AoI-Aware Task Offloading and Transmission Optimization for Large-Scale IIoT Networks: A Branching Deep Reinforcement Learning Approach

Yuang Chen, *Graduate Student Member, IEEE,* Chang Wu, *Graduate Student Member, IEEE,* Fengqian Guo, Hancheng Lu, *Senior Member, IEEE,* Chang Wen Chen, *Life Fellow, IEEE*

***Abstract*—In the sixth-generation mobile communication (6G) discussed by IMT-2030, significant emphasis is placed on massive communications and ubiquitous connectivity, which has greatly driven the explosive growth in the number of devices with stringent timeliness demands, represented by the industrial Internet of Things (IIoT). In this paper, we propose an age-of-information (AoI)-aware multi-base station (BS) real-time monitoring framework to support large-scale IIoT access. We formulate a joint task offloading and resource allocation optimization problem with the goal of minimizing long-term average AoI, while considering the delay, energy consumption, resource allocation, and task scheduling constraints. For the task offloading subproblem, we equivalently transform it into a constrained Markov decision process (CMDP) and propose an innovative branching-based Dueling Double Deep Q-Network (Branch-D3QN) algorithm to effectively tackle it, significantly reducing the action space complexity from exponential to $\mathcal{Q}(n)$, enhancing convergence speed and stability. For the resource allocation subproblem, we prove the convexity of this subproblem by deriving the semi-definite property of the Hessian matrix of bandwidth and computation resources. Furthermore, we propose an iterative optimization algorithm for efficient joint task offloading and resource allocation to achieve optimal average AoI performance. Extensive simulations show that our proposed Branching-D3DN algorithm outperforms both the state-of-the-art DRL methods and classical heuristics, achieving up to 75% enhanced convergence speed and at least 22% reduced long-term average AoI.**

***Index Terms*—Age of Information (AoI); Delay-Sensitive Tasks; Deep Reinforcement Learning (DRL); Task Offloading; Resource Allocation; Industrial Internet of Things (IIoT).**

I. Introduction

With the gradual maturation of fifth-generation mobile communication (5G) technology and the large-scale commercial application of internet of things (IoT) technology, modern communications have entered a new era of interconnected everything, and people's daily lives have undergone significant transformations [1–3]. In this era of the Internet of Everything, numerous items including vehicles, sensors, mobile phones, tablets, and wearable devices will be connected to the network, giving rise to a series of entirely new applications such as smart transportation, smart cities, autonomous driving, and drone rescue. Currently, there are 18.8 billion active IoT devices globally, with Industrial IoT (IIoT) accounting for 60% of new IoT device installations [4]. According to the latest report from the authoritative industry analysis firm Transforma Insights [5], it is projected that the number of active IoT devices worldwide will grow to 40.6 billion by 2034, achieving a compound annual growth rate (CAGR) of 9%. Additionally, annual device sales are expected to increase from 4.3 billion in 2024 to 9 billion 2034 with the CAGR of 8%. According to a Dataprot report, the amount of data generated by global IoT devices has grown from 17.3 ZB in 2019 to an astonishing 73.1 ZB by 2025 [6].

In this context, due to the surge in the scale of IoT devices, future network will inevitably face increased demands for computation and communication resources to effectively handle the massive data generated by these devices and the substantial computational and transmission burdens they impose [7, 8]. Given the limited computation capacity and battery power of IoT devices, relying solely on local computation typically falls short of meeting these emerging task requirements [9, 10]. Consequently, devices usually need to offload computational tasks wirelessly to resource-rich locations for processing. Traditional cloud computing paradigms address this by offloading computationally intensive and delay-sensitive tasks to the cloud [11]. However, the considerable distance between IoT devices and the cloud center, along with path loss during wireless transmission, results in inefficient data transmission [12, 13]. Moreover, the centralized nature of cloud-based computational tasks not only increases the cloud's load but also reduces resource utilization and introduces high system delay [14]. These challenges present opportunities for the rapid development of mobile edge computing (MEC) technology [11–14].

Although offloading computing tasks from IoT devices to nearby edge servers can effectively reduce system delay and improve quality of service (QoS) performance, meeting the task offloading and transmission needs of large-scale devices has become one of the most challenging issues with the rapid growth of IoT device numbers [7, 8]. Even more complex, the tasks involved in IIoT devices typically have strict timeliness requirements (i.e., age of information, AoI) [15, 16], which place higher demands on the computation and communication capabilities of networks. To this end, the system needs to accurately evaluate and optimize timeliness in real-time application

This work was supported in part by the National Science Foundation of China under Grant U21A20452, in part by Hong Kong Research Grants Council (GRF-15213322, GRF-15229423). Yuang Chen and Chang Wu contributed equally to this work and are co-first authors. Yuang Chen, Chang Wu, Fengqian Guo, and Hancheng Lu are with the Laboratory of Future Networks, University of Science and Technology of China (USTC), Hefei, P. R. China (e-mail: {yuangchen21, changwu}@mail.ustc.edu.cn; fqguo@ustc.edu.cn; hclu@ustc.edu.cn). Yuang Chen and Hancheng Lu are also with Deep Space Exploration Laboratory, Hefei 230088, China. Chang Wen Chen is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: changwen.chen@polyu.edu.hk). The corresponding authors are Prof. Hancheng Lu and Prof. Chang Wen Chen.

scenarios, especially in terms of the timeliness of task and device information updates, to ensure that the system can make correct decisions based on the latest information. Therefore, how to effectively evaluate and optimize the timeliness of information in real-time systems has become a key issue in improving the performance of IoT systems [17, 18].

Furthermore, the widespread application of intelligent applications has promoted the rapid development of real-time monitoring applications. In these applications, devices need to quickly perceive data from the surrounding environment and transmit tasks with time sensitive requirements to the MEC server for processing, ensuring that the system can perform effective decision-making and control. For real-time monitoring systems, the tasks generated by devices are usually dynamically changing, therefore, the timeliness of device tasks is crucial for the receiving end to make correct decisions. In such systems, the timeliness of information becomes a key indicator, and the concept of AoI can effectively represent the timeliness of information from a mathematical perspective [19, 20]. AoI is not only closely related to system latency, but also to the generation time interval of tasks (i.e. task arrival rate), which is the biggest difference between AoI and latency indicators [21]. Therefore, the introduction of AoI provides a new perspective for evaluating system timeliness.

A. Motivations and Challenges

The core of implementing the real-time monitoring systems with large-scale IIoT access lies in accurately evaluating how task offloading and resource allocation strategies affect system performance [10, 22–26]. Although extensive research on transmission optimization for delay-sensitive IIoT tasks in edge networks has established a foundation and provided effective solutions for managing massive IIoT device access, significant challenges persist in practical applications.

Existing queuing theory-based AoI models are oversimplified and fail to capture system dynamics accurately. Timeliness optimization in real-time monitoring systems faced multiple challenges, particularly in accurately modeling and controlling dynamically changing system behavior. Traditional approaches typically assume fixed and known service parameters (e.g., service rate), whereas in practice, service capabilities fluctuate due to factors such as network conditions, resource contention, and device performance [27, 28]. By definition, the value of AoI at time t depends only on the current state but also on prior updates, making it naturally compatible with the Markov Decision Process (MDP) framework [28, 29]. Nonetheless, AoI optimization is inherently stochastic and dynamic, making it difficult to solve with conventional convex optimization methods. This challenge motivates the adoption of more flexible optimization tools and intelligent decision-making methods.

Large-scale IIoT access in multi-base station (BS) scenarios intensifies the complexity of task offloading and resource allocation. The majority of existing studies predominantly consider single-BS environments [8, 15, 16, 26, 30], which do not reflect the prevalent multi-BS configurations in practical communication systems. Multi-BS systems can provide enhanced computational and communication resources, essential for supporting the offloading demands of large-scale IIoT devices [12, 13, 31]. However, the introduction of multiple BSs complicates the process of determining optimal offloading strategies. Moreover, in multi-BS systems, the computational complexity increases, necessitating algorithmic refinements for efficient offloading and transmission decisions to ensure rapid convergence. Therefore, the inclusion of multi-BSs introduces greater complexity to the task transmission and offloading processes, requiring further research to resolve these challenges effectively.

Existing reinforcement learning (RL) algorithms face significant limitations, such as the curse of dimensionality in multi-BS scenarios with large-scale IIoT access. While RL serves as an effective tool for solving MDPs [32, 33], task offloading strategies typically involve discrete binary decision variables, rendering traditional RL methods difficult to apply directly [34]. Deep Q-Networks (DQN) were introduced to address discrete decision-making problems, however, it suffers from overestimation and maximization bias due to using the same neural networks for both action evaluation and selection [32]. Double DQN (DDQN) mitigates this issue by decoupling action selection from evaluation, though it still struggles with distinguishing the true impact of actions when states exhibit highly similar or overlapping effects [35]. Dueling DQN improves on this by decomposing the Q-function into state-value functions, further reducing bias [36]. The Dueling Double DQN (D3QN) algorithm combines the architectural benefits of Dueling DQN and the bias-reduction strategy of DDQN [37–40]. However, when applied to multi-BS systems with massive IIoT access, the exponential growth of the action space results in a severe curse of dimensionality, significantly hindering learning efficiency and convergence [39].

B. Main Contributions

In order to effectively overcome the challenges mentioned above, this paper proposes an AoI-aware multi-BS real-time monitoring system for large-scale IIoT access. We systematically formulate the joint task offloading and resource allocation optimization problem with the goal of minimizing the system's long-term average AoI, while considering the delay, bandwidth, computation capacity, energy consumption, and task scheduling constraints. To effectively tackle this non-convex dynamic stochastic optimization problem, we decompose it into two subproblems. For task offloading, we first introduce a branching structure into the DRL framework to effectively mitigate the dimensionality explosion and convergence issues caused by large-scale device and BS access. For resource allocation, we derive and prove the semi-definite property of the Hessian matrix of bandwidth and computation resources with respect to long-term average AoI. Both theoretical analysis and extensive experiments validate the efficiency and superiority of the proposed methods. The primary contributions of this paper are summarized as follows:

- We propose an AoI-aware multi-BS real-time monitoring system for large-scale IIoT access and formulate a dynamic stochastic optimization problem to minimize long-term average AoI under delay, bandwidth, computation

capacity, energy consumption, and task scheduling constraints.

- To effectively tackle this problem, we decompose it into task offloading and resource allocation subproblems. For task offloading, we design an innovative branching structure into the DRL framework and develop the Branching-D3DN algorithm, which reduces the network complexity from exponential to $\mathcal{Q}(n)$. The proposed Branching-D3DN algorithm alleviates the curse of dimensionality caused by large-scale IIoT devices and BS, enabling faster and more stable convergence.
- For resource allocation, we analyze the Hessian matrix of bandwidth and computational resources with respect to long-term average AoI and prove its semi-definite property. Building on this, we propose an iterative optimization algorithm for joint task offloading and resource allocation, guaranteeing efficient solutions to the minimization of the system's long-term average AoI.
- Extensive simulations demonstrate that the proposed Branching-D3DN algorithm significantly outperforms mainstream DRL algorithms, including DQN and D3QN, achieving at least a 75% improvement in convergence speed. Furthermore, compared to DQN, D3QN, Greedy algorithm, and random offloading, the proposed joint iterative optimization scheme achieves superior performance in device scalability and resource utilization, achieving up to at least 22% reduced average AoI performance.

The remainder of this paper is organized as follows. Sec. II reviews related works. Section III introduces the system model of the proposed AoI-aware multi-BS real-time monitoring system. Section IV formulates the dynamic stochastic optimization problem of minimizing the system's long-term average AoI. Section V provides the algorithm designs and solutions. Section VI provides extensive performance evaluations against the state-of-the-art benchmarks. Finally, Section VII concludes the paper and explores future directions.

II. RELATED WORKS

In this section, we first review the literature on AoI-based timeliness research concerning task offloading and resource allocation. Next, we provide a comprehensive survey of reinforcement learning (RL)–driven approaches in MEC environments. Finally, we identify key limitations from these studies that are closely aligned with the aforementioned challenges.

A. Research on AoI-based Timeliness for Task offloading and Resource Allocation

As a critical metric for evaluating the timeliness and promptness of wireless networks, AoI has been extensively applied in various network systems [19]. In [30], the authors analyzed the average AoI of MEC systems, where a BS generates and transmits computation-intensive packets to users. They studied the AoI performance of MEC systems under three computing schemes: local computing, edge computing and partial computing schemes. The authors in [41] considered expediting the computations of resource-intensive tasks by offloading them to the edge of the network, aiming to minimize the expected sum AoI for the MEC-assisted IoT network and provide mathematically traceable expressions for the AoI. In [42], the authors proposed a three-layer multi-unmanned aerial vehicle (UAV) assisted MEC system with stochastic task arrival, where the joint computation offloading, UAV trajectory and communication resources optimization scheme was developed to minimize the AoI of their proposed three-layer multi-UAV assisted MEC system. To enhance the trust, security, and efficiency within MEC environments, the authors in [43] explored a blockchain-based MEC model to guarantee secure task processing across diverse stakeholders and introduced the AoI as a metric to characterize the freshness of status reports. The authors in [22] used AoI metric to investigate the data freshness of the end-edge-cloud computing (EECC) framework and proposed joint task offloading, communication and computing resource allocation optimization scheme to minimize AoI and energy consumption under constraints of deadlines and capacity constraints. To meet the real-time demands of computation-intensive tasks, the authors in [23] adopted the AoI as the evaluation metric to determine it is suitable for making effective offloading decisions, and proposed the joint detection and computation offloading scheme to maximize the average computation rate of the edge system with the AoI constraints.

In recent years, artificial intelligence (AI) technologies represented by reinforcement learning (RL) have been widely applied in mobile networks, significantly promoting the advancement of IoT towards higher efficiency and intelligence [7, 8, 24, 44]. The authors in [10] jointly studied the optimal AoI-aware energy control and computation offloading problem. Then, the DRL techniques was applied to design an intelligent energy control and computation offloading algorithm for adapting to large-scale dynamic IIoT environments. Toward the rapid growth of the IIoT, the authors in [25] proposed a directed acyclic graph task model-based computation offloading method by modeling task dependencies, and formulated a joint latency, energy consumption, and AoI optimization problem, which is addressed by proposing an improved dueling double deep Q-network computation offloading algorithm. The authors in [45] studied the AoI and energy tradeoff (AET) problem in an aerial-ground collaborative MEC system, where the AET problem is formulated as the multi-objective optimization problem, which aims to simultaneously minimize the total AoI of ground devices and total energy consumption of the UAV by optimizing its flight paths and task offloading ratios. To address this problem, a multi-objective proximal policy optimization (PPO)-based learning algorithm was proposed. AoI is an important metric of measuring data freshness in UAV crowdsensing networks. The authors in [46] proposed a decentralized multi-agent DRL framework for multi-UAV trajectory planning to maximize the total amount of collected data from diverse Point-of-Interests while minimizing AoI and AoI threshold violation ratio under limited energy supplement. In order to enhance the data freshness, the authors in [47] minimized the AoI of the UAV systems by the UAV trajectory design and network configuration under the cost and practical constraints, where DRL was leveraged to help the UAV obtains the minimum AoI along with the optimal destined position, and

a soft actor-critic DRL method was developed for the HAPs completing the configuration of the coverage area and height of the HAPs and UAVs.

B. Limitations of the Existing Research Works

Although considerable progress has been made on AoI-aware MEC systems and RL-based optimization methods, existing research still faces several critical limitations when applied to large-scale IIoT access in multi-BS real-time monitoring systems. Firstly, most current AoI analyses heavily rely on overly simplified queueing-based model with static assumptions, which fails to comprehensively capture the stochastic and dynamic nature of real-time monitoring systems. Secondly, the majority of current studies concentrate on single-BS scenarios and thus fail to tackle the intricacies introduced by multi-BS scenarios, where large-scale IIoT access significantly amplifies the complexity of task offloading and resource allocation. Finally, despite the potential of DRL algorithms in handling MDP-based formulations, the most advanced DRL algorithms widely adopted, such as DQN, DDQN, and D3QN, suffer from poor scalability in high-dimensional action spaces, leading to exponential growth in computational burden and severe degradation in both efficiency and convergence stability.

These limitations highlight the pressing need for novel delay-sensitive IIoT networks that can achieve more accurate AoI modeling, scalable task offloading and resource allocation optimization schemes, and robust DRL-based algorithms tailored for AoI-aware multi-BS real-time monitoring system supporting large-scale IIoT access.

III. System Model

As shown in Fig. 1, we propose an AoI-aware multi-BS real-time monitoring system to support large-scale IIoT access. Each BS is equipped with a colocated MEC server, referred to as BS-MEC, to provide both communication and computational resources. The system comprises $N \geq 1$ single-antenna IIoT devices that can monitor and perceive the surrounding environment in real-time and generate update packets as needed. Given that these real-time monitoring systems emphasize the immediate processing of tasks, traditional delay-based metrics only consider the perspective of transmitting devices, failing to reflect the timeliness perceived at the receiving end [21]. Therefore, traditional delay evaluation metrics are no longer applicable to this scenario. To accurately evaluate real-time performance, we adopt the concept of AoI as the performance metric [19, 20].

We define the index set for BS-MEC as $\mathcal{M} = \{1, 2, \cdots, M\}$, the index set for IIoT devices as $\mathcal{N} = \{1, 2, \cdots, N\}$, and the priority of IIoT device $i \in \mathcal{N}$ as α_i. The task arrival process for each IIoT device follows a Bernoulli distribution. The system time is divided into T time slots, each of length τ_{max}, with the time slot index set represented as $\mathcal{T} = \{1, 2, \cdots, T\}$. The tasks for each IIoT device are independent and follow a passive sampling strategy, which generates status update packets (i.e., tasks) only when the environment changes. Assuming the task size collected by the IIoT device i is d_i and the CPU cycles required to complete the task are c_i, each IIoT device is equipped with a buffer queue to store the generated tasks. To ensure the real-time performance, newly generated tasks will be placed at the forefront of the buffer queue.

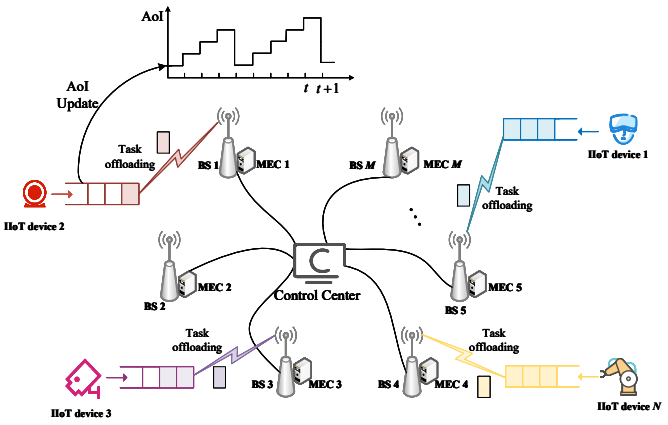


Fig. 1: The proposed AoI-aware multi-BS real-time monitoring system for large-scale IIoT access.

Next, we analyze the task offloading process in the proposed real-time monitoring system. To eliminate inter-cell interference, orthogonal frequency bands are allocated among different BSs. For mission-critical services, the data generated by a single IIoT device is typically small and requires limited spectrum resources. To further mitigate inter-user interference, we adopt orthogonal frequency division multiple access (OFDMA), allowing each IIoT device to access the system over orthogonal channels during transmission. The offloading matrix $\mathbf{X}$ can be given by Eq. (1), as follows:

$$\mathbf{X} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N] = \begin{bmatrix} \tilde{\boldsymbol{a}}_1 \\ \tilde{\boldsymbol{a}}_1 \\ \cdots \\ \tilde{\boldsymbol{a}}_N \end{bmatrix}. \tag{1}$$

This paper considers that each IIoT device can offload its task to at most one BS-MEC within a time slot t. Let $\tilde{a}_i$ denote the offloading vector of the IIoT device i, and $a_{i,j}^t$ be the binary decision variable indicating whether device $i \in \mathcal{N}$ offloads to BS $j \in \mathcal{M}$. The parameter ε represents the success probability of task offloading, with a corresponding transmission failure probability of $1 - \varepsilon$. During transmission process, the small-scale fading between each IIoT device and BS can be modeled as an exponential distribution with zero mean, while the large-scale fading follows $d_{i,j}^{-\tau}$, where $d_{i,j}$ represents the distance between IIoT device i and BS j, and τ represents the path-loss exponent. Accordingly, based on Shannon's Capacity, the transmission rate between IIoT devices i and BS j is given by

$$R_{i,j}(t) = B_{i,j}(t) \log_2 \left(1 + \frac{p_i(t) h_{i,j}(t)}{\sigma^2}\right), \tag{2}$$

where $B_{i,j}(t)$ represents the bandwidth allocated to IIoT device i during transmission, $p_i(t)$ denotes the transmission power of IIoT device i during transmission, $h_{i,j}(t)$ indicates the channel coefficient, and σ^2 is the noise power.

During task offloading, the total service time at the BS-

MEC consists of two components: (i) the wireless transmission delay over the air interface, and (ii) the computation delay incurred by task processing at the BS-MEC. Consequently, the transmission delay of IIoT device i can be expressed as

$$t_{i,j}^{trans} = \frac{d_{i,j}}{R_{i,j}(t)}. \tag{3}$$

Then, the energy consumption resulted by IIoT device i during wireless transmission can be given by

$$E_i(t) = \sum_{j=1}^{M} p_i(t) t_{i,j}^{trans}. \tag{4}$$

When BS-MEC receives the task from IIoT device i, the corresponding computation delay can be expressed as

$$t_{i,j}^{comp} = \frac{d_{i,j} c_i}{f_{i,j}(t)}, \tag{5}$$

where $f_{i,j}(t)$ denotes the computation resources allocated by BS-MEC j to IIoT device i. Accordingly, the total processing delay for IIoT device i can be represented by

$$D_i(t) = \sum_{j=1}^{M} a_{i,j}^t \left(t_{i,j}^{trans} + t_{i,j}^{comp} \right). \tag{6}$$

To evaluate the real-time performance of task transmission, we first analyze the AoI, which is defined as the time elapsed since the last valid state update was received and processed by the MEC-BS. In this context, AoI continues to increase until the computation of the next task is completed. Although existing studies extensively explore AoI through queuing theory, these studies typically rely on specific queuing models (e.g., M/M/1, D/M/1, etc.) and relevant prior knowledge regarding the distribution of random events [27, 28]. In real-time monitoring systems, task arrivals and service processes are inherently random and asynchronous, making it difficult to construct a general queuing model for accurate analysis. Unlike conventional approaches constrained by predefined models, we adopt AoI as a direct performance metric. This allows a more flexible and practical framework for analyzing and optimizing real-time task transmission under randomness. In the following, we further examine the dynamics of the proposed real-time monitoring system, as shown in Fig. 2.

Firstly (***Case 1***), if the task of IIoT device i has been successfully offloaded but not yet completed, or if the device remains in an unoffloaded or transmission failure state, the value of the AoI in the next time slot can be denoted as

$$A_i^{t+1} = A_i^t + \tau_{max}, \tag{7}$$

where the AoI at the time slot $t+1$ is simply the previous AoI plus the slot length τ_{max}.

Secondly (***Case 2***), if the task has been offloaded and completed, the AoI decreases at the next time slot $t+1$, since the device obtains a fresh update only after the task processing is completed. In this case, both transmission and computation delays are included before the update is applied, thus the AoI can be given by

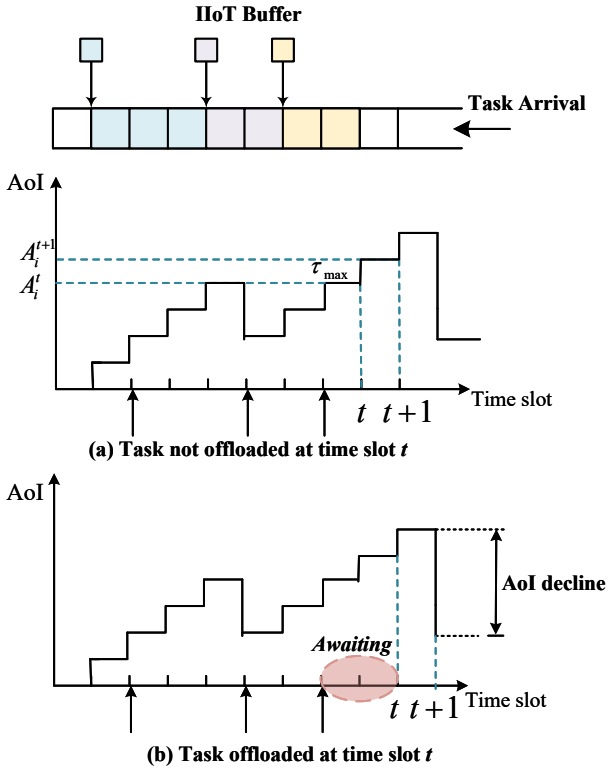


Fig. 2: The AoI changes of the proposed real-time monitoring system.

$$A_i^{t+1} = t - G_i^t + t_{i,j}^{trans} + t_{i,j}^{comp}, \tag{8}$$

where t denotes the current time, G_i^t indicates the generation time of the task, $t_{i,j}^{trans}$ and $t_{i,j}^{comp}$ represents the transmission and computation delays, respectively. Accordingly, the AoI evolution of device i under different cases can be summarized as (9), and the overall trend of AoI dynamics can be illustrated in Fig. 2.

$$A_i^{t+1} = \begin{cases} A_i^t + 1, & \text{Case 1;} \\ t - G_i^t + t_{i,j}^{trans} + t_{i,j}^{comp}, & \text{Case 2.} \end{cases} \tag{9}$$

IV. Problem Formulation

To effectively optimize the timeliness of the proposed AoI-aware multi-BS real-time monitoring system, we formulate a minimization problem for the long-term average AoI performance, subject to strict constraints on delay, bandwidth, computation capacity, energy consumption, and task scheduling. The optimization problem can be expressed as follows:

$$\mathcal{P}1: \min_{\{B_{i,j}(t), f_{i,j}(t), a_{i,j}^t\}_{\forall i \in \mathcal{N}, j \in \mathcal{M}}} \lim_{T \to \infty} \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{t=1}^{T} \alpha_i A_i^t, \tag{10a}$$

$$\text{s.t.} \quad \sum_{j=1}^{M} a_{i,j}^t \leq 1, a_{i,j}^t \in \{0,1\}, \quad \forall i \in \mathcal{N}, \tag{10b}$$

$$\sum_{i=1}^{N} a_{i,j}^t \leq K, \quad \forall j \in \mathcal{M}, \tag{10c}$$

$$E_i(t) \leq E_{max}, \quad \forall i \in \mathcal{N}, \tag{10d}$$

$$\sum_{i=1}^{N} B_{i,j}(t) \leq B_j^{max}, \quad \forall j \in \mathcal{M}, \tag{10e}$$

$$\sum_{i=1}^{N} f_{i,j}(t) \leq f_j^{max}, \quad \forall j \in \mathcal{M}, \tag{10f}$$

$$D_i(t) \leq \tau_{max}, \quad \forall i \in \mathcal{N}, \tag{10g}$$

where constraint (10b) specifies that $a_{i,j}^t$ is a binary variable, guaranteeing that each IIoT device can only offload its task at most one BS-MEC, which also imposes a limit on the number of accessing devices. Constraint (10d) imposes the energy consumption limits on each IIoT device, which is particularly critical since IIoT devices are battery-powered. Constraints (10e) and (10f) restrict the total allocated bandwidth and computation resource, respectively. Constraint (10g) guarantees that the task transmission and computation must be completed within one time slot.

Observing problem $\mathcal{P}1$, both integer variables $a_{i,j}^t$ and continuous variables $B_{i,j}(t)$ and $f_{i,j}(t)$ are involved. The complex coupling between these discrete and continuous variables renders $\mathcal{P}1$ a non-convex dynamic stochastic optimization problem, resulting in the traditional convex optimization schemes being suitable for solving $\mathcal{P}1$.

V. Algorithm Designs and Solutions

The primary challenges in solving this long-term dynamic stochastic optimization problem lies in the fact that the AoI depends on not only the current state but also on prior states. Furthermore, the optimization objective of $\mathcal{P}1$ involves both discrete offloading decisions and continuous variables, which significantly increases the problem's complexity. To simplify the solving process, we decouple the original problem into two subproblems: the task offloading subproblem and the resource allocation problem.

A. The Solution for Task Offloading Problem

To effectively tackle the task offloading subproblem, we first reformulate it as an equivalent Constrained Markov Decision Process (CMDP). The current state is influenced by past states and also determines future states. The state in a MDP can be defined as a multidimensional array, denoted as $[\mathbb{A}, \mathbb{S}, \mathbf{R}, \mathbf{P}, \boldsymbol{\pi}]$, and the meanings of each element in the array are as follows:

1) $\mathbb{A}$**: Action Space** represents the set of the interactions between BS-MEC and IIoT devices, consisting of the offloading variables $a_{i,j}^t$. In this case, the action space $\mathbb{A}$ can be expressed as the offloading matrix $\mathbf{X}$.
2) $\mathbb{S}$**: State Space** comprises four components, as follows:
   - The AoI information of all IIoT devices at time slot t, denoted as $\boldsymbol{A}(t) = [A_1^t, A_2^t, \cdots, A_N^t]$.
   - The energy consumption information of all IIoT devices at time slot t, denoted as $\boldsymbol{E}(t) = [E_1(t), E_2(t), \cdots, E_N(t)]$.
   - The delay information of all IIoT devices at time slot t, denoted as $\boldsymbol{D}(t) = [D_1(t), D_2(t), \cdots, D_N(t)]$.
   - Channel state information of the considered monitoring system $\boldsymbol{H}(t) = \{h_{i,j}(t)\}_{i\in\mathcal{N}, j\in\mathcal{M}}$ at time slot $t \in \mathcal{T}$.
3) **R: Reward function** is determined by both the action and state at time slot t. To align the reward with the objective of minimizing the system's long-term average AoI performance, we define the reward function as the negative weighted sum of AoI values:

$$r(t) = -\sum_{i=1}^{N} \alpha_i A_i^t. \tag{11}$$

4) **P: State transition probability** refers to the probability of selecting an action to transition to the next state $s(t+1)$ under the current state $s(t)$.
5) $\boldsymbol{\pi}$**: Strategy** denotes the state update strategy over the time slot set $\mathcal{T}$, defined as $\theta = \{\pi_1, \cdots, \pi_T\}$, and the optimal strategy for the proposed real-time monitoring system is given by

$$\pi^* = \arg\min_{\boldsymbol{\pi}} \frac{1}{T} \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=1}^{T} c(t) | s(1) \right]. \tag{12}$$

Since this problem is constructed as a CMDP, directly tackling the CMDP is typically intractable. To effectively solve this, we transform CMDP into an unconstrained MDP by introducing relaxation terms into the reward function (11), thereby eliminating the influence of constraint conditions. This method balances the original objectives while implicitly enforcing the constraints, resulting in a new unconstrained optimization problem. Specifically, the modified reward function can be given as follows:

$$r(t) = -\sum_{i=1}^{N} \Big[\alpha_i A_i^t - \zeta \left(E_{max} - E_i(t) \right) - \beta \left(\tau_{max} - D_i(t) \right) \Big]. \tag{13}$$

Solving CMDP is equivalent to solving the unconstrained MDP. Therefore, we propose to employ DRL to tackle it, which focuses on maximizing the long-term cumulative utility of the system. In DRL, the utility is typically represented by the Q-value, also known as the action value function, which estimates the expected return of taking a specific action and following a certain strategy under a given state. The Bellman equation establishes a recursive relationship between the value of an action at time slot t and the expected value of future actions, enabling the recursive calculation of the Q-value. Formally, the specific expression of the Bellman equation can be denoted as

$$Q(s_t, a_t) = r(t) + \gamma \mathbb{E} \left[Q(s_{t+1}, a_{t+1}) \right], \tag{14}$$

where $\gamma \in [0, 1]$ denotes the discount factor that balances immediate and future rewards, and smaller values of γ emphasize short-term gains, whereas $\gamma \to 1$ prioritize long-term returns. Here, s_t denotes the current state, a_t the selected action, and $r(t)$ the reward signal observed after executing the action a_t, with the environment transitioning to the next state s_{t+1}.

For the considered task offloading, the action space $\mathbb{A}$ consists of discrete binary offloading variables. For such discrete-

action problem, the deep Q-network (DQN) algorithm is a widely adopted solution, combining reinforcement learning with deep neural networks (DNNs) to approximate the Q-table of traditional Q-learning [32]. This enables handling of high-dimensional state spaces. In this setting, each IIoT device is modeled as an agent, and the principles and workflow of DQN can be summarized as Fig. 3.

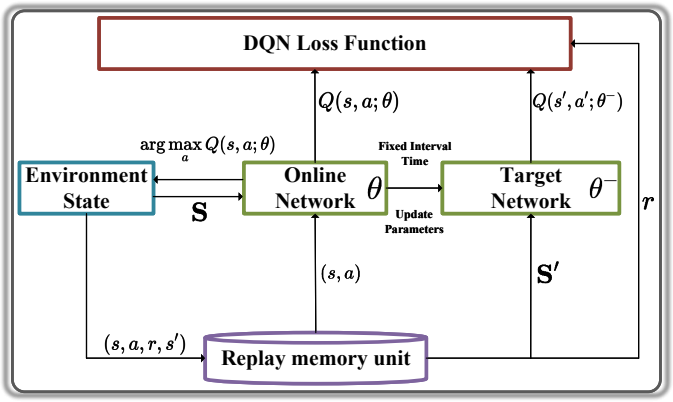


Fig. 3: The basic framework of DQN framework [32].

As illustrated in Fig. 3, the DQN architecture consists of two key components: the online network and the target network, with their parameters θ and θ^-, respectively. The online network estimates Q-value based on the current state $\mathbf{S}$, and its parameters are updated after each step by learning from samples (s,a) in the pool of relay memory unit. On the other hand, the target network, updated only at regular intervals by copying parameters from the online network, stabilizes the training process by mitigating target value fluctuations. The design of this decouple network structure can significantly improve the problem of target movement during the learning process, enhancing the learning and training stability [32]. The loss function of DQN is defined as mean squared error (MSE) loss between the predicted Q-value $Q(s,a;\theta)$ and the target Q-value $Q(s',a';\theta^-)$, which is given by [48]

$$L(\theta) = \mathbb{E}\left[\left(r(t) + \gamma \max_{a'} Q(s',a';\theta) - Q(s,a;\theta)\right)^2\right]. \quad (15)$$

Although DQN has significantly advanced reinforcement learning, it suffers from overestimation and maximization bias since the the same network is used for action selection and evaluation, leading to overestimation of the maximum Q value in certain states. Double DQN (DDQN) improves this by decoupling the two processes: (i) the online network selects the action with the highest Q-value, while (ii) the target network evaluates it, leveraging its stability to reduce bias [35]. For the DDQN network, its loss function can be expressed as

$$L(\theta) = \mathbb{E}\left[\left(r(t) + \gamma Q^{-1}\left(s', \arg\max_{a'} Q(s',a';\theta);\theta\right) - Q(s,a;\theta)\right)^2\right]. \quad (16)$$

Although DDQN alleviates overestimation problems compared to traditional DQN, it struggles to effectively distinguish actions with highly similar or overlapping effects, especially in environment states with noise or small differences in rewards. Moreover, its reliance on a single state representation may also fail to capture all factors influencing decisions, limiting the ability to learn optimal policies. To overcome these limitations, Dueling DQN decomposes the Q-function into a state value function $V(s)$ and an advantage function $A'(s,a)$, rewritten as $Q(s,a) = V(s) + A'(s,a)$, with two parallel output structures parameterized by ϕ and δ, respectively [36, 37]. However, directly summing V and A' can obscure their individual contributions, reducing network performance. To address this, a modified Q-function with a penalty term is introduced [37]. Denoting the set of devices offloading tasks at time slot t as $B(t)$, the modified Q-function is reformulated as follows:

$$Q(s,a;\theta,\delta,\phi) = V(s;\delta,\lambda) + \left[A'(s,a;\theta,\phi) - \frac{1}{|B(t)|}\sum_{a^*\in B(t)} A'(s,a^*;\theta,\phi)\right], \quad (17)$$

where V and A' are relatively fixed and unchanged with different inputs when Q is determined, thereby solving the problem of recognizability.

Dueling DQN and DDQN complement each other. On the one hand, Dueling DQN separates the network architecture into an estimated state value function and an estimated advantage function for each action, improving Q-value estimation and enabling finer discrimination between actions with similar outcomes. On the other hand, DDQN reduces overestimation by using two networks to evaluate action selection (online network) and action evaluation (target network) separately, thus reducing bias inherent in traditional DQN. Combining the architectural advantages of Dueling DQN with the method of reducing overestimation in DDQN, the Dueling Double DQN (D3QN) algorithm is proposed [38–40]. D3QN alleviates overestimation, while more accurately estimating the state-action values, making it well-suited for the task offloading problem considered here and theoretically closer to the optimal solution. Its loss function follows the DDQN form, while Q-value calculations adopt the Dueling architecture, as shown in (17). In (17), θ^-, δ^-, and ϕ^- represent the target network parameters, θ, δ, and ϕ denote the online network parameters, s' and a' indicate the next-state observation and action. The target network parameters θ^-, δ^-, and ϕ^- are periodically updated by copying the corresponding online network parameters. However, D3QN alone cannot fully address the considered task offloading subproblem, as the actions space grows exponentially with the number of IIoT devices and BSs, significantly slowing learning and convergence. This curse of dimensionality poses a major challenge in reinforcement learning and limits the scalability of D3QN for large-scale systems. The D3QN architecture tailored for the task offloading problem is illustrated in Fig. 4.

To address the issue of exponential growth in the action space dimensionality, this paper proposes an innovative enhancement to the D3QN algorithm by incorporating a branch-

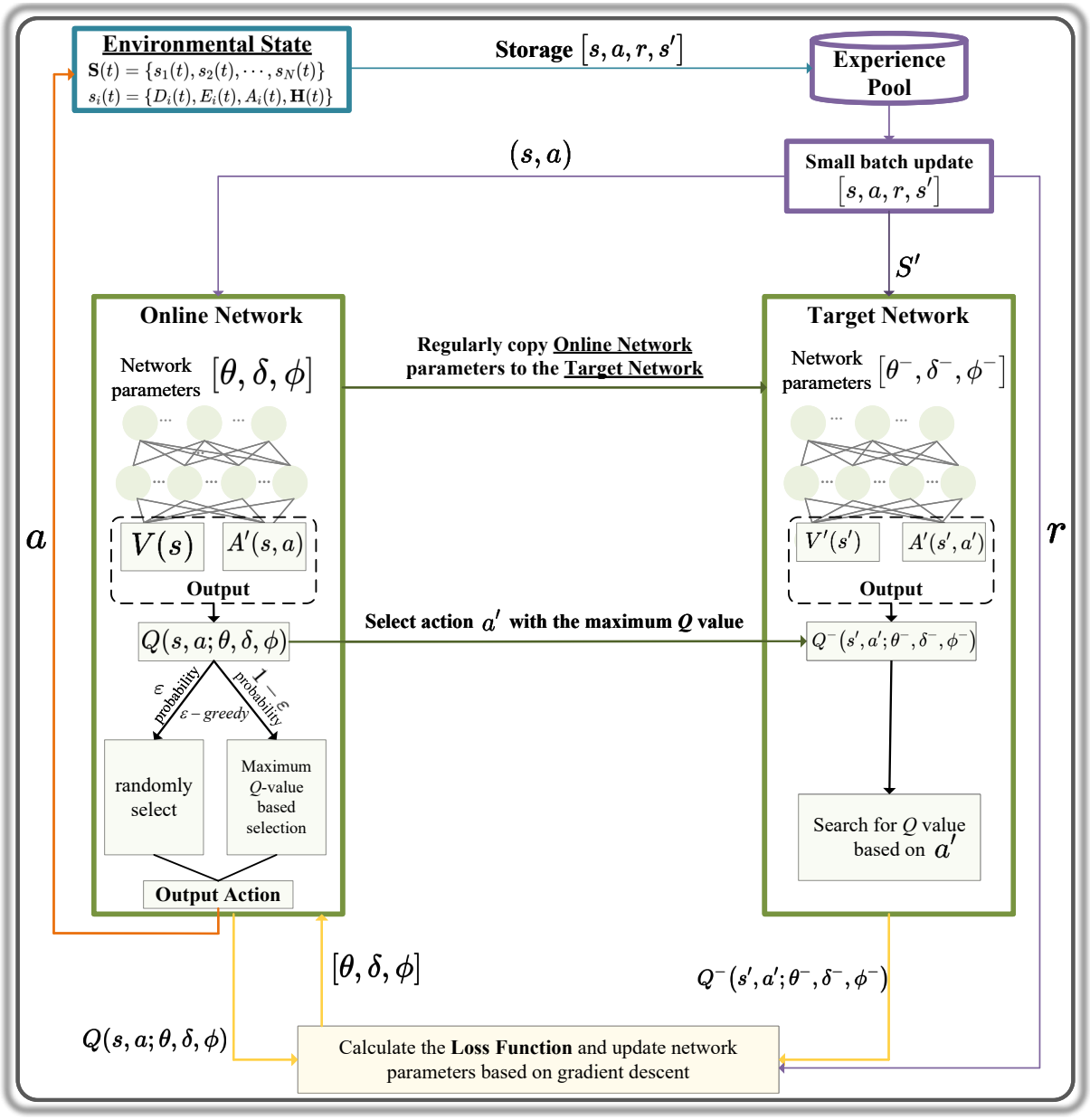


Fig. 4: The basic framework of D3QN framework.

ing structure, resulting in the Branching D3QN (BD3QN) algorithm. As illustrated in Fig. 5, the proposed BD3QN can significantly mitigate the complexity of the decision-making process by decomposing the high-dimensional action space into smaller, more manageable subspaces. The core innovation of BD3QN lies in its ability to treat different features (or dimensions) of each action as independent branches, with each branch optimizing decisions specific to its corresponding feature. This architectural design not only reduces the number of output variables per network but also enhances the efficiency of the learning process by parallelizing the optimization across multiple subspaces. Consequently, the BD3QN algorithm effectively addresses the curse of dimensionality, enabling scalable and faster convergence in systems with a large number of IIoT devices and BSs.

In particular, when applied the BD3QN algorithm to task offloading, the task offloading decision of each IIoT device is regarded as an independent branch, resulting a total of N branches, and each branch has $M+1$ possible actions, that is selecting one of the BSs to offload or not. The branching structure allows our proposed BD3QN algorithm to optimize the offloading decisions of each device in parallel, effectively addressing the problem of expanding action space caused by an increase in the number of IIoT devices and BSs. As a result, the time complexity of the solution becomes $N(M+1)+1$. Moreover, the detailed branch structure can be illustrated in Fig. 5. After adopting the branch structure, the Q value for each sub-branch (taking here the branch n as an example) can be denoted as

$$Q(s, a_n; \theta, \delta, \phi) = V(s; \theta, \delta) + \Big[A'(s, a_n; \theta, \phi) - \frac{1}{M+1} \sum_{a_n^* \in \mathcal{A}_n} A'(s, a_n^*; \theta, \phi)\Big]. \tag{20}$$

$$L\left(\theta,\delta,\phi\right)=\mathbb{E}\left[\left(r_t+\gamma Q^{-}\left(s',\arg\max_{a'}Q^{-}(s',a';\theta,\delta,\phi);\theta^{-1},\delta^{-1},\phi^{-1}\right)-Q(s,a;\theta,\delta,\phi)\right)^2\right]. \tag{18}$$

$$L\left(\theta,\delta,\phi\right)=\mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N}\left(r_t+\gamma\max_{n}Q_n^{-}\left(s_{t+1},\arg\max_{a'_n\in\mathcal{A}_n}Q_n(s_{t+1},a'_n;\theta,\delta,\phi);\theta^{-1},\delta^{-1},\phi^{-1}\right)-Q_n(s,a_n;\theta,\delta,\phi)\right)^2\right]. \tag{19}$$

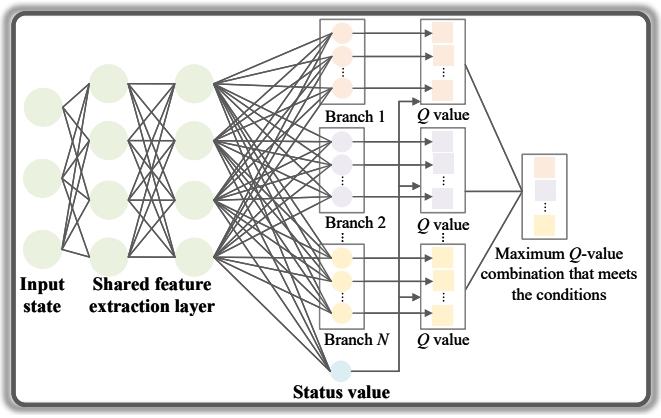


Fig. 5: The proposed branching structure for D3QN algorithm.

In this case, the loss value of BD3QN can be reformulated as (19), and the specific algorithm process can be detailed in **Algorithm 1**.

B. The Solution for Resource Allocation Problem

Based on the task offloading strategies $a_{i,j}^t$ obtained by **Algorithm 1**, the original problem $\mathcal{P}1$ can be transformed into $\mathcal{P}\in$, as follows:

$$\mathcal{P}2:\min_{\{B_{i,j}(t),f_{i,j}(t)\}_{\forall i\in\mathcal{N},j\in\mathcal{M}}}\lim_{T\to\infty}\frac{1}{T}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{T}\alpha_i A_i^t, \tag{21a}$$

$$s.t.\quad E_i(t)\le E_{max},\quad \forall i\in\mathcal{N}, \tag{21b}$$

$$\sum_{i=1}^{N}B_{i,j}(t)\le B_j^{max},\quad \forall j\in\mathcal{M}, \tag{21c}$$

$$\sum_{i=1}^{N}f_{i,j}(t)\le f_j^{max},\quad \forall j\in\mathcal{M}, \tag{21d}$$

$$D_i(t)\le\tau_{max},\quad \forall i\in\mathcal{N}. \tag{21e}$$

We decompose $\mathcal{P}2$ into a resource allocation decision-making process with granularity for each time slot, as follows:

$$\mathcal{P}3:\min_{\{B_{i,j}(t),f_{i,j}(t)\}_{\forall i\in\mathcal{N},j\in\mathcal{M}}}\sum_{i=1}^{N}\sum_{j=1}^{M}\alpha_i A_i^t, \tag{22a}$$

$$s.t.\quad (21b),(21c),(21d),\text{and }(21e). \tag{22b}$$

Expanding the objective function (22a), and $\mathcal{P}\ni$ can be equivalently transformed into $\mathcal{P}4$, as follows:

$$\mathcal{P}4:\min_{\{B_{i,j}(t),f_{i,j}(t)\}_{\forall i\in\mathcal{N},j\in\mathcal{M}}}\sum_{i=1}^{N}\alpha_i\left\{\sum_{j=1}^{M}a_{i,j}^t\left(t-G_i^t+\frac{d_i}{B_{i,j}(t)\log_2\left(1+\frac{p_i(t)h_{i,j}(t)}{\sigma^2}\right)}+\frac{d_ic_i}{f_{i,j}(t)}\right)\right\}, \tag{23a}$$

$$s.t.\quad (21b),(21c),(21d),\text{and }(21e).$$

Next, we demonstrate through Theorem 1 that $\mathcal{P}4$ is a strictly convex optimization problem, as follows:

Theorem 1. *For problem $\mathcal{P}4$, the objective function is convex regarding the resource allocation variables $B_{i,j}(t),f_{i,j}(t)_{\forall i\in\mathcal{N},j\in\mathcal{M}}$, and all its constraints are convex. Therefore, $\mathcal{P}4$ is a strictly convex optimization problem.*

Proof. The objective function (23) of $\mathcal{P}4$ is denoted as F. To demonstrate its convexity, we derive the first and second derivatives of F with respect to the variables $B_{i,j}(t)$ and $f_{i,j}(t)$.

Firstly, for $f_{i,j}(t)$:

$$\nabla F_{\{f_{i,j}(t),\forall i\in\{i|a_{i,j}^t=1\}\}}=-\frac{\alpha_i d_i c_i}{(f_{i,j}(t))^2}<0, \tag{24}$$

$$\nabla^2 F_{\{f_{i,j}(t),\forall i\in\{i|a_{i,j}^t=1\}\}}=\frac{2\alpha_i d_i c_i}{(f_{i,j}(t))^3}>0, \tag{25}$$

Next, for $B_{i,j}(t)$:

$$\nabla F_{\{B_{i,j}(t),\forall i\in\{i|a_{i,j}^t=1\}\}}=\frac{-\alpha_i d_i}{(B_{i,j}(t))^2\log_2\left(1+\frac{p_i(t)h_{i,j}(t)}{\sigma^2}\right)}<0, \tag{26}$$

$$\nabla^2 F_{\{B_{i,j}(t),\forall i\in\{i|a_{i,j}^t=1\}\}}=\frac{2\alpha_i d_i}{(B_{i,j}(t))^3\log_2\left(1+\frac{p_i(t)h_{i,j}(t)}{\sigma^2}\right)}>0. \tag{27}$$

Since F is separable in terms of $B_{i,j}(t)$ and $f_{i,j}(t)$, the mixed partial derivatives are zero. Hence, the Hessian matrix of F is given by:

$$\mathbf{H}(\boldsymbol{f},\boldsymbol{B})=\begin{bmatrix}\frac{2\alpha_i d_i c_i}{(f_{i,j}(t))^3} & 0\\ 0 & \frac{2\alpha_i d_i}{(B_{i,j}(t))^3\log_2\left(1+\frac{p_i(t)h_{i,j}(t)}{\sigma^2}\right)}\end{bmatrix} \tag{28}$$

Given that the diagonal elements of the Hessian matrix are all positive for $\boldsymbol{f}>0$ and $\boldsymbol{B}>0$, this implies that the matrix is strictly diagonally dominant. Therefore, the Hessian matrix

is semi-positive definite, which confirms that the objective function F is convex over the domain $\boldsymbol{f} > 0$ and $\boldsymbol{B} > 0$. Consequently, $\mathcal{P}4$ is a strictly convex optimization problem for the variables $\boldsymbol{f}$ and $\boldsymbol{B}$. □

According to **Theorem 1**, the problem $\mathcal{P}3$ can be easily tackled using the CVX tool.

Algorithm 1: The Branching-D3QN-based Task Offloading Algorithm.

Input: IIoT device set $\mathcal{N}$, BS set $\mathcal{M}$, channel status of devices h_n, delay threshold τ_{max}, energy constraint of device E_{max}, AoI weights of devices α_i, parameter update cycle $\mathbf{\Gamma}$;

Initialize: Environment status $\boldsymbol{S}(t)$, parameters of online network $[\theta, \delta, \phi]$, and target network $\left[\theta^-, \delta^-, \phi^-\right]$, ε, and ε_{min};

for $t = 0, 1, \cdots, T$ **do**
- Generate a random number $\kappa \in (0, 1)$;
- **if** $\kappa \geq \varepsilon$ **then**
  - Select the action combination with the maximum Q value;
  - //**Calculate Q value and select the optimal action combination.**
- **else**
  - Select actions randomly;
- **end**
- //**Update the AoI of each IIoT device.**
- Update AoI for devices with action changes according to (9)-Case 2;
- Update AoI for other devices without action changes based on (9)-Case 1;
- **if** Devices with unchanged actions AoI $\geq$ AoI$_{max,i}$ **then**
  - Update the AoI of this device to AoI$_{max,i}$;
  - Increase the selection probability of this device;
- **end**
- Execute action $a(t)$, observe reward function $r(t)$ and next state $s(t+1)$;
- Store the combination $[s(t), r(t), s(t+1)]$ into the experience pool;
- Randomly sample a batch from the experience pool for training;
- Calculate the Loss function based on Eq. (19);
- //**Update the online network parameters.**
- Update online network parameters $[\theta, \sigma, \phi]$ via using gradient descent method;
- //**Update the target network parameters.**
- **if** Training Steps $\%\Gamma == 0$ **then**
  - Copy the online network parameters to the target network;
  - Update target network parameters: $\theta^- = \theta$, $\sigma^- = \sigma$, and $\phi^- = \phi$;
- **end**

end

Output: Optimal task offloading strategy $\boldsymbol{\pi}^*$.

C. Joint Task Offloading and Resource Allocation Scheme

In Sections V-A and V-B, the original problem was decoupled into two subproblems: the task offloading problem and the resource allocation problem, with corresponding optimization schemes developed for each. To efficiently solve the original joint problem, we adopt an alternating optimization (AO) framework, iteratively applying the proposed BD3QN algorithm for task offloading and the resource allocation algorithm until the system converges to a stable state. The detailed procedure is summarized in **Algorithm 2**.

Algorithm 2: AO-Based Joint Task Offloading and Resource Allocation Scheme

Input: IIoT device set $\mathcal{N}$, BS set $\mathcal{M}$, device channel states h_n, delay threshold $\tau_{\max}$, energy limit $E_{\max}$, AoI weights α_i, parameter update cycle $\mathbf{\Gamma}$.

Output: Optimized strategies $\left\{B^*_{i,j}(t), f^*_{i,j}(t), a^*_{i,j}(t)\right\}$ for all $i \in \mathcal{N}, j \in \mathcal{M}, t \in \mathcal{T}$.

Initialize:
- Environment state $\boldsymbol{S}(t)$;
- Online network parameters $[\theta, \delta, \phi]$ and target network parameters $\left[\theta^-, \delta^-, \phi^-\right]$;
- Initial feasible resource allocation and offloading scheme $\{B_{i,j}(t), f_{i,j}(t), a_{i,j}(t)\}$ satisfying constraints $(10b)$-$(10g)$.

repeat
- Execute **Algorithm 1** to update task offloading strategy $\{a_{i,j}(t)\}$;
- Solve subproblem $\mathcal{P}4$ via convex optimization (e.g., CVX) to update $\{B_{i,j}(t), f_{i,j}(t)\}$;
- Update the joint strategy $\{B_{i,j}(t), f_{i,j}(t), a_{i,j}(t)\}$ for the next iteration;

until *convergence or maximum iterations reached*;

VI. Performance Evaluation

In this section, we conduct extensive simulation experiments to rigorously evaluate the effectiveness of the proposed schemes. Specifically, we benchmark the proposed BD3QN algorithm against both classical baselines and state-of-the-art algorithms, and provide detailed comparative analyses. Furthermore, comprehensive ablation studies are performed to verify the contribution of each key component and to demonstrate the overall superiority of our proposed schemes.

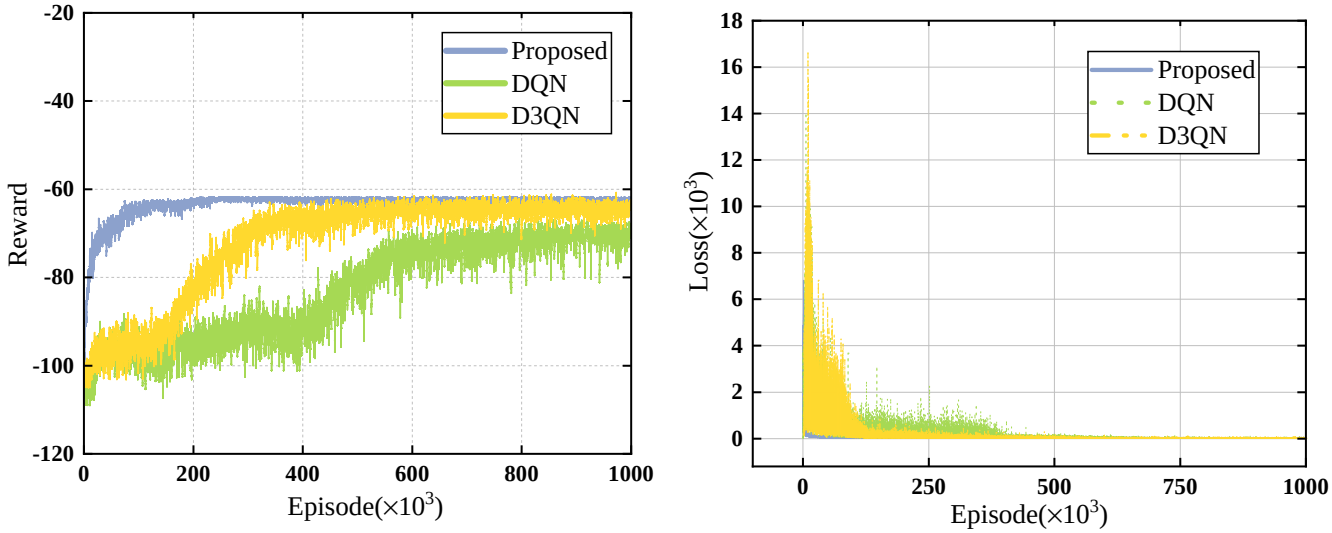
A. Experimental Parameter Settings

In our simulations, the IIoT devices and BSs are randomly distributed within a $1000m \times 1000m$ rectangular region. All experiment are executed on servers equipped with Nvidia RTX 3090Ti GPU to guarantee efficient training and evaluation. The simulation experiment is implemented in Python 3.9, while the training process leverages TensorFlow 2.0. The detailed simulation parameters used throughout our performance evaluation are summarized in Table I.

B. Baseline Schemes for Comparison

To comprehensively evaluate the effectiveness of the proposed BD3QN algorithm and joint task offloading and resource allocation scheme (denoted as ***Proposed***), we compare it with several representative baseline schemes, including classical heuristics and state-of-the-art DRL-based algorithms:

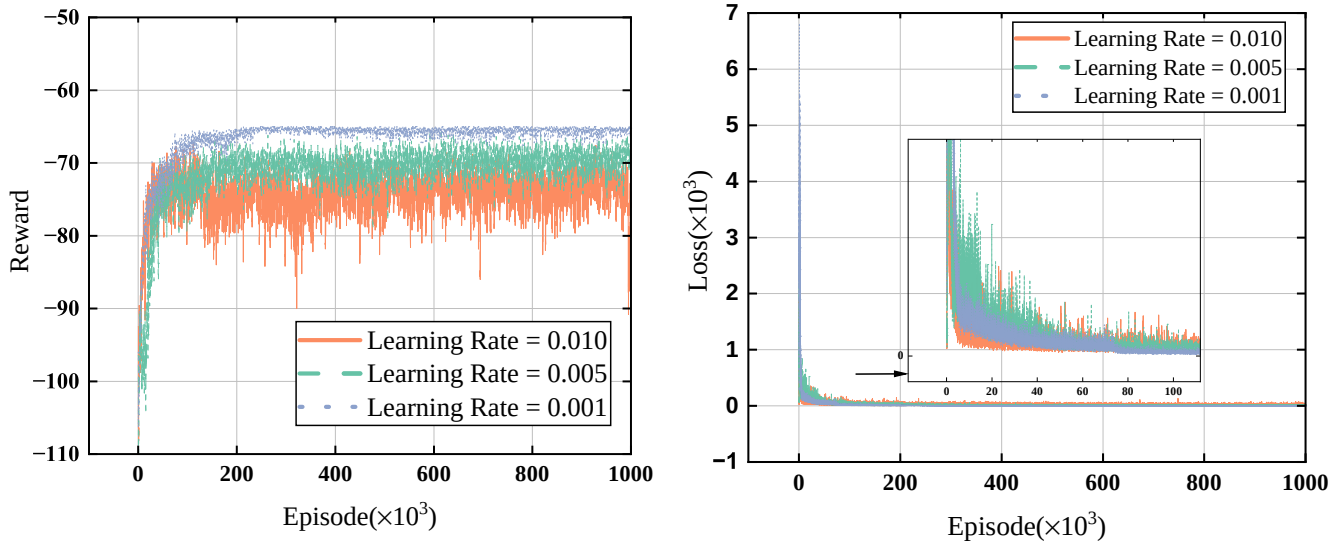
- **Greedy Algorithm** [49]: A heuristic method that prioritizes IIoT devices with higher AoI values for task offloading, while allocating idle BSs accordingly to minimize the overall AoI.
- **Random Offloading**: A benchmark strategy in which idle BSs randomly select IIoT devices for task offloading, serving as a performance lower bound.



(a) Reward changes over episode.

(b) Loss changes over episode.

Fig. 6: Performance Comparison of the proposed Branching D3QN Algorithm with DQN and D3QN. (a) Reward changes over episode. (b) Loss changes over episode.



(a) Reward changes over episode.

(b) Loss changes over episode.

Fig. 7: The convergence performance verification of the proposed Branching-D3QN algorithm at different learning rates. (a) Reward changes over episode. (b) Loss changes over episode.

- **DQN Algorithm** [32, 48]: A DRL-based method that employs the standard DQN architecture to optimize the long-term average AoI without considering overestimation or state–action value decomposition.
  D3QN Algorithm [38–40]: An enhanced DRL approach that integrates DDQN and dueling DQN network architectures to mitigate overestimation and improve state–action value estimation for long-term average AoI minimization.

C. Experimental Results

1) Convergence Verification: We first validate the convergence performance of the proposed BD3QN algorithm and benchmark it against the popular DQN and D3QN algorithms. Convergence is a prerequisite for ensuring the stability and reliability of reinforcement learning models. In this experiment, the average reward is adopted as the evaluation metric. In reinforcement learning, the definition of an episode is usually defined as a complete sequence of interactions between the agent and the environment until reaching a terminal state, and this process can be expressed as a complete sequence of <state–action–reward>, denoted as $< s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_n, a_n, r_n >$. Each episode represents a complete instance of the task from the start to the end, allowing the agent to learn and adapt to the environment while attempting.

As illustrated in Fig. 6 (a) and (b), the proposed BD3QN algorithm demonstrate a significantly faster and more stable convergence behavior compared with both DQN and D3QN algorithms. It can be observed that the proposed BD3QN

TABLE I: Simulation Parameter Settings

Parameter	Value
System bandwidth B	400 kHz
Limitation of each BS's device access number K	3
Large-scale fading model $d_{i,j}^{\tau}$	$128.1 + 37.6\log(d_{i,j}[\text{km}])$
Noise power spectral density N_0	-174 dBm/Hz
Task package size d_i	$[1, 3]$ Mbits
Maximum delay constraint τ_{max}	1 s
MEC server computing capability f_j^{max}	$[7, 10]$ Gbps
Maximum power limit of IIoT devices $p_i^{\max}$	27.8 dBm/Hz
Maximum energy limit of IIoT devices E_{max}	$[0.5, 1.5]$ J
Update steps Γ	100
Pool size	50000
Batch size $\lvert\mathcal{B}\rvert$	64
Learning rate	64
Decay rate of learning rate	0.95
Decay steps of learning rate	10000
Activation function	ReLU
Optimizer	Adam
Episode	1000000
Discount-related coefficients ς, β	[0.8, 0.95]

algorithm converges at least 75% faster in terms of both reward and loss, clearly highlighting its efficiency advantage. This remarkable improvement is attributed to our proposed branching structure, which reduces the action space complexity from exponential level $(M+1)^N$ to linear level $(M+1)(N+1)$, thereby significantly alleviating the curse of dimensionality and accelerating convergence. Meanwhile, D3QN shows better convergence than DQN due to its dueling architecture, which effectively distinguishes between state values and action advantages, leading to more accurate Q-value estimation. Nevertheless, its performance still lags far behind the proposed BD3QN algorithm. These results collectively confirm that the proposed algorithm not only achieves faster convergence but also maintains superior stability throughout training.

Furthermore, Fig. 7 illustrates the impact of learning rate on the convergence behavior of the proposed BD3QN algorithm. These experimental results demonstrate that the learning rate critically influence both convergence speed and stability. A relatively small learning rate allows for fine-grained parameter updates, leading to stable and precise convergence, but requires more episodes to reach steady performance. Conversely, an excessively high learning rate accelerates early convergence but often induces oscillations near the optimum or even divergence, thereby degrading overall performance. This observation underscores the inherent trade-off between convergence speed and stability in learning rate selection. In extreme cases, an overly small learning rate may also trap the algorithm in local optima, preventing convergence within a limited number of training episodes.

2) Performance Comparison: Next, we comprehensively compare the proposed BD3QN algorithm with baseline schemes in terms of average AoI performance. In DRL methods, it is common to observe the impact of different batch sizes on the performance and training efficiency of the model by changing the batch size during the training process [1].

As shown in Fig. 8, we analyze the impact of batch size on the average AoI under varying numbers of IIoT devices. Firstly, it can be observed that under the same Batchsize, the more IIoT devices accessed, the worse the average AoI performance of the system. Moreover, the results show that increasing batch size initially improves performance by reducing gradient variance and making updates closer to the optimal direction. However, once the gradient estimation becomes sufficiently accurate, further increases yield diminishing returns and may even restrict parameter exploration, reducing adaptability. In practice, batch sizes of 64 or 128 strike a good balance between training efficiency and model performance, making appropriate batch size selection crucial for effective model training.

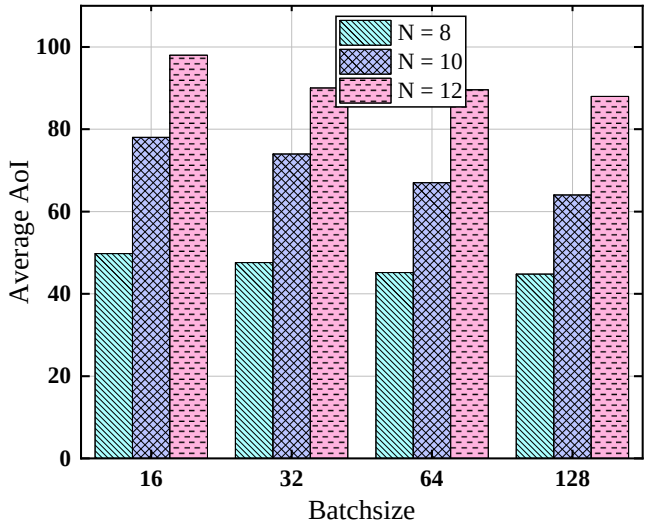


Fig. 8: The variation of average AoI with Batchsize size under varied number of IIoT devices.

In real-time monitoring systems, the task arrival rate λ $(0 \leq \lambda \leq 1)$ critically influences AoI performance. The task arrival rate quantifies the frequency of new task generation per unit time, where higher values accelerate task generation, thereby increasing device update frequency. In this case, the system can process and respond to the latest data faster, further optimizing system performance and user experience. Conversely, a lower λ reduces update frequency, leading to relatively outdated information on the device, increased average AoI, and diminished the timeliness and efficiency of the system. As shown in Fig. 9, we conduct experiments to demonstrate that average AoI decreases with the increase of λ since frequent data generation maintains data freshness at device nodes. Furthermore, it can be observed that the proposed BD3QN algorithm significantly outperforms benchmarks, achieving the 2% average AoI reduction over D3QN, 10.4% over DQN, and 17.4% over the Greedy algorithm at $\lambda = 0.8$, underscoring the superior performance gains of the proposed algorithm compared to other baseline schemes in terms of high task arrival rates, data freshness, and system efficiency.

As illustrated in Fig. 10, we investigate the average AoI performance of the system under different numbers of IIoT devices. It can be observed that as the number of IIoT devices increases, the system's average AoI also grows accordingly due

[1] Batchsize denotes the number of training samples per iteration. Smaller batch sizes enhance randomness and help escape local optima but increase gradient variance and instability, while larger batch sizes yield more stable gradients and faster convergence at the cost of higher resource demands and potential premature convergence.

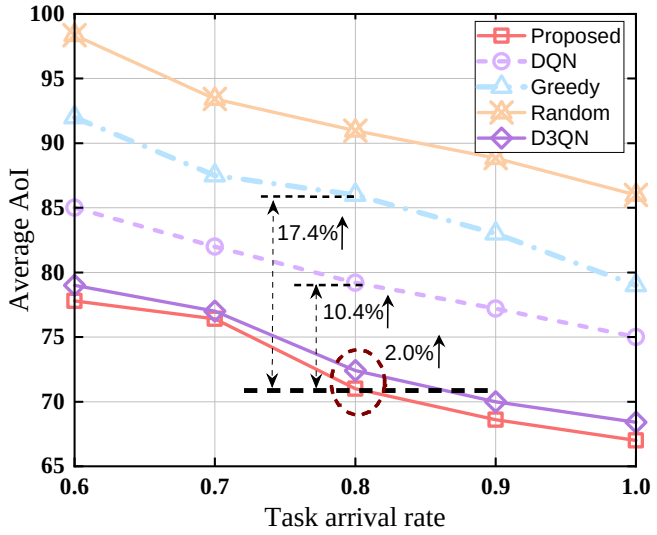


Fig. 9: Performance comparison of average AoI variation with task arrival rate.

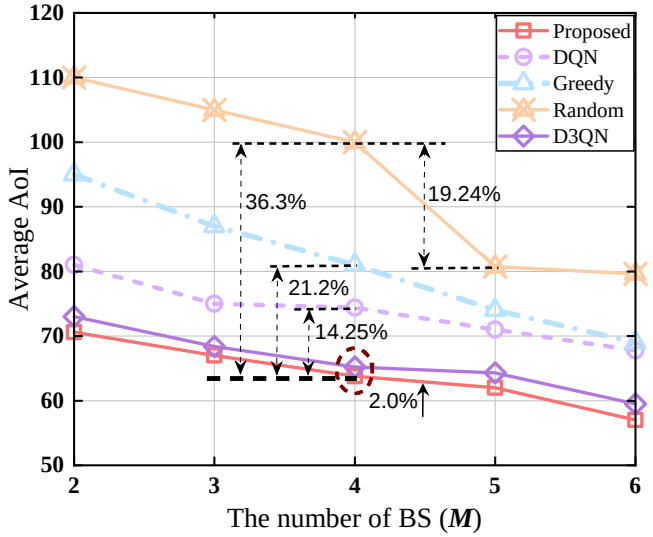


Fig. 11: The performance comparison of average AoI with varying numbers of BS under different schemes.

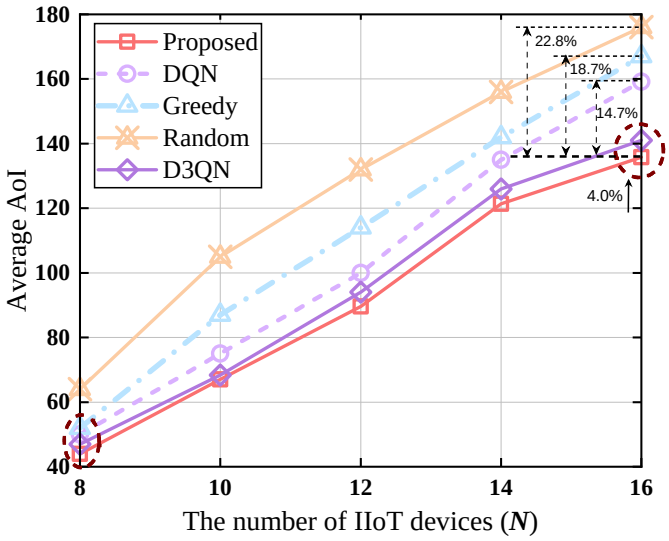


Fig. 10: The performance comparison of average AoI with varying numbers of IIoT devices under different schemes.

to heightened resource contention. However, compared to other baseline schemes, the proposed BD3QN algorithm demonstrates superior performance. Specifically, when the scale of IIoT devices is small (e.g., $N = 8$), compared to D3QN, DQN, and Greedy, the performance gains achieved by the proposed schemes in average AoI are not as pronounced; however, the proposed schemes achieve significant AoI reductions compared to the Random Offloading strategy. Nevertheless, as the scale of IIoT devices gradually increases (e.g., $N = 16$), the proposed algorithm achieves AoI reductions of 4%, 14.7%, 18.7%, and 22.8% over D3QN, DQN, Greedy, and Random Offloading, respectively. These results demonstrate that the proposed algorithm's performance advantage increases with the scale of IIoT devices, underscoring its scalability and efficacy in dense network environments.

As depicted in Fig. 11, we analyze the system's average AoI performance under different numbers of BSs. It can be seen that as the number of BSs increases, the average AoI performance is significantly improved. For instance, with the Random Offloading strategy, when the number of BSs increases from 4 to 5, the system's average AoI performance improves by 19.24%. However, the continuous increase in the number of BSs does not significantly improve the performance of average AoI. This is primarily because when the number of BSs exceeds a certain threshold, the impact of performance degradation caused by the uneven offloading resulting from the random deployment of the Random Offloading strategy diminishes. Additionally, when $M = 4$, the proposed algorithm shows a significant improvement in average AoI performance compared to D3QN, DQN, and Greedy, with performance gains of 2%, 14.25%, 21.2%, and 36.3%, respectively. These results highlight the proposed algorithm's superior scalability and efficacy in leveraging increased BS availability to optimize data freshness in multi-BS IIoT systems.

Drawing from Fig. 10 and Fig. 11, the superior performance of the proposed schemes stems from their efficient learning and selection of optimal task offloading strategies, significantly reducing the system's average AoI. For example, when the number of IIoT devices is $N = 10$ and the number of base stations is $M = 5$, the proposed schemes reduce the average AoI by 23% and 12% compared to the Greedy algorithm and DQN algorithm, respectively, and outperform the D3QN algorithm. These results suggest that the introduced branching structure can effectively enhance the system's real-time performance and data update efficiency by quickly determining the best offloading decisions. Furthermore, we can observe that increasing the number of IIoT devices can heighten the competition for offloading among IIoT devices, and the contention for BS services becomes more intense, leading to an increase in the system's average AoI. Conversely, more BSs reduce resource contention since more IIoT devices have the opportunity to offload data to the BSs, thereby reducing the average AoI. These dynamics highlight the need to balance device and BS counts in offloading strategies. While additional BSs enhance

data freshness, scaling IIoT devices increases costs. Therefore, in practical applications, appropriate algorithms and strategies should be adopted based on specific requirements and resource conditions to achieve optimal system performance.

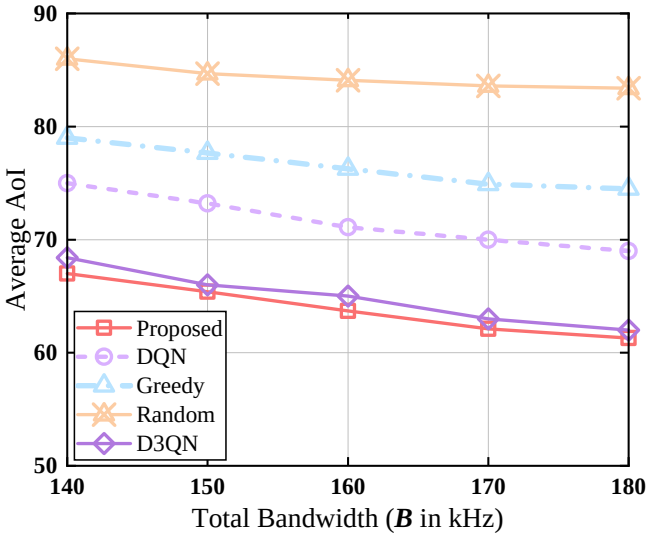


Fig. 12: The performance comparison of average AoI with varying bandwidth resources under different schemes.

Additionally, resource allocation also significantly influences the system's AoI performance. As shown in Fig. 12 and Fig. 13, as bandwidth and computational resources increase, the system's average AoI decreases, indicating an improvement in the system's timeliness. According to Eq. (3), when the amount of bandwidth allocated to a device increases, the device's transmission time will decrease. Similarly, as indicated by Eq. (5), when the amount of computational resources obtained by a device increases, its computation time will also shorten accordingly. These two changes are directly related to the calculation of the system's AoI. Based on the definition of AoI in Eq. (8), the system's average is closely related to the transmission and computation delays. However, as can be further observed from Fig. 12 and Fig. 13, when bandwidth and computational resources increase to a certain threshold, the AoI reduction of IIoT devices gradually slows down and even tends to stabilize. This is mainly because excessive resource allocation yields marginal improvements in transmission and computation delays, limiting further AoI decreases. Therefore, optimal resource allocation is critical to balance timeliness and resource utilization in real-time systems.

VII. Conclusion

In this paper, we propose an AoI-aware multi-BS real-time monitoring system for large-scale IIoT access, where a joint task offloading and resource allocation optimization problem is formulated to minimize the long-term average AoI. To effectively tackle this non-convex dynamic stochastic optimization problem, we decomposed it into the task offloading and resource allocation subproblems. For task offloading, we equivalently transform it into a CMDP and propose an innovative branching D3QN algorithm that reduces the action space complexity from exponential to $\mathcal{Q}(n)$. For resource

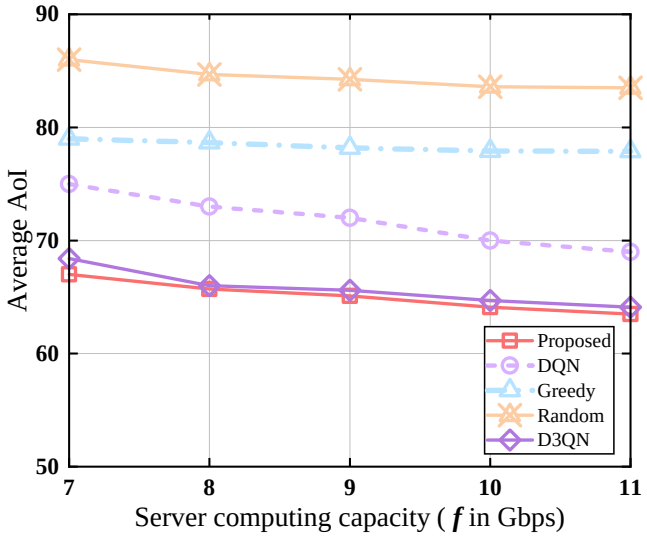


Fig. 13: The performance comparison of average AoI with varying server computation resources under different schemes.

allocation, we prove the convexity of this subproblem by deriving the semi-definite property of the Hessian matrix of bandwidth and computation resources. Furthermore, by employing an alternating optimization strategy, we propose a joint task offloading and resource allocation scheme that achieves optimal average AoI performance. Extensive simulations show that our proposed Branching-D3DN algorithm outperforms both the state-of-the-art DRL methods and classical heuristics, achieving up to 75% enhanced convergence speed and at least 22% reduced long-term average AoI.

Although this work addresses large-scale IIoT access in multi-BS scenarios, it assumes that each IIoT device is associated with a single BS. In real-world scenarios, devices may dynamically switch among multiple BS, leading to service migration. In future work, we intend to extend our framework to incorporate service migration, aiming to design more robust task offloading strategies with stronger adaptability to highly dynamic IIoT environments.

References

[1] Y. Chen, H. Lu, C. Wu, L. Qin, and X. Guo, "Performance optimization in RSMA-assisted uplink xURLLC IIoT networks with statistical QoS provisioning," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2025.

[2] M. A. Ferrag, O. Friha, B. Kantarci, N. Tihanyi, L. Cordeiro, M. Debbah, D. Hamouda, M. Al-Hawawreh, and K.-K. R. Choo, "Edge learning for 6G-enabled internet of things: A comprehensive survey of vulnerabilities, datasets, and defenses," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2654–2713, 2023.

[3] C. W. Chen, "Internet of video things: Next-generation IoT with visual sensors," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6676–6685, 2020.

[4] R. A. Lee, "Internet of things statistics 2025: Devices, security, and adoption," 2025, accessed: 2025-09-23. [Online]. Available: https://sqmagazine.co.uk/internet-of-things-statistics/

[5] T. Insights, "Global IoT forecast report, 2024-2034," 2025, accessed: 2025-09-23. [Online]. Available: https://transformainsights.com/research/reports/global-iot-forecast-report-2024-2034

[6] B. Jovanovic, "Internet of things statistics for 2024 – taking things apart," 2023, accessed: 2025-09-23. [Online]. Available: https://dataprot.net/blog/iot-statistics/

[7] M. Bansal, I. Chana, and S. Clarke, "A survey on IoT big data: current status, 13 v's challenges, and future directions," *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, pp. 1–59, 2020.

[8] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "A survey on network methodologies for real-time analytics of massive IoT data and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1457–1477, 2017.

[9] X. Dai, Z. Xiao, H. Jiang, M. Alazab, J. C. Lui, S. Dustdar, and J. Liu, "Task co-offloading for D2D-assisted mobile edge computing in industrial internet of things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 480–490, 2022.

[10] J. Huang, H. Gao, S. Wan, and Y. Chen, "AoI-aware energy control and computation offloading for industrial IoT," *Future Generation Computer Systems*, vol. 139, pp. 29–37, 2023.

[11] Y. Chen, C. Lu, Y. Huang, C. Wu, F. Guo, H. Lu, and C. W. Chen, "DMSA: A decentralized microservice architecture for edge networks," *arXiv preprint arXiv:2501.00883*, 2025.

[12] L. Qin, H. Lu, Y. Chen, Z. Gu, D. Zhao, and F. Wu, "Energy-efficient blockchain-enabled user-centric mobile edge computing," *IEEE Trans. Cogn. Commun. Netw.*, vol. 10, no. 4, pp. 1452–1466, 2024.

[13] L. Qin, H. Lu, Y. Chen, B. Chong, and F. Wu, "Toward decentralized task offloading and resource allocation in user-centric MEC," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 11 807–11 823, 2024.

[14] Y. Chen, C. Wu, F. Zhang, C. Lu, Y. Huang, and H. Lu, "Topology-aware microservice architecture in edge networks: Deployment optimization and implementation," *IEEE Trans. Mobile Comput.*, vol. 24, no. 7, pp. 6090–6105, 2025.

[15] Y. Chen, H. Lu, L. Qin, Y. Deng, and A. Nallanathan, "When xURLLC meets NOMA: A stochastic network calculus perspective," *IEEE Commun. Mag.*, vol. 62, no. 6, pp. 90–96, 2024.

[16] L. Hu, D. Deng, F. Chen, P. Wu, and Z. Chen, "Timeliness analysis for real-time updates across different service time," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 6, pp. 8054–8068, 2024.

[17] N. Lu, B. Ji, and B. Li, "Age-based scheduling: Improving data freshness for wireless real-time traffic," in *Proceedings of the eighteenth ACM international symposium on mobile ad hoc networking and computing*, 2018, pp. 191–200.

[18] D. Guo, K. Nakhleh, I.-H. Hou, S. Kompella, and C. Kam, "AoI, timely-throughput, and beyond: A theory of second-order wireless network optimization," *IEEE/ACM Trans. Netw.*, 2024.

[19] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, 2021.

[20] M. A. Abd-Elmagid, N. Pappas, and H. S. Dhillon, "On the role of age of information in the internet of things," *IEEE Commun. Mag.*, vol. 57, no. 12, pp. 72–77, 2019.

[21] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. Inform. Theory*, vol. 65, no. 3, pp. 1807–1827, 2018.

[22] Y. Zeng, Y. Zeng, J. Chen, Y. Shen, L. Li, P. Cong, J. Zhou, and K. Li, "AoI-oriented computation offloading and resource allocation for end-edge-cloud computing systems," *IEEE Internet of Things J.*, vol. 12, no. 19, pp. 41 118–41 134, 2025.

[23] J. Feng and J. Gong, "Joint detection and computation offloading with age of information in mobile edge networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 3, pp. 1417–1430, 2023.

[24] O. A. Amodu, C. Jarray, R. Azlina Raja Mahmood, H. Althumali, U. Ali Bukar, R. Nordin, N. F. Abdullah, and N. Cong Luong, "Deep reinforcement learning for AoI minimization in UAV-aided data collection for WSN and IoT applications: A survey," *IEEE Access*, vol. 12, pp. 108 000–108 040, 2024.

[25] K. Peng, P. Xiao, S. Wang, and V. C. M. Leung, "AoI-aware partial computation offloading in IIoT with edge computing: A deep reinforcement learning based approach," *IEEE Trans. Cloud Comput.*, vol. 11, no. 4, pp. 3766–3777, 2023.

[26] L. Tan, Z. Kuang, L. Zhao, and A. Liu, "Energy-efficient joint task offloading and resource allocation in OFDMA-based collaborative edge computing," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1960–1972, 2022.

[27] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 2731–2735.

[28] E. Najm and E. Telatar, "Status updates in a multi-stream M/G/1/1 preemptive queue," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2018, pp. 124–129.

[29] M. Hatami, M. Leinonen, and M. Codreanu, "AoI minimization in status update control with energy harvesting sensors," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8335–8351, 2021.

[30] Z. Tang, Z. Sun, N. Yang, and X. Zhou, "Age of information of multi-user mobile-edge computing systems," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 1600–1614, 2023.

[31] W. Fan, "Blockchain-secured task offloading and resource allocation for cloud-edge-end cooperative networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8092–8110, 2024.

[32] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[33] W. Zhan, C. Luo, J. Wang, G. Min, and H. Duan, "Deep reinforcement learning-based computation offloading in vehicular edge computing," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[34] C. Shang, Y. Sun, H. Luo, and M. Guizani, "Computation offloading and resource allocation in NOMA–MEC: A deep reinforcement learning approach," *IEEE Internet of Things J.*, vol. 10, no. 17, pp. 15 464–15 476, 2023.

[35] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[36] M. Sewak, "Deep Q network (DQN), double DQN, and dueling DQN: A step towards general artificial intelligence," in *Deep reinforcement learning: frontiers of artificial intelligence*. Springer, 2019, pp. 95–108.

[37] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[38] Y. Jiang, J. Liu, I. Humar, M. Chen, S. A. AlQahtani, and M. S. Hossain, "Age-of-information-based computation offloading and transmission scheduling in mobile-edge-computing-enabled IoT networks," *IEEE Internet of Things J.*, vol. 10, no. 22, pp. 19 782–19 794, 2023.

[39] Z. Zabihi, A. M. Eftekhari Moghadam, and M. H. Rezvani, "Reinforcement learning methods for computation offloading: a systematic review," *ACM Comput. Surv.*, vol. 56, no. 1, pp. 1–41, 2023.

[40] J. Chi, X. Zhou, F. Xiao, Y. Lim, and T. Qiu, "Task offloading via prioritized experience-based double dueling DQN in edge-assisted IIoT," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 14 575–14 591, 2024.

[41] A. Muhammad, I. Sorkhoh, M. Samir, D. Ebrahimi, and C. Assi, "Minimizing age of information in multiaccess-edge-computing-assisted IoT networks," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13 052–13 066, 2022.

[42] H. Li, J. Zhang, H. Zhao, Y. Ni, J. Xiong, and J. Wei, "Joint optimization on trajectory, computation and communication resources in information freshness sensitive MEC system," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 4162–4177, 2024.

[43] Y. Le, R. Jiang, Y. Jiang, X. Ling, J. Wang, D. W. K. Ng, and Y. Huang, "Age-of-information analysis for blockchain-based mobile edge computing," *IEEE Trans. Commun.*, pp. 1–1, 2025.

[44] C. Wu, Y. Chen, Y. Chen, F. Guo, X. Qin, and H. Lu, "Physiological signal-driven QoE optimization for wireless virtual reality transmission," *arXiv preprint arXiv:2508.09151*, 2025.

[45] F. Song, Q. Yang, M. Deng, H. Xing, Y. Liu, X. Yu, K. Li, and L. Xu, "AoI and energy tradeoff for aerial-ground collaborative MEC: A multi-objective learning approach," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 11 278–11 294, 2024.

[46] H. Wang, C. H. Liu, H. Yang, G. Wang, and K. K. Leung, "Ensuring threshold AoI for UAV-assisted mobile crowdsensing by multi-agent deep reinforcement learning with transformer," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 566–581, 2024.

[47] G. Zhang, X. Wei, X. Tan, Z. Han, and G. Zhang, "AoI minimization based on deep reinforcement learning and matching game for IoT information collection in SAGIN," *IEEE Trans. Commun.*, vol. 73, no. 8, pp. 5950–5964, 2025.

[48] X. Chen, S. Hu, C. Yu, Z. Chen, and G. Min, "Real-time offloading for dependent and parallel tasks in cloud-edge environments using deep reinforcement learning," *Trans. Parallel Distrib. Syst.*, vol. 35, no. 3, pp. 391–404, 2024.

[49] H. Zhou, X. Chen, S. He, C. Zhu, and V. C. M. Leung, "Freshness-aware seed selection for offloading cellular traffic through opportunistic mobile networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2658–2669, 2020.