Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4621440, 12 pages
https://doi.org/10.1155/2022/4621440

Research Article

Joint Radio Map Construction and Dissemination in MEC Networks: A Deep Reinforcement Learning Approach

Xingguang Liu, Li Zhou, Xiaoying Zhang, Xiang Tan, and Jibo Wei

College of Electronic Science and Technology, National University of Defense Technology, Changsha, Hunan 410073, China

Correspondence should be addressed to Li Zhou; zhouli2035@nudt.edu.cn

Received 16 May 2022; Accepted 29 June 2022; Published 19 July 2022

Academic Editor: Xiangjie Kong

With the development of 6G, the rapidly increasing number of smart devices deployed in the Industrial Internet of Things (IIoT) environment has been witnessed. The radio environment is showing a trend of complexity, and spectrum conflicts are becoming increasingly acute. User equipment (UE) can accurately sense and utilize spectrum resources through radio map (RM). However, the construction and dissemination of RM incur a heavy computational burden and large dissemination delay, which limit the real-time sensing of spatial spectrum situations. In this paper, we propose an RM construction and dissemination method based on deep reinforcement learning (DRL) in the context of mobile edge computing (MEC) networks. We formulate the dissemination modes selection and resource allocation problems during RM construction and dissemination as a mixed-integer nonlinear programming problem. Then, we propose an actor-critic-based joint offloading and resource allocation (ACJORA) algorithm for intelligent scheduling of computational offloading and resource allocation. We design a novel weighted loss function for the actor network, which combines the discrete actions for offloading decisions and the continuous actions for resource allocation. And the simulation results show that the proposed algorithm can reduce the cost of dissemination by optimizing the offloading strategies and resources, which is more applicable for real-time RM applications in MEC networks.

1. Introduction

With the development of 6G and the rapid growth of mobile data traffic, new business scenarios are constantly emerging. The Industrial Internet of Things (IIoT) is expected to be a crucial technology changing the manufacturing way [1–3]. IIoT is a variety of acquisition or controllers with sensing and monitoring capabilities. And it integrates mobile communication, intelligent analysis, and other technologies into all aspects of the industrial production process, thereby greatly improving manufacturing efficiency and realizing the intelligence of traditional industries. However, with the explosion of IIoT applications, the radio environment has become increasingly complex, which brings unparalleled challenges such as scarce spectrum resources, intermittent wireless connections, and high propagation delays. Further research is needed to address the above issues.

Radio map (RM) is an important tool for understanding radio environments and analyzing network performance. It incorporates geographic information to describe the radio environment from multiple dimensions such as time, frequency, space, and power [4]. RM can not only effectively acquire the distribution of the radio spectrum resources, but also utilize the multidimensional spectrum data and manage the spectrum resources in a straightforward and flexible way. It has been widely used in cognitive radio [5–7], interference management [8], coverage analysis [9–11], and active resource allocation [12–14]. The spectrum data can be collected by interconnected sensors or smart devices, and the spectrum data needs to be further processed to be constructed as an RM. In the traditional cloud-based network architecture, all spectrum data must be uploaded to a centralized cloud server, constructed as an RM, and disseminated to user equipment (UE). Due to the large size of RM, the traditional dissemination scheme from cloud server to UE consumes more bandwidth and time, which cannot meet the low-latency requirements of IIoT.

In recent years, edge intelligence has integrated edge computing and artificial intelligence (AI) technologies to effectively promote edge-end collaboration [15]. In mobile

edge computing (MEC) systems, various services originally deployed on the central cloud server can be deployed on MEC servers, which fundamentally shortens the data transmission delay [16]. At the same time, AI technology represented by reinforcement learning (RL) can schedule the computation offloading and resource allocation in MEC networks [17–19], which improves the efficiency of edge computing. The authors in [20] explored the joint optimization of computational offloading and resource allocation in dynamic multiuser MEC systems and proposed a Q-learning-based method and a double deep Q-networks- (DDQN-) based method to determine joint strategies for computational offloading and resource allocation. [21] considers both the multiuser computation offloading and edge server deployment in an unmanned aerial vehicle- (UAV-) enabled MEC network. The authors proposed two learning algorithms to minimize the system-wide computation cost under a dynamic environment. To solve the joint optimization of computing offloading and service caching in the edge computing-based smart grid, the authors in [22] proposed a gradient descent allocation algorithm to determine the computing resource allocation strategy, and an algorithm based on game theory to determine the computing strategy. The authors in [23] proposed a method of saving the content service provider (CSP) based on a method of motivating drivers and deep Q-networks (DQN). [24] proposed an auction algorithm and a dynamic task admission algorithm to maximize the system average throughput in a 5G-enabled UAV-to-community offloading system. [25] proposed a deep reinforcement learning (DRL) additional particle swarm optimization algorithm to maximize the long-term utility of all mobile devices in the MEC-based mobile blockchain framework, which takes into account the limited bandwidth and computing power of small base stations. Edge intelligence technology empowers edge users with more powerful information processing and content delivery capabilities by scheduling computing, storage, and other resources for users. It profoundly changes the function of mobile applications and the utilization mode of network resources. What is more, it provides inspiration for constructing and disseminating RM with computational complexity, high bandwidth, and delay-sensitive requirements.

In this paper, we decompose the RM construction task into two subtasks, which are deployed in the MEC server and UEs according to offloading modes. In MEC networks, RMs are compressed before transmission in order to reduce bandwidth consumption. Then, we propose an actor-critic-based joint offloading and resource allocation (ACJORA) algorithm for intelligent scheduling of computation offloading and resource allocation in MEC networks. Our principal contributions are summarized as follows:

(1) We propose three modes of disseminating RMs in MEC networks and formulate the process as a mixed-integer nonlinear programming (MINLP) problem. Our objective is to minimize the energy consumption and delay of RM construction and dissemination

(2) To solve the above problem, we propose a DRL algorithm based on actor-critic for joint computation offloading and resource allocation. Considering the offloading decision actions are discrete and resource allocation actions are continuous, we design a weighted loss function including the two types of actions in one actor network, which significantly reduces the number of training parameters and improves the convergence efficiency of the algorithm

(3) Simulation experiments prove that the proposed ACJORA algorithm can find offloading and resource allocation strategies for RM dissemination effectively, which is more applicable for real-time RM applications

The remaining sections of this paper are organized as follows. Section 2 reviews the related work on the problem. Then, the network model and problem formulations are introduced in Section 3. Section 4 specifies the implementation details of our ACJORA algorithm. Performance evaluations are provided in Section 5, and Section 6 concludes the paper.

2. Related Work

2.1. Construction of RM. Accurate RM can provide better services, and the ways to improve the accuracy of RM mainly include optimizing the deployment of sensor devices and improving the accuracy of spatial interpolation [26, 27]. However, in some cases, sensor devices are predeployed and the deployment may not be optimal [28]. What is more, it is an uneconomical way to increase the number of sensor devices. Therefore, the interpolation accuracy needs to be improved under the limit of the number of sensors [29]. However, the construction complexity also increases as the accuracy increases. At present, RM is mainly constructed in the central cloud server, which is used for network planning or spectrum management and control in advance or for a long period. Some research has been carried out to reduce the complexity of RM construction. The authors in [30] proposed a method based on the Kalman filter. The work in [31] proposed a method based on regression kriging and incremental clustering. Both works of [30, 31] reduced the complexity of RM construction. [32] proposed RM construction method based on a superresolution (SR) algorithm which greatly shortens the construction time while improving the accuracy. This algorithm contains two phases: offline training and online conversion. In the offline training phase, RM images with different interpolation resolutions are used to train the return forest model with the best parameters. In the online conversion phase, the trained model can directly convert LR RM to high-resolution (HR) RM. In addition, the dissemination of accurate RM requires the scheduling of computing and communication resources, and edge intelligence provides us with solutions.

2.2. Edge Intelligence-Based IIoT. There has been a lot of research on edge intelligence in recent years, and some are used to solve problems in IIoT. The authors in [33] make a

review of research results that expounds on the development and convergence process of IIoT and edge computing. They propose an architecture for edge computing in IIoT and comprehensively explain it from multiple performance metrics. The authors in [34] considered the energy cost optimization problem of computing and caching in the Internet of Vehicles and integrated a deep deterministic policy gradient (DDPG) algorithm to solve this problem. [35] constructed a blockchain-enabled crowdsensing framework in intelligent transportation system. The authors proposed a DRL-based algorithm and a distributed alternating direction method of multipliers algorithm for distributed traffic management. [36] proposes a novel framework for optimizing edge collaborative network (ECN) to improve the stability between edge devices and the performance of edge computing tasks. [37] proposed a multiagent imitation learning-enabled UAV research deployment approach, which enables different UAV owners to provide services with differentiated service capabilities in a shared area. In a survey paper [38], the authors explored the emerging opportunities brought by 6G technologies in IoT networks and applications. They shed light on some 6G technologies that are expected to empower future IIoT networks, including edge intelligence, massive ultrareliable and low-latency communications, and blockchain.

2.3. Video Streaming Based on Edge Intelligence. Since the transmission of video streams is time-sensitive, and many video streaming tools (i.e. smartphones and VR devices) are limited by energy and computational capacity, just like the case we proposed. There are some research on video streaming transmission. The authors in [39] used a DRL algorithm to simultaneously optimize energy consumption and quality of service (QoS) for users during video streaming in edge networks. [40] proposed a peer-to-peer video streams transfer method based on MEC, which can perceive QoS for different users. The author modeled the transmission, computation, and offloading problem of video streams as a problem of maximizing QoS for users and then implemented an anti-fuzzy particle swarm optimization algorithm to optimize it. Distributed edge computing was used to optimize bandwidth consumption during video streaming [41]. The above methods of video streaming transmitting in edge intelligent networks mainly focused on computation offloading and resource allocation. However, RM dissemination needs to consider the construction and dissemination of RM at the same time. It is necessary to optimize the construction algorithms and compression coding of RM to reduce the use of computing and communication resources. Therefore, further research is needed on the construction and dissemination of RM.

3. System Model and Problem Formulation

We consider a MEC network with a base station (BS), a MEC server, and N UEs, as shown in Figure 1. The set of UEs is denoted by $\mathcal{N} = \{1, 2, \cdots, N\}$. UEs have radio receivers that collect spectrum data 5 times/sec based on crowdsensing without deploying radio receivers additionally. They send spectrum and location information to the BS with newly collected spectrum data.

We adopt the RM construction method based on the SR model in [32], which was trained by RM images with different interpolation resolutions. Due to the small size of the trained SR model, it consumes less bandwidth for transmission. The construction task of RM can be divided into two subtasks, e.g., the kriging interpolation algorithm and the SR model. The kriging interpolation algorithm can construct spectrum data as low-resolution (LR) RMs, and SR models convert LR RMs into HR RMs. In order to reduce time and energy consumption, the task of RM construction is decomposed and offloaded to the MEC server and UE. Therefore, it can transform the task of disseminating RMs into the task of disseminating mode with a small amount of data. The MEC server can determine whether it is necessary to offload RM construction task to the UE due to computation resources and bandwidth resources in the downlink. The dissemination mode decision variable of UE i can be denoted as $m_i \in \{0, 1, 2\}$. The three dissemination modes are described as follows:

(a) All server (mode 0, $m_i = 0$): in this mode, the construction task of RM only occurs on MEC server. First, the edge server completes the construction of the HR RM. Then, the HR RM is compressed and sent to UEs. This mode is suitable for situations in which the bandwidth of the dissemination link is abundant or the processing power of UE i is limited

(b) Partial offloading (mode 1, $m_i = 1$): in this mode, UE i needs to execute part of RM construction task. First, the edge server completes the construction of the LR RM and the training of the SR model. Then, the edge server disseminates the compressed LR RM and SR model to UE i, and UE i only needs to complete the SR conversion task. This mode is suitable for situations in which the dissemination link bandwidth is abundant and UE i has a certain processing capability

(c) All local (mode 2, $m_i = 2$): in this mode, MEC server disseminates the raw spectrum data of all UEs and trained SR model to UE i. UE i constructs the raw spectrum data into an LR RM. Then, it is transformed into an HR RM by the SR model. This mode is suitable for situations in which the bandwidth of the dissemination link is limited or UE i has a certain processing capability

We denote the RM construction computation task of UE i as $\tau_i = \{s_i, c_i, T_i^{\max}\}$. Here, s_i expresses the size of computation input data, c_i represents the number of CPU cycles required to accomplish the computation task, and $T_i^{\max}$ is the maximum tolerant delay of the task. In our model, computation tasks are considered to be decomposable. As a result, we decompose the construction task of RM into two subtasks, LR RM construction, and SR transformation. In addition, there is also a process of compression when MEC server disseminates RM to UE in mode 0 and mode 1. Table 1 shows the computation tasks in MEC networks.

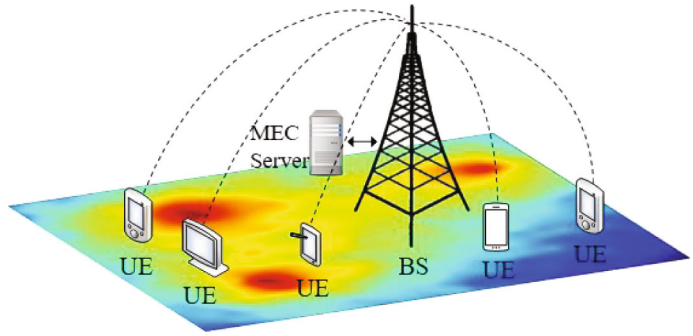

Figure 1: Network model.

Table 1: Computation tasks in MEC networks.

Computation tasks	Input data	CPU cycles
Spectrum data➞LR RM	$s_{i,1}$	$c_{i,1}$
LR RM➞HR RM	$s_{i,2}$	$c_{i,2}$
RM compression	$s_{i,3}$	$c_{i,3}$

3.1. Edge Server Execution Model. We denote the computational capacity (i.e., CPU cycles per second) of the edge server as F. And the computational capacity allocated by the edge server to UE i is F_i. The energy consumption of the edge server is calculated as

$$e_i^{\text{server}} = kF_i^2 c_i^{\text{server}}, \tag{1}$$

where $k = 10^{-27}$ is the effective switched capacitance of the CPU, determined by the CPU hardware architecture. c_i^{server} is the number of CPU cycles of computation tasks of UE i executed on the edge server, which is determined by the dissemination mode. (1) Mode 0: MEC server executes the computation tasks of constructing the spectrum data into an LR RM, constructing the LR RM into an HR RM, and compression of the RM. So, the number of CPU cycles of computation tasks executed by the edge server is $c_i^{\text{server}} = c_{i,1} + c_{i,2} + c_{i,3}$. (2) Mode 1: MEC server executes the computation task of constructing the spectrum data into an LR RM and compression of RM, $c_i^{\text{server}} = c_{i,1} + c_{i,3}$. (3) Mode 2: MEC server does not execute the computation task, $c_i^{\text{server}} = 0$. The time for edge server to execute the computation task can be expressed as

$$t_i^{\text{server}} = \frac{c_i^{\text{server}}}{F_i}. \tag{2}$$

3.2. Cognitive User Execution Model. We denote the computational capacity of UE i as f_i. The computation energy consumption of UE i is calculated as

$$e_i^{\text{local}} = kf_i^2 c_i^{\text{local}}, \tag{3}$$

where c_i^{local} is the number of CPU cycles of computation tasks executed by UE i, which is determined by the dissemination mode. (1) Mode 0: UE does not execute the computation task of RM, $c_i^{\text{local}} = 0$. (2) Mode 1: UE executes the computation task of SR model transformation, $c_i^{\text{local}} = c_{i,2}$. (3) Mode 2: UE executes the computation tasks of LR RM construction and SR model transformation, $c_i^{\text{local}} = c_{i,1} + c_{i,2}$. The time for edge server to execute the computation task can be expressed as

$$t_i^{\text{local}} = \frac{c_i^{\text{local}}}{f_i}. \tag{4}$$

3.3. Communication Model. The total communication bandwidth of the edge network is W, and the bandwidth allocated to UE i is W_i. Hence, the downlink transmission rate of UE i is calculated as

$$r_i = W_i \log_2\left(1 + \frac{ph_i}{\sigma^2}\right), \tag{5}$$

where p denotes the transmit power of the base station which is constant. h_i expresses the channel gain between UE i and the base station. σ^2 represents the noise power. The data transmission delay can be represented as

$$t_i^{\text{trans}} = \frac{d_i}{r_i}, \tag{6}$$

where d_i denotes the size of downlink transmission data between BS and UE i, which is determined by the dissemination mode. (1) Mode 0: MEC server transmits the HR RM to UE i, so the size of transmission data is denoted as $d_i = d_i^{\text{mode0}}$. (2) Mode 1: the edge server transmits the LR RM to UE i, so the size of transmission data is denoted as $d_i = d_i^{\text{mode1}}$. (3) Mode 2: the edge server transmits the raw spectrum data and trained SR model to UE i, so the size of transmission data is denoted as $d_i = d_i^{\text{mode2}}$. The data of computation and communication in different modes are shown in Table 2.

The transmission energy consumption of the data transmitted by MEC server to UE i can be calculated as

$$e_i^{\text{trans}} = pt_i^{\text{trans}}. \tag{7}$$

Table 2: The data of computation and communication in different modes.

Offloading modes	Computing data at edge server	Computing data at UE i	Transmission data
Mode 0	$c_i^{\text{server}} = c_{i,1} + c_{i,2} + c_{i,3}$	$c_i^{\text{local}} = 0$	d_i^{mode0}
Mode 1	$c_i^{\text{server}} = c_{i,1} + c_{i,3}$	$c_i^{\text{local}} = c_{i,2}$	d_i^{mode1}
Mode 2	$c_i^{\text{server}} = 0$	$c_i^{\text{local}} = c_{i,1} + c_{i,2}$	d_i^{mode2}

3.4. Problem Formulation. The total energy consumption of RM construction and dissemination for UE i can be calculated as

$$E_i = w_{\text{server}}^e e_i^{\text{server}} + w_{\text{local}}^e e_i^{\text{local}} + w_{\text{trans}}^e e_i^{\text{trans}}, \tag{8}$$

where w_{server}^e, w_{local}^e, and w_{trans}^e are the energy consumption weights for edge server execution, UE execution, and communication, respectively.

The total time spent on computation and communication can be calculated as

$$T_i = w_{\text{server}}^t t_i^{\text{server}} + w_{\text{local}}^t t_i^{\text{local}} + w_{\text{trans}}^t t_i^{\text{trans}}, \tag{9}$$

where w_{server}^t, w_{local}^t, and w_{trans}^t are the execution delay weights for edge server execution, UE execution, and communication, respectively.

In order to minimize the sum cost of execution delay and energy consumption for RM construction and dissemination, we formulate the weighted sum of energy and delay as the total consumption of the MEC system. Under the constraint of computation and bandwidth capacity and maximum tolerable delay, the problem can be optimized as follows:

$$\begin{aligned} &\min_{m,W,F} \sum_{i=1}^{n} (1-\varpi) E_i + \varpi T_i \\ &\text{s.t.} C1: m_i \in \{0, 1, 2\} \\ &C2: k f_i^{\,2} c_i^{\text{local}} \le e_i^{\text{local}} \\ &C3: \sum_{i}^{N} F_i \le F_{\max} \\ &C4: \sum_{i}^{N} W_i \le W_{\max} \\ &C5: 0 \le T_i^{\text{sum}} \le T_{\max}. \end{aligned} \tag{10}$$

In the above problem, ϖ is the weight for execution delay. $m = (m_1, m_2, \cdots, m_n)$ is the offloading decision vector, $W = (W_1, W_2, \cdots, W_n)$ is the bandwidth allocation, and $F = (F_1, F_2, \cdots, F_n)$ is the computation resource allocation of edge server. Besides, $C1$ represents the offloading mode of UE i. $C2$ expresses the energy consumption of UE i that does not exceed its remaining energy. $C3$ indicates the sum of computation resources allocated to all UEs that cannot exceed the computation capacity of MEC server. $C4$ expresses that the sum of the bandwidth allocated to all UEs cannot exceed the available bandwidth of MEC network. $C5$ represents that the sum time for RM construction and dissemination does not exceed the tolerance time of the task. T_i^{sum} denotes the total time of RM construction and dissemination.

$$T_i^{\text{sum}} = t_i^{\text{server}} + t_i^{\text{local}} + t_i^{\text{trans}}. \tag{11}$$

Note that the offloading decision variables and the resource allocation variables correspond to integer variables and continuous variables, respectively. Therefore, it is an MINLP and NP-hard problem for the objective function, which has no convex feasible set. And the complexity of the feasible set grows exponentially with the number of UEs. Since traditional model-based methods are incapable of dealing with dynamic scenarios, we adopt a DRL approach, which is model-free.

4. DRL-Based Joint Offloading and Resource Allocation Algorithm

According to the optimization objectives and constraints of the problem, we solve it with an ACJORA algorithm. This section first defines the state space, action space, and reward function of the model. Then, we introduce the proposed actor-critic algorithm framework in detail.

4.1. State Space, Action Space, and Reward Function. According to the system model, the state space, action space, and reward function are defined as follows.

4.1.1. State Space. The state space can be presented by

$$S = \left\{ s_t | s_t = \left(e_t^{\text{local}}, F_t^{\text{remain}}, W_t^{\text{remain}} \right), t \in M \right\}, \tag{12}$$

where s_t denotes the network state at step t. The available resource state at t is represented as $F_t^{\text{remain}} = F_{\max} - \sum_i^n F_i$ and $W_t^{\text{remain}} = W_{\max} - \sum_i^n W_i$. F_t^{remain} expresses the available computation resources of the MEC server, and W_t^{remain} denotes the available communication bandwidth resource of the MEC network. The purpose of observing them is to ensure meet the constraints of computational capacity and communication channel capacity. In addition, we also needs to observe the remaining energy of UEs e_t^{local} to avoid that the energy of UE is not enough to complete the computation tasks allocated in the next period.

4.1.2. Action Space. The action space can be denoted as

$$A = \{ a_t | a_t = (m_t, F_t, W_t), t \in M \}, \tag{13}$$

which consists of three vectors: offload decision vector $m_t = (m_1^t, m_2^t, \cdots, m_n^t)$, computation resource allocation vector $F_t = (F_1^t, F_2^t, \cdots, F_n^t)$, and spectrum resource allocation vector $W_t = (W_1^t, W_2^t, \cdots, W_n^t)$. In a MEC system network, MEC server disseminates offload strategies to UEs. Meanwhile, the computation and communication resources allocated to UEs should also be determined.

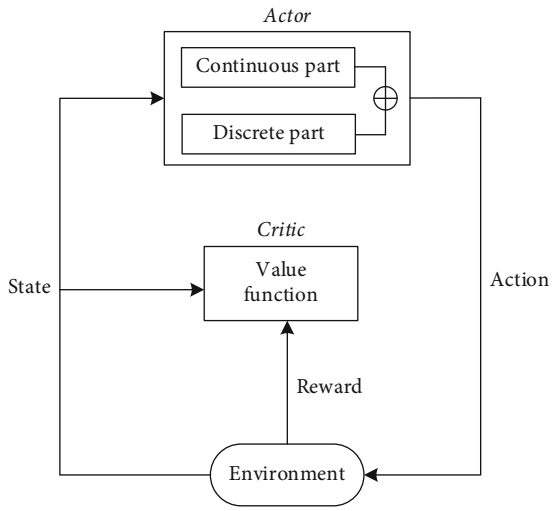

Figure 2: The actor-critic algorithm framework.

4.1.3. Reward Function. The immediate reward function is generally related to the objective function. In Equation (10), our goal is to obtain the smallest sum cost of energy and time consumption, while the goal of reinforcement learning is to obtain the largest reward. Therefore, the value of the reward needs to be negatively related to the value of the sum cost. The sum cost of the system at time t is denoted as cost_t. Thus, the immediate reward obtained by executing policy a_t in state s_t can be defined as

$$r_t = \text{-cost}_t. \tag{14}$$

4.2. DRL-Based Algorithm. Basically, reinforcement learning algorithms can be classified into three types: actor-only, critic-only and actor-critic [42]. Actor-only methods employ policy functions (i.e., policy gradient methods) to learn stochastic policies efficiently for models with large action spaces and converge asymptotically to local optima, which is more feasible for the models with continuous actions. However, they often cause high variance in expected reward estimates and slow learning. Critic-only methods with value functions (i.e., action-value methods) typically use time difference (TD) iterations and thus have lower variance in expected reward estimates. However, they need to use optimizers in each state encountered to find an action with the highest expected rewards. Therefore, they are not effective to solve problems with large action spaces, which is the case for the problem that we have stated. Furthermore, they need to discretize continuous actions since they are based on discrete action values. Consequently, we adopt an actor-critic approach, which combines the merits of critic-only and actor-only algorithms. The actor can produce continuous or discrete actions without requiring an optimizer for the value function. The critic employs the estimate function to estimate the output of the actor, and the actor updates the policy parameters according to the estimated value to make the variance lower. [43, 44].

We propose an actor-critic algorithm to determine continuous actions for resource allocation and discrete actions for task offloading, as shown in Figure 2. The actor-critic model contains a critic network and an actor network. For the actor network, we derive a novel weighted loss function with two different action outputs, the continuous part and the discrete part. The learning rate of discrete and continuous action training is updated according to the weighted loss function in the actor training phase. And the actor parameters of continuous actions are iterated by gradient ascent based on DDPG [45], which is given as

$$\frac{1}{Z}\sum_{k}\nabla_{a_c}Q\left(s,\mu(s|\theta^{\mu})\middle|\theta^{Q}\right)\Big|_{s=s_k}\nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})\Big|_{s_k}, \tag{15}$$

where Z represents the size of sample batch; θ^Q and θ^μ, respectively, express the weight and bias parameters in the critic network and the actor network; $\mu(\cdot)$ and $Q(\cdot)$ correspond to the output of actor network and critic network, respectively; and a_c is the continuous part action component for resource allocation F_t and W_t. We compute the gradient of $Q(s,\mu(s|\theta^\mu)|\theta^Q)$ to a_c instead of the whole $\mu(\cdot)$. Then, we consider it is constant for the discrete part action component (i.e., offloading decision vector m_t). The gradient of $Q(s,\mu(s|\theta^\mu)|\theta^Q)$ for the discrete action is calculated as

$$-\frac{1}{Z}\sum_{k}Q\left(s,\mu(s|\theta^{\mu})\middle|\theta^{Q}\right)\Big|_{s=s_k}\cdot\nabla_{\theta^{\mu}}\left(\frac{1}{n}\sum_{i}m_i^k p_i^k\right), \tag{16}$$

where $m_i^k \in \{0,1,2\}$ is the offloading decision variable in the kth sample from the replay buffer and $(p_1^k, p_2^k, \cdots, p_n^k)$ denotes the probability of offloading modes for UEs, which is the first component of the actor output $\mu(\cdot)$. Different from Equation (15), the continuous part action a_c and $Q(s,\mu(s|\theta^\mu)|\theta^Q)$ are, respectively, fixed at constant action and constant weight. The weighted loss function for discrete and continuous actions can be expressed as

$$\begin{aligned}L^{\text{actor}} = &-w_c\sum_{k}Q\left(s,\mu(s|\theta^{\mu})\middle|\theta^{Q}\right)\Big|_{s=s_k}\\ &-w_d\sum_{k}Q\left(s,\mu(s|\theta^{\mu})\middle|\theta^{Q}\right)\Big|_{s=s_k}\cdot\left(\frac{1}{n}\sum_{i}m_i^k p_i^k\right),\end{aligned} \tag{17}$$

where w_c and w_d, respectively, correspond to the weight of the discrete and continuous part loss function. α_d and α_c denotes the learning rates of discrete and continuous part training phases. And they are update according to Equation (17) to get a good convergence performance.

For the critic network, we use the average square error loss function to iterate the parameters, which is defined as

$$L^{\text{critic}} = \frac{1}{Z}\sum_{k}\left(y_k - Q\left(s_k,\mu(s_k|\theta^{\mu})\middle|\theta^{Q}\right)\right)^2. \tag{18}$$

Based on the above definitions, the proposed ACJORA algorithm is presented in Algorithm 1.

Input: actor network parameters θ^{μ}, critic network parameters θ^{Q}, actor target network parameters $\theta^{\mu'}$, critic target network parameters $\theta^{Q'}$, discount factor γ, replay buffer B, batch size Z, epsilon greedy ε
Output: the best strategy (m, W, F)
Initialize: randomly initialize θ^{μ} and θ^{Q}, $\theta^{\mu'} \longleftarrow \theta^{\mu}$, $\theta^{Q'} \longleftarrow \theta^{Q}$, $B \longleftarrow \varnothing$
forepisode = 1 to M**do**
  Initialize state s_0
  for$t = 1$ to T**do**
    Actor output $(\widehat{m}_t, \widehat{F}_t, \widehat{W}_t) \longleftarrow \mu(s_t|\theta^{\mu})$.
    Add noise on $\mu(s_t|\theta^{\mu})$ with ε-greedy on $\widehat{m}_t$ and Gaussian distribution with mean $(\widehat{F}_t, \widehat{W}_t)$.
    Get action $a_t = (\widehat{m}_t, \widehat{F}_t, \widehat{W}_t)$ with exploration variance V.
    Take action a_t, observe reward r_t and next state s_{t+1}.
    Store transition (s_t, a_t, r_t, s_{t+1}) in B.
    Sample a random batch of Z transitions $(s_k, \mu(s_k|\theta^{\mu}), r_k, s_{k+1})$ from B.
    Set $y_k = r_k + \gamma Q'(s_{k+1}, \mu'(s_{k+1}|\theta^{\mu'}|\theta^{Q'}))$.
    Update the critic with minimizing the loss L^{critic} by Equation (18).
    According to the loss L^{actor}, update the actor through the continuous part training phase lr_c and the discrete part training phase lr_d by Equations (15) and (16)
    Update the target networks: $\theta^{Q'} \longleftarrow \lambda\theta^{Q} + (1-\lambda)\theta^{Q'}$, $\theta^{\mu'} \longleftarrow \lambda\theta^{\mu} + (1-\lambda)\theta^{\mu'}$.
  end for
end for

Algorithm 1: Actor-critic-based joint offloading and resource allocation algorithm.

Table 3: Simulation parameters.

Parameters	Value
The number of UEs (N)	5
The communication bandwidth (W)	10 MHz
The computational capacity of MEC server (F)	8 GHz
The computational capacity of UE i (f_i)	[0.5,1.5] GHz
The distance between BS and UEs	[0, 200] m
The transmit power of BS (p)	500 mW
The maximum tolerated delay for RM construction and dissemination	1 s
The LR RM size	100 Kbit
The HR RM size	250 Kbit
Discount factor (γ)	0.99
Replay buffer (B)	100
Batch size (Z)	32
Epsilon greedy (ε)	0.9

Table 4: The average latency of three scheme.

	All server	All local	Random offloading	DQN-based	Proposed
Average latency (ms)	878	643	735	286	194

5. Simulation Results

5.1. Parameter Setting. In the simulations, we consider a MEC network as show in Figure 1, which includes a BS and N UEs. A MEC server is connected to the BS. UEs are randomly distributed in an area of [0, 200] meters from the BS. Communication bandwidth is $W = 10$ MHz. The computational capacity of MEC server is $F = 8$ GHz, the CPU frequency of the UE is random in the range [0.5, 1.5] GHz. The LR RM size is set as 100 Kbit (e.g., a LR image is 1280×720 with 16 bit/pixel, using a compression ratio of 150 : 1 [46]). And the HR RM size is set as 250 Kbit (e.g., a HR image is 1920×1080 with 16 bit/pixel, using a compression ratio of 150 : 1). The maximum tolerated delay for RM construction and dissemination is $T_{\max} = 1$ s. The transmit power of BS is $p = 500$ mW. Detailed simulation parameters are listed in Table 3.

5.2. Result Analysis. For fair performance evaluation, we compare our proposed scheme with four baseline schemes: (1) All server: the offloading decision variable of all UEs is $m_i = 0$. The MEC server constructs RM and disseminates the compressed RM. (2) All local: the offloading decision

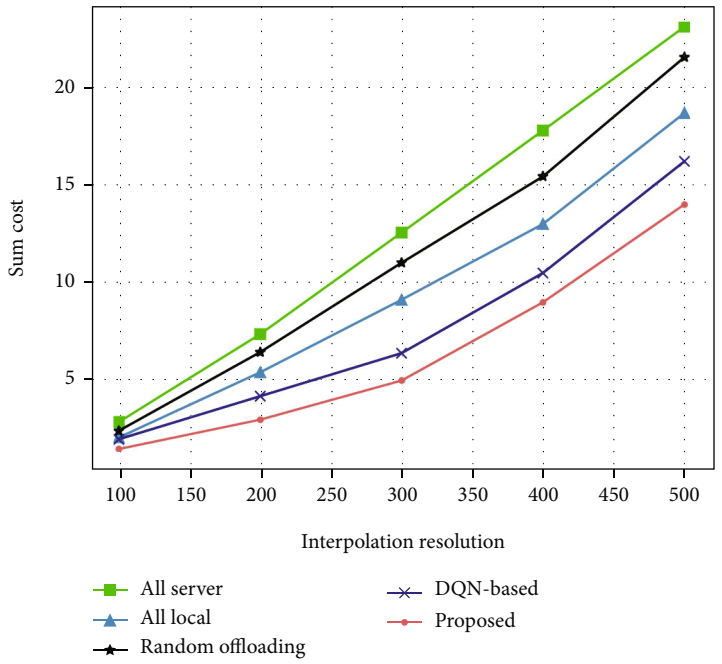

Figure 3: The effect of interpolation resolution on the sum cost.

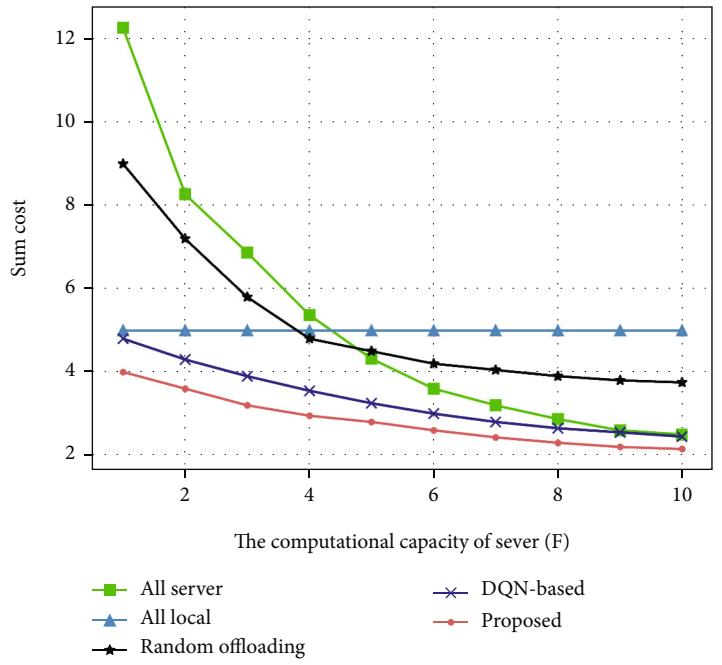

Figure 4: The effect of the computational capacity for server on total consumption.

variable of all UEs is $m_i = 2$, distributing global spectrum information and constructing RM at user equipment. (3) Random offloading: all UEs randomly select one of the three dissemination methods: all server, all local, and partial offloading, $m_i \in \{0, 1, 2\}$. (4) DQN-based: a resource scheduling scheme based on deep Q-network (DQN), which is an action-value method [47].

Table 4 shows the average latency of disseminating RM to UEs in a scenario where the number of UEs is 5, the computational capacity of MEC server is 8 GHz/sec, and the communication bandwidth is 10 MHz. We conducted 5000 Monte Carlo experiments and took the average value of the experimental results. The average delay of the proposed scheme is 194 ms, which is 77.90% lower than that of all server construction scheme, 69.83% lower than that of all local construction scheme, and 73.61% lower than that of the random offloading scheme, 32.17% lower than DQN-based scheme.

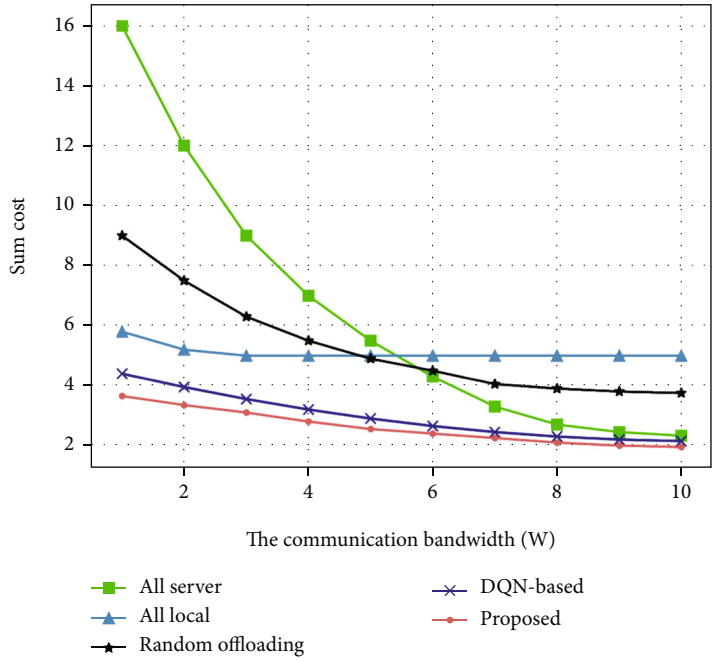

Figure 5: The effect of communication bandwidth on total consumption.

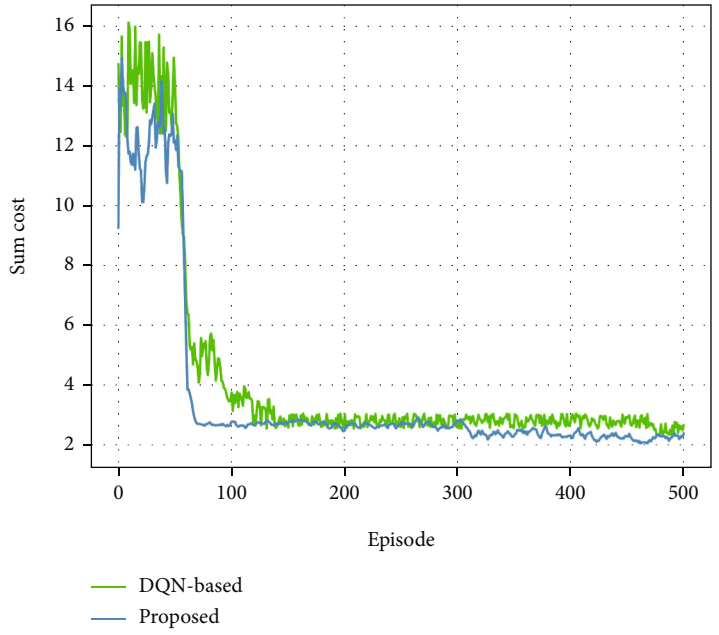

Figure 6: The comparison of convergence performance.

As shown in Figure 3, with the increasing of interpolation resolution (i.e., the number of interpolation points by Kriging interpolation), the sum cost of all schemes increase at the same time. Because the computational complexity and the data size of RM dissemination will increase as the interpolation resolution increases. The DRL scheme can efficiently allocate computation and communication resources. However, DQN-based scheme cannot produce continuous actions, so it is need to discretize continuous variables. Thus, the strategies of the DQN-based scheme are worse than that of the proposed scheme in resource allocation. The proposed scheme can achieve the best performance with minimal sum cost with the increase of RM interpolation resolution.

Figure 4 illustrates the sum cost of the MEC system as the computational capacity of the edge server increases. The all local curve does not change with the increase of the computation resources of MEC server because UEs does not use the computation resources of the server. The other curves

decrease as the computational capacity of MEC server increases. Because each UE is allocated more server computation resources, the computation time will be shortened accordingly. In addition, when $F > 8$ GHz/sec, the sum cost of all server scheme and the proposed scheme decreases slowly. The result shows that when the computation resources of MEC server are far more than the computation resource of UE, the sum cost of MEC network is mainly limited by other factors such as communication bandwidth resources.

Figure 5 depicts the sum cost of the MEC system as communication bandwidth increases. Due to the small transmission data of the all local scheme, it is less affected by the communication bandwidth. And the sum cost of the all local scheme at $W = 3$ MHz does not change with the increase of communication bandwidth. The sum cost of other schemes decrease with the increase of the communication bandwidth, because each UE can be allocated more bandwidth resources, and the communication transmission time will be shortened. And the proposed scheme has the least sum cost. Figures 4 and 5 show that the proposed scheme has good adaptability in a varying radio environment.

We compared the convergence performance of DQN-based scheme and proposed scheme in Figure 6. The sum cost of the both RL learning schemes has decreased rapidly with the number of episodes. Finally, the most effective offloading and resource allocation strategies are learned and the sum cost of the system has stabilized. Compared with the DQN-based scheme, the proposed scheme can converge with fewer episodes, and its sum cost is less. Figure 6 shows that the proposed scheme can efficiently train offloading and resource allocation strategies.

6. Conclusions

RM is an important tool for cognitive radio in the 6G era, which can provide data support for IIoT. To solve the problem of RM construction and dissemination in resource-limited and delay-sensitive MEC networks, we have proposed a joint RM construction and dissemination approach based on DRL, which is described as actor-critic-based joint offloading and resource allocation (ACJORA) algorithm. In the algorithm, we designed a novel actor-critic model with a weighted loss function for the actor network, which combines the discrete actions for task offloading and continuous actions for resource allocation. Compared with baseline schemes, the proposed algorithm can effectively reduce the energy consumption and delay of RM construction and dissemination, which significantly reduces the cost of acquiring RM for UEs.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 62171449, 62001483, and U19B2024.

References

[1] K. K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic solutions for industrial internet-of-things: research challenges and opportunities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3567–3569, 2018.

[2] M. Z. Hasan and H. al-Rizzo, "Optimization of sensor deployment for industrial internet of things using a multiswarm algorithm," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10344–10362, 2019.

[3] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial internet of things security: requirements and fog computing opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489–2520, 2020.

[4] K. Katagiri and T. Fujii, "Mesh-clustering-based radio maps construction for autonomous distributed networks," in *2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 345–349, Jeju Island, Korea, August 2021.

[5] P. Bednarek, J. Łopatka, and D. Bicki, "Radio environment map for the cognitive radio network simulator," *International Journal of Electronics and Telecommunications*, vol. 64, no. 1, pp. 45–49, 2018.

[6] N. Ezzati and H. Taheri, "Distributed spectrum sensing in rem based cognitive radio networks," *Journal of Modeling in Engineering*, vol. 17, no. 56, pp. 223–233, 2019.

[7] P. Kaniewski, J. Romanik, E. Golan, and K. Zubel, "Spectrum awareness for cognitive radios supported by radio environment maps: zonal approach," *Applied Sciences*, vol. 11, no. 7, p. 2910, 2018.

[8] Y. H. Santana, D. Plets, R. M. Alonso et al., "Tool for recovering after meteorological events using a real-time REM and IoT management platform," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 9767404, 13 pages, 2019.

[9] S. C. Arum, D. Grace, and P. D. Mitchell, "A review of wireless communication using high-altitude platforms for extended coverage and capacity," *Computer Communications*, vol. 157, pp. 232–256, 2020.

[10] H. Braham, S. B. Jemaa, G. Fort, E. Moulines, and B. Sayrac, "Fixed rank kriging for cellular coverage analysis," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4212–4222, 2017.

[11] H. Braham, S. B. Jemaa, G. Fort, E. Moulines, and B. Sayrac, "Spatial prediction under location uncertainty in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7633–7643, 2016.

[12] L. Yin, J. Luo, and H. Luo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4712–4721, 2018.

[13] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks," *IEEE Access*, vol. 6, pp. 32328–32338, 2018.

[14] R. Atawia, H. S. Hassanein, N. Abu Ali, and A. Noureldin, "Utilization of stochastic modeling for green predictive video delivery under network uncertainties," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 2, pp. 556–569, 2018.

[15] M. Li, J. Gao, L. Zhao, and X. Shen, "Deep reinforcement learning for collaborative edge computing in vehicular networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 4, pp. 1122–1135, 2020.

[16] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[17] F. Khoramnejad and M. Erol-Kantarci, "On joint offloading and resource allocation: a double deep q-network approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1126–1141, 2021.

[18] S. Nath and J. Wu, "Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems," *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 181–198, 2020.

[19] A. M. Seid, G. O. Boateng, S. Anokye, T. Kwantwi, G. Sun, and G. Liu, "Collaborative computation offloading and resource allocation in multi-UAV-assisted IoT networks: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12203–12218, 2021.

[20] H. Zhou, K. Jiang, X. Liu, X. Li, and V. C. M. Leung, "Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1517–1530, 2021.

[21] Z. Ning, Y. Yang, X. Wang et al., "Dynamic computation offloading and server deployment for UAV-enabled multi-access edge computing," *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[22] H. Zhou, Z. Zhang, D. Li, and Z. Su, "Joint optimization of computing offloading and service caching in edge computing-based smart grid," *IEEE Transactions on Cloud Computing*, p. 1, 2022.

[23] H. Zhou, T. Wu, H. Zhang, and J. Wu, "Incentive-driven deep reinforcement learning for content caching and D2D offloading," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2445–2460, 2021.

[24] Z. Ning, P. Dong, M. Wen et al., "5G-enabled UAV-to-community offloading: joint trajectory design and task scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 11, pp. 3306–3320, 2021.

[25] Z. Ning, S. Sun, X. Wang et al., "Intelligent resource allocation in mobile blockchain for privacy and security transactions: a deep reinforcement learning based approach," *Science China Information Sciences*, vol. 64, no. 6, article 162303, 2021.

[26] M. Höyhtyä, A. Mämmelä, M. Eskola et al., "Spectrum occupancy measurements: a survey and use of interference maps," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2386–2414, 2016.

[27] L. Zhou, Z. Sheng, L. Wei et al., "Green cell planning and deployment for small cell networks in smart cities," *Ad Hoc Networks*, vol. 43, pp. 30–42, 2016.

[28] X. Wang, Z. Ning, X. Hu et al., "Future communications and energy management in the internet of vehicles: toward intelligent energy-harvesting," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 87–93, 2019.

[29] H. B. Yilmaz, T. Tugcu, F. Alagöz, and S. Bayhan, "Radio environment map as enabler for practical cognitive radio networks," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 162–169, 2013.

[30] V. P. Chowdappa, C. Botella, J. J. Samper-Zapater, and R. J. Martinez, "Distributed radio map reconstruction for 5G automotive," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 36–49, 2018.

[31] A. P. Sergeev, D. A. Tarasov, A. G. Buevich et al., "High variation subarctic topsoil pollutant concentration prediction using neural network residual kriging," *AIP Conference Proceedings*, vol. 1836, no. 1, article 020023, 2017.

[32] Y. Deng, L. Zhou, L. Wang et al., "Radio environment map construction using super-resolution imaging for intelligent transportation systems," *IEEE Access*, vol. 8, pp. 47272–47281, 2020.

[33] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, and D. O. Wu, "Edge computing in industrial internet of things: architecture, advances and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2462–2488, 2020.

[34] X. Kong, G. Duan, M. Hou et al., "Deep reinforcement learning-based energy-efficient edge computing for internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6308–6316, 2022.

[35] Z. Ning, S. Sun, X. Wang et al., "Blockchain-enabled intelligent transportation systems: a distributed crowdsensing framework," *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[36] X. Kong, S. Tong, H. Gao et al., "Mobile edge cooperation optimization for wearable internet of things: a network representation-based framework," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5050–5058, 2021.

[37] X. Wang, Z. Ning, S. Guo, M. Wen, L. Guo, and V. Poor, "Dynamic UAV deployment for differentiated services: a multi-agent imitation learning based approach," *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[38] D. C. Nguyen, M. Ding, P. N. Pathirana et al., "6G internet of things: a comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2022.

[39] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "MEC-assisted immersive VR video streaming over terahertz wireless networks: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9517–9529, 2020.

[40] A. E. M. Taha, N. Abu Ali, H. R. Chi, and A. Radwan, "MEC resource offloading for QoE-aware HAS video streaming," in *ICC 2021 - IEEE International Conference on Communications*, pp. 1–5, Montreal, QC, Canada, June 2021.

[41] K. Genda, M. Abe, and S. Kamamura, "Video communication optimization using distributed edge computing," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 381–384, Daegu, Korea (South), September 2020.

[42] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: an actor-critic reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 680–692, 2018.

[43] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "TACT: a transfer actor-critic learning framework for energy saving in cellular radio access networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 2000–2011, 2014.

[44] W. Jiang, D. Feng, Y. Sun, G. Feng, Z. Wang, and X.-G. Xia, "Proactive content caching based on actor–critic reinforcement learning for mobile edge networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1239–1252, 2021.

[45] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2015, https://arxiv.org/abs/1509.02971.

[46] E. Bastug, M. Bennis, M. Medard, and M. Debbah, "Toward interconnected virtual reality: opportunities, challenges, and enablers," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.

[47] Z. Wu and D. Yan, "Deep reinforcement learning-based computation offloading for 5G vehicle-aware multi-access edge computing network," *China Communications*, vol. 18, no. 11, pp. 26–41, 2021.