**(4), 1035–1042 (2012)

[9] Brik, B., Frangoudis, P.A., Ksentini, A.: Service-oriented MEC applications placement in a federated edge cloud architecture. In: ICC 2020-2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020). IEEE

[10] Liao, Y., Han, Y., Yu, Q., Ai, Q., Liu, Q., Leeson, M.S.: Wireless body area network mobility-aware task offloading scheme. IEEE Access **6**, 61366–61376 (2018)

[11] Vimalachandran, P., Liu, H., Lin, Y., Ji, K., Wang, H., Zhang, Y.: Improving accessibility of the australian my health records while preserving privacy and security of the system. Health Information Science and Systems **8**(1), 1–9 (2020)

[12] You, M., Yin, J., Wang, H., Cao, J., Miao, Y.: A minority class boosted framework for adaptive access control decision-making. In: International Conference on Web Information Systems Engineering, pp. 143–157 (2021). Springer

[13] Alnoman, A., Sharma, S.K., Ejaz, W., Anpalagan, A.: Emerging edge computing technologies for distributed IoT systems. IEEE Network **33**(6), 140–147 (2019)

[14] Tawhid, M., Ahad, N., Siuly, S., Wang, K., Wang, H.: Data mining based artificial intelligent technique for identifying abnormalities from brain signal data. In: International Conference on Web Information Systems Engineering, pp. 198–206 (2021). Springer

[15] Chen, Y., Liu, Z., Zhang, Y., Wu, Y., Chen, X., Zhao, L.: Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial internet of things. IEEE Transactions on Industrial Informatics **17**(7), 4925–4934 (2020)

[16] You, M., Yin, J., Wang, H., Cao, J., Wang, K., Miao, Y., Bertino, E.: A

knowledge graph empowered online learning framework for access control decision-making. World Wide Web, 1–22 (2022)

[17] Yuan, X., Tian, H., Wang, H., Su, H., Liu, J., Taherkordi, A.: Edge-enabled WBANs for efficient QOS provisioning healthcare monitoring: A two-stage potential game-based computation offloading strategy. IEEE Access **8**, 92718–92730 (2020)

[18] Nath, S., Wu, J.: Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems. Intelligent and Converged Networks **1**(2), 181–198 (2020)

[19] Xu, Y.-H., Xie, J.-W., Zhang, Y.-G., Hua, M., Zhou, W.: Reinforcement learning (RL)-based energy efficient resource allocation for energy harvesting-powered wireless body area network. Sensors **20**(1), 44 (2019)

[20] Yadav, R., Zhang, W., Elgendy, I.A., Dong, G., Shafiq, M., Laghari, A.A., Prakash, S.: Smart healthcare: Rl-based task offloading scheme for edge-enable sensor networks. IEEE Sensors Journal **21**(22), 24910–24918 (2021)

[21] Heidari, A., Jabraeil Jamali, M.A., Jafari Navimipour, N., Akbarpour, S.: Deep Q-learning technique for offloading offline/online computation in blockchain-enabled green IoT-edge scenarios. Applied Sciences **12**(16), 8232 (2022)

[22] Sarki, R., Ahmed, K., Wang, H., Zhang, Y.: Automated detection of mild and multi-class diabetic eye diseases using deep learning. Health Information Science and Systems **8**(1), 1–9 (2020)

[23] Liu, J., Ahmed, M., Mirza, M.A., Khan, W.U., Xu, D., Li, J., Aziz, A., Han, Z.: RL/DRL meets vehicular task offloading using edge and vehicular cloudlet: A survey. IEEE Internet of Things Journal **9**(11), 8315–8338 (2022)

[24] Yin, J., You, M., Cao, J., Wang, H., Tang, M., Ge, Y.-F.: Data-driven hierarchical neural network modeling for high-pressure feedwater heater group. In: Australasian Database Conference, pp. 225–233 (2020). Springer

[25] Yin, J., Tang, M., Cao, J., You, M., Wang, H., Alazab, M.: Knowledge-driven cybersecurity intelligence: software vulnerability co-exploitation behaviour discovery. IEEE Transactions on Industrial Informatics (2022)

[26] Lyu, X., Tian, H., Sengul, C., Zhang, P.: Multiuser joint task offloading and resource optimization in proximate clouds. IEEE Transactions on Vehicular Technology **66**(4), 3435–3447 (2016)

[27] Zheng, J., Cai, Y., Wu, Y., Shen, X.: Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach. IEEE Transactions on Mobile Computing **18**(4), 771–786 (2018)

[28] Wu, H., Sun, Y., Wolter, K.: Energy-efficient decision making for mobile cloud offloading. IEEE Transactions on Cloud Computing **8**(2), 570–584 (2018)

[29] Zanette, A.: Exponential lower bounds for batch reinforcement learning: Batch rl can be exponentially harder than online rl. In: International Conference on Machine Learning, pp. 12287–12297 (2021). PMLR

[30] Shakarami, A., Ghobaei-Arani, M., Shahidinejad, A.: A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective. Computer Networks **182**, 107496 (2020)

[31] Chen, J., Xing, H., Xiao, Z., Xu, L., Tao, T.: A DRL agent for jointly optimizing computation offloading and resource allocation in MEC. IEEE Internet of Things Journal **8**(24), 17508–17524 (2021)

[32] Pandey, D., Wang, H., Yin, X., Wang, K., Zhang, Y., Shen, J.: Automatic breast lesion segmentation in phase preserved DCE-MRIs. Health Information Science and Systems **10**(1), 1–19 (2022)

[33] Huang, L., Bi, S., Zhang, Y.-J.A.: Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks. IEEE Transactions on Mobile Computing **19**(11), 2581–2593 (2019)

[34] Xu, X., Li, D., Dai, Z., Li, S., Chen, X.: A heuristic offloading method for deep learning edge services in 5G networks. IEEE Access **7**, 67734–67744 (2019)

[35] Lu, H., Gu, C., Luo, F., Ding, W., Liu, X.: Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning. Future Generation Computer Systems **102**, 847–861 (2020)

[36] Ale, L., King, S.A., Zhang, N., Sattar, A.R., Skandaraniyam, J.: D3PG: Dirichlet DDPG for task partitioning and offloading with constrained hybrid action space in mobile edge computing. IEEE Internet of Things Journal (2022)

[37] Li, Y., Qi, F., Wang, Z., Yu, X., Shao, S.: Distributed edge computing offloading algorithm based on deep reinforcement learning. IEEE Access **8**, 85204–85215 (2020)

[38] Chen, X., Ge, H., Liu, L., Li, S., Han, J., Gong, H.: Computing offloading decision based on ddpg algorithm in mobile edge computing. In: 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), pp. 391–399 (2021)

[39] Hu, H., Wu, D., Zhou, F., Jin, S., Hu, R.Q.: Dynamic task offloading in mec-enabled iot networks: A hybrid ddpg-d3qn approach. In: 2021 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2021)

[40] Zhang, L., Jiang, Y., Zheng, F.-C., Bennis, M., You, X.: Computation offloading and resource allocation in f-rans: A federated deep reinforcement learning approach. In: 2022 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 97–102 (2022)

[41] Qiu, Y., Zhang, H., Long, K.: Computation offloading and wireless resource management for healthcare monitoring in fog-computing-based internet of medical things. IEEE Internet of Things Journal **8**(21), 15875–15883 (2021)

[42] Zhang, H., Guo, J., Yang, L., Li, X., Ji, H.: Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC. In: 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 115–120 (2017). IEEE

[43] Yu, Y., Tang, J., Huang, J., Zhang, X., So, D.K.C., Wong, K.-K.: Multi-objective optimization for UAV-assisted wireless powered IoT networks based on extended DDPG algorithm. IEEE Transactions on Communications **69**(9), 6361–6374 (2021)

[44] Ye, H., Li, G.Y., Juang, B.-H.F.: Deep reinforcement learning based resource allocation for V2V communications. IEEE Transactions on Vehicular Technology **68**(4), 3163–3173 (2019)