

AGE OF INFORMATION-AWARE PROXIMAL POLICY OPTIMIZATION IN MEC-ENABLED HEALTHCARE INTERNET OF THINGS

Mohammed Maray^{*1}, Ali Mere², Razan Alwadei³

^{*1,2,3}Department of Informatics and Computer Systems, College of Computer Science, King Khalid University, AlQura'a, Abha, P.O. Box 960, Saudi Arabia

^{*1}mmarey@kku.edu.sa

DOI: <https://doi.org/10.5281/zenodo.21485265>

Keywords

Healthcare, Internet of Things, Age of Information, Proximal Policy Optimization, Mobile Edge Computing.

Article History

Received: 02 February 2026

Accepted: 20 March 2026

Published: 31 March 2026

Copyright @Author

Corresponding Author: *

Mohammed Maray

Abstract

The Healthcare Internet of Things (H-IoT) enables continuous remote patient monitoring through wearable sensors and smart medical devices. However, the limited computational capability of H-IoT devices and the dynamic nature of wireless edge networks make efficient computation offloading a challenging task. Moreover, conventional offloading schemes primarily optimize latency and energy consumption while overlooking the freshness of medical information, which is critical for timely healthcare services. To address this issue, this paper proposes an Age of Information-aware Proximal Policy Optimization (AA-PPO) framework for computation offloading in Mobile Edge Computing (MEC)-enabled H-IoT networks. The proposed framework jointly minimizes the Average Age of Information (AoI), end-to-end latency, and energy consumption by learning adaptive offloading policies according to dynamic network conditions. A comprehensive system model is developed, incorporating healthcare task generation, wireless communication, computation, queueing, AoI, latency, and energy consumption. The computation offloading problem is formulated as a Markov Decision Process (MDP), and an actor-critic-based PPO algorithm is employed to obtain an efficient offloading policy. Simulation results demonstrate that the proposed AA-PPO consistently outperforms Local Execution, Random Offloading, Greedy Edge Offloading, DQN, and Double DQN. Compared with Double DQN, the proposed approach reduces the Average AoI by up to 20.5%, Peak AoI by 23.4%, end-to-end latency by 12.8%, and energy consumption by 13.3%. These results demonstrate the effectiveness of the proposed framework in maintaining fresh healthcare information while improving the overall efficiency of MEC-enabled H-IoT systems.

Introduction

The Healthcare Internet of Things (H-IoT) has become an important technology for enabling continuous patient monitoring through wearable sensors, smart medical devices, and wireless communication networks [1][2]. These devices periodically collect physiological signals,

including electrocardiograms (ECG), blood pressure, blood glucose, body temperature, and oxygen saturation, allowing healthcare professionals to monitor patients remotely and respond promptly to critical medical conditions. However, most H-IoT devices possess limited computational capability and battery capacity,

making it difficult to process computation-intensive healthcare applications locally [3][4][5]. Mobile Edge Computing (MEC) has emerged as an effective solution for addressing these limitations by providing computational resources close to healthcare devices [6][7][8]. Instead of executing tasks locally, H-IoT devices can offload computational workloads to nearby MEC servers, thereby reducing processing delay, conserving device energy, and improving the overall quality of healthcare services. Nevertheless, the dynamic nature of wireless networks, varying computational workloads, and fluctuating edge resource availability make efficient computation offloading a challenging problem [9]-[10].

For many healthcare applications, minimizing computation latency alone is insufficient. Medical information continuously loses its usefulness as it becomes older. For example, an electrocardiogram or oxygen saturation measurement collected several seconds earlier may no longer accurately represent a patient's current physiological condition. Consequently, maintaining fresh healthcare information is essential for supporting timely diagnosis and emergency medical responses. Age of Information (AoI) has recently emerged as an effective metric for quantifying information freshness by measuring the time elapsed since the latest successfully received update was generated [11].

Although several computation offloading approaches have been proposed for MEC-enabled H-IoT systems, most existing studies primarily optimize latency, energy consumption, or resource utilization while overlooking information freshness. As a result, computation offloading decisions that minimize communication delay do not necessarily minimize the Age of Information. Furthermore, conventional optimization techniques require accurate mathematical models and often become computationally expensive in dynamic healthcare environments characterized by stochastic task arrivals, time-varying wireless channels, and continuously changing network conditions [12].

Deep Reinforcement Learning (DRL) provides an effective framework for sequential decision-making under uncertain environments by

enabling agents to learn optimal offloading policies directly through interactions with the network [13][14]. Among DRL algorithms, Proximal Policy Optimization (PPO) offers stable policy updates, efficient learning, and strong convergence characteristics, making it suitable for dynamic Healthcare IoT environments. By continuously adapting computation offloading decisions according to current network conditions and information freshness, PPO can effectively balance multiple system objectives without requiring prior knowledge of the environment [15][16][17].

Motivated by these challenges, this paper proposes an Age of Information-aware Proximal Policy Optimization (AA-PPO) framework for computation offloading in MEC-enabled Healthcare IoT networks. The proposed framework jointly minimizes the Average Age of Information, end-to-end latency, and energy consumption by learning adaptive computation offloading policies under dynamic network conditions.

The main contributions of this paper are summarized as follows:

- A comprehensive system model is developed for MEC-enabled Healthcare Internet of Things networks, incorporating healthcare task generation, wireless communication, computation offloading, queueing dynamics, Age of Information, latency, and energy consumption into a unified framework.
- An Age-of-Information-aware Proximal Policy Optimization (AA-PPO) algorithm is proposed to learn adaptive computation offloading policies that jointly optimize information freshness, end-to-end latency, and energy consumption under dynamic H-IoT environments.
- Extensive simulations are conducted to evaluate the proposed framework against conventional computation offloading methods and existing DRL approaches. The results demonstrate that the proposed AA-PPO consistently achieves lower Average AoI, Peak AoI, latency,

and energy consumption, thereby improving the overall performance of System Model

Healthcare IoT systems.

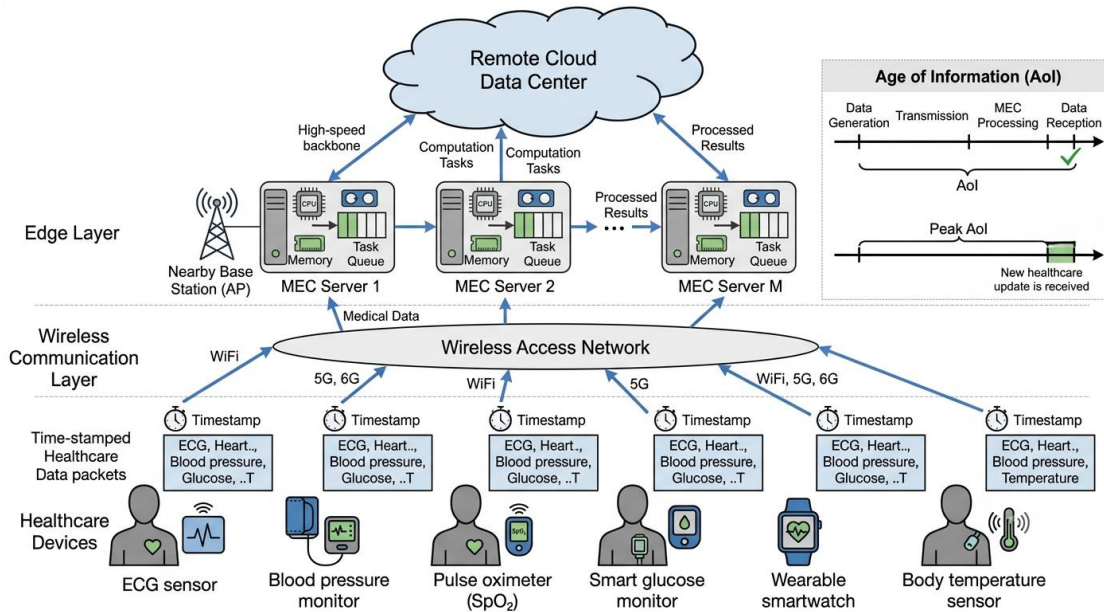


Figure 1: System Model of Proposed Work

Network Architecture

We consider a Healthcare Internet of Things (H-IoT) system consisting of a set of resource-constrained healthcare devices, multiple Mobile Edge Computing (MEC) servers, and a remote cloud server, as illustrated in Figure 1. The healthcare devices continuously monitor patients using wearable or implantable medical sensors, while the MEC servers provide nearby computational resources for delay-sensitive healthcare applications. The cloud server possesses abundant computational capability and storage resources but is mainly utilized for long-term data storage and computationally intensive analytics due to its relatively higher communication latency [18][19][20].

Let the set of H-IoT devices be denoted by

$$\mathcal{N} = \{1, 2, \dots, N\},$$

where N represents the total number of healthcare devices deployed within the network.

Similarly, let

$$\mathcal{M} = \{1, 2, \dots, M\},$$

denote the set of MEC servers, where M is the total number of available edge servers.

Each H-IoT device periodically senses physiological information, including

electrocardiogram (ECG), blood pressure, body temperature, oxygen saturation, glucose level, and other vital signs. The collected measurements are encapsulated into computational tasks requiring real-time processing for disease diagnosis, anomaly detection, and patient monitoring.

The generated healthcare tasks can either be executed locally on the originating H-IoT device or offloaded to one of the available MEC servers through wireless communication. Since the computational capability and battery capacity of H-IoT devices are limited, local execution may result in excessive execution delay and energy consumption for computation-intensive healthcare applications. Consequently, MEC servers become the preferred execution platforms due to their proximity to healthcare devices and lower communication latency.

Time is divided into equal decision intervals indexed by

$$t = 0, 1, 2, \dots, T,$$

where T denotes the total number of decision epochs. During each decision interval, every H-IoT device generates at most one healthcare computation task and determines whether the

task should be processed locally or offloaded to one of the MEC servers.

The offloading decision of healthcare device i during time slot t is represented as

$$a_i(t) \in \{0, 1, \dots, M\},$$

where

$$a_i(t) = \begin{cases} 0, & \text{Local execution,} \\ j, & \text{Offloading to MEC server } j, j \in \mathcal{M}. \end{cases}$$

Each MEC server maintains an independent computation queue for processing the received healthcare tasks according to its available computational resources. Owing to the dynamic wireless environment, varying computational workloads, and fluctuating queue occupancies, the network state changes continuously over time. Therefore, intelligent computation offloading decisions are required to efficiently utilize the available edge resources.

Unlike conventional MEC systems that primarily optimize latency or energy consumption, the considered H-IoT framework explicitly incorporates the freshness of medical information. Since physiological data gradually lose their clinical significance as they become outdated, maintaining a low Age of Information (AoI) is essential for reliable healthcare monitoring. Therefore, each offloading decision simultaneously affects communication delay, computation delay, queueing delay, energy consumption, and the freshness of healthcare information available for medical decision making.

Healthcare Task Model

Each H-IoT device periodically generates computation tasks by collecting physiological measurements from wearable or implantable medical sensors. These tasks correspond to healthcare applications such as electrocardiogram (ECG) monitoring, blood pressure analysis, blood glucose monitoring, pulse oximetry, body temperature assessment, and medical image processing. Since different healthcare applications generate heterogeneous workloads, the corresponding computation tasks exhibit varying input data sizes and computational requirements [21].

Let the healthcare task generated by H-IoT device i during time slot t be represented as

$$\mathcal{T}_i(t) = \{D_i(t), C_i(t)\},$$

where

- $D_i(t)$ (bits) denotes the input data size of the healthcare task.
- $C_i(t)$ (CPU cycles) represents the computational workload required to complete the task.

The generated healthcare tasks are heterogeneous due to different sensing modalities and application requirements. Accordingly,

$$D_i^{\min} \leq D_i(t) \leq D_i^{\max},$$

and

$$C_i^{\min} \leq C_i(t) \leq C_i^{\max},$$

where $D_i^{\min}$ and $D_i^{\max}$ denote the minimum and maximum task input sizes, while $C_i^{\min}$ and $C_i^{\max}$ represent the corresponding computational workload limits.

Healthcare tasks are generated continuously according to the sensing frequency of each medical device. Let

$$A_i(t) \in \{0, 1\}$$

denote the task arrival indicator, where

$$A_i(t) = \begin{cases} 1, & \text{if a new healthcare task arrives during time slot } t, \\ 0, & \text{otherwise.} \end{cases}$$

The average task arrival rate of H-IoT device i is denoted by

$$\lambda_i = \mathbb{E}[A_i(t)],$$

where $\mathbb{E}[\cdot]$ represents the expectation operator.

Each generated healthcare task is associated with a sensing timestamp

$$t_i^g,$$

which records the instant at which the physiological information is acquired from the patient. Unlike conventional computation tasks, the clinical significance of healthcare information gradually decreases as the elapsed time from its generation increases. Therefore, the sensing timestamp plays a fundamental role in evaluating the freshness of healthcare information through the Age of Information (AoI), which will be formally defined in Section III-F.

After task generation, H-IoT device i determines the execution destination according to the offloading decision $a_i(t)$. If

$$a_i(t) = 0,$$

the task is processed locally on the healthcare device. Otherwise, if

$$a_i(t) = j, \quad j \in \mathcal{M},$$

the task is transmitted through the wireless channel to MEC server j for edge execution.

To simplify the analysis, each healthcare task is assumed to be indivisible, implying that a task cannot be partitioned between local execution and multiple MEC servers. Consequently, each generated task is executed entirely at a single computing node, either locally or at one selected MEC server.

Healthcare Task Model

Each H-IoT device periodically generates computation tasks by collecting physiological measurements from wearable or implantable medical sensors. These tasks correspond to healthcare applications such as electrocardiogram (ECG) monitoring, blood pressure analysis, blood glucose monitoring, pulse oximetry, body temperature assessment, and medical image processing. Since different healthcare applications generate heterogeneous workloads, the corresponding computation tasks exhibit varying input data sizes and computational requirements.

Let the healthcare task generated by H-IoT device i during time slot t be represented as

$$\mathcal{T}_i(t) = \{D_i(t), C_i(t)\},$$

where

- $D_i(t)$ (bits) denotes the input data size of the healthcare task.
- $C_i(t)$ (CPU cycles) represents the computational workload required to complete the task.

The generated healthcare tasks are heterogeneous due to different sensing modalities and application requirements. Accordingly,

$$D_i^{\min} \leq D_i(t) \leq D_i^{\max},$$

and

$$C_i^{\min} \leq C_i(t) \leq C_i^{\max},$$

where $D_i^{\min}$ and $D_i^{\max}$ denote the minimum and maximum task input sizes, while $C_i^{\min}$ and $C_i^{\max}$ represent the corresponding computational workload limits.

Healthcare tasks are generated continuously according to the sensing frequency of each medical device. Let

$$A_i(t) \in \{0,1\}$$

denote the task arrival indicator, where

$$A_i(t)$$

$$= \begin{cases} 1, & \text{if a new healthcare task arrives during time slot } t, \\ 0, & \text{otherwise.} \end{cases}$$

The average task arrival rate of H-IoT device i is denoted by

$$\lambda_i = \mathbb{E}[A_i(t)],$$

where $\mathbb{E}[\cdot]$ represents the expectation operator.

Each generated healthcare task is associated with a sensing timestamp

$$t_i^g,$$

which records the instant at which the physiological information is acquired from the patient. Unlike conventional computation tasks, the clinical significance of healthcare information gradually decreases as the elapsed time from its generation increases. Therefore, the sensing timestamp plays a fundamental role in evaluating the freshness of healthcare information through the Age of Information (AoI), which will be formally defined in Section III-F.

After task generation, H-IoT device i determines the execution destination according to the offloading decision $a_i(t)$. If

$$a_i(t) = 0,$$

the task is processed locally on the healthcare device. Otherwise, if

$$a_i(t) = j, \quad j \in \mathcal{M},$$

the task is transmitted through the wireless channel to MEC server j for edge execution.

To simplify the analysis, each healthcare task is assumed to be indivisible, implying that a task cannot be partitioned between local execution and multiple MEC servers. Consequently, each generated task is executed entirely at a single computing node, either locally or at one selected MEC server.

Computation Model

After the healthcare task is generated, it is executed either locally at the originating H-IoT device or remotely at the selected MEC server according to the offloading decision. The execution time depends on the computational

workload of the task and the available processing capability of the selected computing node.

Local Computation

If healthcare task $\mathcal{T}_i(t)$ is executed locally, the execution latency depends on the computational capability of H-IoT device i . Let f_i^L (CPU cycles/s) denote the local CPU frequency of healthcare device i . The local computation time is therefore expressed as

$$T_i^L(t) = \frac{C_i(t)}{f_i^L},$$

where $C_i(t)$ denotes the required CPU cycles for completing the healthcare task.

The corresponding local computation energy consumption follows the widely adopted dynamic voltage-frequency scaling (DVFS) model and is given by

$$E_i^L(t) = \kappa_i C_i(t) (f_i^L)^2,$$

where κ_i is the effective switched capacitance coefficient of H-IoT device i .

Although local execution avoids wireless transmission delay, the limited processing capability of healthcare devices generally leads to relatively high computation latency and energy consumption, particularly for computation-intensive medical applications.

MEC Computation

When healthcare task $\mathcal{T}_i(t)$ is offloaded to MEC server j , the execution latency depends on the computational capability of the selected edge server. Let f_j^M (CPU cycles/s) denote the CPU frequency allocated by MEC server j for processing task i . The MEC computation time is therefore expressed as

$$T_{ij}^M(t) = \frac{C_i(t)}{f_j^M}.$$

Since MEC servers possess substantially higher computational resources than H-IoT devices, the allocated processing frequency generally satisfies

$$f_j^M \gg f_i^L,$$

which enables significantly faster execution of healthcare tasks.

The overall execution time of an offloaded healthcare task consists of three sequential components:

1. Wireless transmission,

2. Queue waiting,
3. MEC computation.

Accordingly, the total edge execution delay is expressed as

$$T_{ij}^E(t) = T_{ij}^{tx}(t) + T_{ij}^Q(t) + T_{ij}^M(t),$$

where $T_{ij}^Q(t)$ denotes the queue waiting time at MEC server j .

Finally, the computation latency experienced by healthcare task i is determined according to its execution location as

$$T_i^C(t) = \begin{cases} T_i^L(t), & a_i(t) = 0, \\ T_{ij}^E(t), & a_i(t) = j, j \in \mathcal{M}. \end{cases}$$

The above computation model establishes the relationship between task execution latency and the selected execution platform. Since the completion time directly determines when newly sensed healthcare information becomes available after processing, it also has a significant influence on the Age of Information, which will be formally modeled in the subsequent sections.

Queueing Model

Since multiple H-IoT devices may simultaneously offload healthcare tasks to the same MEC server, the arriving tasks are buffered in a waiting queue before execution. Consequently, the queueing delay becomes an important component of the overall execution latency and directly affects the freshness of healthcare information.

Let $Q_j(t)$ denote the queue length of MEC server j at the beginning of time slot t , which represents the total amount of computational workload waiting to be processed. The queue length evolves dynamically according to the incoming and completed computation tasks.

The queue dynamics at MEC server j are expressed as

$$Q_j(t+1) = \max\{Q_j(t) - S_j(t), 0\} + A_j(t),$$

where

- $A_j(t)$ denotes the total computational workload of newly arrived healthcare tasks during time slot t ,
- $S_j(t)$ represents the computational workload completed by MEC server j during the same time slot.

The total arriving workload is computed as

$$A_j(t) = \sum_{i=1}^N \mathbb{I}(a_i(t) = j) C_i(t),$$

where $\mathbb{I}(\cdot)$ is the indicator function defined by

$$\mathbb{I}(x) = \begin{cases} 1, & x \text{ is true,} \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, the service workload processed by MEC server j during one decision interval is

$$S_j(t) = f_j^M \tau,$$

where

- f_j^M denotes the computational capability of MEC server j (CPU cycles/s),
- τ is the duration of one decision interval.

The average queue waiting time experienced by healthcare task i at MEC server j is approximated by

$$T_{ij}^Q(t) = \frac{Q_j(t)}{f_j^M},$$

which indicates that the waiting delay is proportional to the accumulated computational workload and inversely proportional to the processing capability of the selected MEC server. Therefore, the total execution delay of an offloaded healthcare task becomes

$$T_{ij}^E(t) = T_{ij}^{\text{tx}}(t) + T_{ij}^Q(t) + T_{ij}^M(t).$$

As the number of H-IoT devices increases or multiple devices simultaneously select the same MEC server, the queue length grows, leading to longer waiting times before execution. Consequently, excessive queue congestion not only increases the overall computation latency but also causes healthcare information to become progressively older before processing. This degradation in information freshness directly influences the Age of Information (AoI), which forms the primary optimization objective of the proposed framework and is mathematically formulated in the following subsection.

Age of Information (AoI) Model

The Age of Information (AoI) is adopted as the primary metric to quantify the freshness of healthcare information available at the MEC server. Unlike conventional latency metrics that only measure task completion delay, AoI represents the elapsed time since the most recently received healthcare information was originally generated. Consequently, AoI provides

a direct measure of information freshness, which is particularly important for continuous patient monitoring and time-sensitive medical diagnosis.

Let $t_i^g(k)$ denote the generation time of the k -th healthcare update produced by H-IoT device i , and let $t_i^r(k)$ denote the corresponding reception (successful processing) time at the destination MEC server.

The instantaneous Age of Information of healthcare device i at time t is defined as

$$\Delta_i(t) = t - t_i^g(u_i(t)),$$

where $u_i(t)$ denotes the latest successfully processed healthcare update available at time t .

As time progresses without receiving a fresh healthcare update, the AoI increases linearly according to

$$\Delta_i(t+1) = \Delta_i(t) + \tau,$$

where τ denotes the duration of one decision interval.

Whenever a newly generated healthcare task is successfully processed and delivered to the MEC server, the AoI is refreshed to the total elapsed time between task generation and successful completion. Accordingly,

$$\Delta_i(t+1) = T_i^{E2E}(t),$$

where $T_i^{E2E}(t)$ denotes the end-to-end completion delay of healthcare task i .

Combining the above cases, the AoI evolution can be expressed as

$$\Delta_i(t+1) = \begin{cases} T_i^{E2E}(t), & \text{if a fresh healthcare update is successfully processed} \\ \Delta_i(t) + \tau, & \text{otherwise.} \end{cases}$$

The average Age of Information of healthcare device i over an observation period of T decision intervals is given by

$$\bar{\Delta}_i = \frac{1}{T} \sum_{t=1}^T \Delta_i(t).$$

The network-wide average AoI is therefore

$$\bar{\Delta} = \frac{1}{N} \sum_{i=1}^N \bar{\Delta}_i.$$

Besides the average AoI, another important performance indicator is the Peak Age of Information (PAoI), which represents the maximum information age immediately before receiving a fresh healthcare update. The Peak AoI of healthcare device i is defined as

$$\Delta_i^{\text{Peak}} = \max_t \Delta_i(t).$$

Similarly, the average Peak AoI of the entire H-LoT network is computed as

$$\bar{\Delta}^{\text{Peak}} = \frac{1}{N} \sum_{i=1}^N \Delta_i^{\text{Peak}}.$$

The end-to-end completion delay directly determines the freshness of healthcare information after each successful update. Let

$$T_i^{\text{E2E}}(t) = T_i^{\text{tx}}(t) + T_i^{\text{Q}}(t) + T_i^{\text{C}}(t),$$

where $T_i^{\text{tx}}(t)$, $T_i^{\text{Q}}(t)$, and $T_i^{\text{C}}(t)$ denote the transmission delay, queueing delay, and computation delay, respectively. It follows that reducing any of these delay components immediately lowers the resulting AoI, enabling fresher physiological information to be available for healthcare monitoring and medical decision making.

Therefore, unlike conventional MEC optimization that minimizes latency alone, the

proposed framework explicitly minimizes the Average AoI while simultaneously reducing latency and energy consumption, ensuring that healthcare decisions are made using the freshest possible patient information.

Problem Formulation

The objective of the proposed Healthcare Internet of Things (H-LoT) framework is to determine an optimal computation offloading policy that maintains the freshness of healthcare information while simultaneously reducing computation latency and energy consumption. Since these performance metrics are interrelated, the computation offloading problem is formulated as a multi-objective optimization problem.

The primary optimization objective is to minimize the network-wide average Age of Information (AoI), which is expressed as

$$\min \bar{\Delta} = \frac{1}{N} \sum_{i=1}^N \bar{\Delta}_i.$$

Meanwhile, the overall end-to-end latency of all healthcare tasks is minimized according to

$$\min \sum_{i=1}^N T_i^{\text{E2E}},$$

where

$$T_i^{\text{E2E}} = T_i^{\text{tx}} + T_i^{\text{Q}} + T_i^{\text{C}}.$$

Similarly, the total energy consumption of the H-LoT system is minimized as

$$\min \sum_{i=1}^N E_i,$$

where

$$E_i = E_i^{\text{L}} + E_i^{\text{tx}}.$$

Accordingly, the joint optimization problem can be formulated as

$$\min_{\mathbf{A}} \left[\alpha \bar{\Delta} + \beta \sum_{i=1}^N T_i^{\text{E2E}} + \eta \sum_{i=1}^N E_i \right],$$

subject to the following constraints:

$$\begin{aligned} a_i &\in \{0, 1, \dots, M\}, & \forall i \in \mathcal{N}, \\ \sum_{i=1}^N \mathbb{I}(a_i = j) C_i &\leq f_j^M, & \forall j \in \mathcal{M}, \\ Q_j &\leq Q_j^{\max}, & \forall j \in \mathcal{M}, \\ R_{ij} &> 0, & \forall i, j, \\ \alpha + \beta + \eta &= 1, \end{aligned}$$

where

- Constraint 1 ensures that each healthcare task is executed either locally or at one MEC server.
- Constraint 2 guarantees that the computational workload assigned to each MEC server does not exceed its available computational capability.
- Constraint 3 limits the queue occupancy of each MEC server to avoid excessive congestion.
- Constraint 4 guarantees successful wireless communication for task offloading.
- Constraint 5 normalizes the weighting coefficients of the optimization objectives.

The formulated optimization problem is a mixed-integer nonlinear optimization problem due to the discrete offloading decisions, dynamic wireless channel conditions, stochastic healthcare task arrivals, and continuously varying queue states. Furthermore, the Age of Information evolves dynamically according to previous offloading decisions, making the optimization problem computationally intractable for conventional optimization techniques.

Therefore, a DRL-based solution is adopted to learn the optimal computation offloading policy directly through interactions with the H-IoT environment. Specifically, this paper proposes an Age of Information-aware Dueling Double Deep Q Network (AA-D3QN), which enables the learning agent to jointly minimize information freshness, computation latency, and energy consumption under dynamic network conditions.

Proposed Age of Information-aware Proximal Policy Optimization

To solve the formulated computation offloading problem, this paper proposes an Age of Information-aware Proximal Policy Optimization (AA-PPO) framework. Unlike conventional optimization methods that require complete knowledge of the dynamic H-IoT environment, the proposed AA-PPO continuously interacts with the network and learns an optimal computation offloading policy through trial-and-error learning. The learned policy jointly

minimizes the Average Age of Information (AoI), end-to-end latency, and energy consumption while adapting to dynamic wireless channels, varying computational workloads, and changing queue conditions [22]-[23].

The proposed framework follows an actor-critic architecture consisting of two deep neural networks. The actor network learns the optimal computation offloading policy by selecting the execution destination for each healthcare task, whereas the critic network estimates the expected long-term cumulative reward corresponding to the current system state. During each decision interval, the actor observes the current network state, generates an offloading decision, receives an immediate reward from the environment, and updates both networks using the PPO optimization strategy. By iteratively interacting with the H-IoT environment, the proposed framework gradually converges toward an optimal computation offloading policy that preserves healthcare information freshness while efficiently utilizing edge computational resources.

Actor-Critic Network Architecture

The proposed AA-PPO employs two independent neural networks, namely the actor network and the critic network, to approximate the optimal policy and value function, respectively.

The actor network is parameterized by

$$\pi_{\theta}(a|s),$$

where

- s denotes the current network state,
- a represents the selected computation offloading action,
- θ denotes the trainable parameters of the actor network.

Given the current state s_t , the actor outputs a probability distribution over all feasible computation offloading actions,

$$\pi_{\theta}(a_t|s_t) = P(a_t = s_i),$$

where

$$a_t \in \{0, 1, \dots, M\}.$$

The selected action determines whether the healthcare task is executed locally or offloaded to one of the available MEC servers.

The critic network estimates the expected cumulative discounted reward associated with the current state and is represented by

$$V_{\phi}(s_t),$$

where ϕ denotes the trainable parameters of the critic network.

The critic provides an estimate of the state value, enabling the actor to evaluate the quality of the selected computation offloading decision. Consequently, both networks are optimized simultaneously throughout the training process to improve the computation offloading policy.

State Representation

At every decision interval, the actor observes the current H-IoT environment represented by

$$s_t = [D_i, C_i, \Delta_i, Q_1, \dots, Q_M, R_{i1}, \dots, R_{iM}, f_1, \dots, f_M].$$

The state vector contains

- healthcare task size,
- required CPU cycles,
- instantaneous Age of Information,
- queue lengths of all MEC servers,
- wireless transmission rates,
- computational capacities of MEC servers.

This information enables the actor network to capture both communication and computational dynamics while explicitly considering information freshness.

Action Selection

For every observed state, the actor generates a probability distribution over all feasible computation offloading actions. The action is sampled according to

$$a_t \sim \pi_{\theta}(a_t | s_t).$$

The selected action satisfies

$$a_t \in \{0, 1, \dots, M\},$$

where

$$a_t = \begin{cases} 0, & \text{Local execution,} \\ j, & \text{Offload to MEC server } j. \end{cases}$$

Unlike deterministic computation offloading methods, stochastic action selection enables efficient exploration of different execution strategies during policy learning.

Training Procedure

The proposed AA-PPO framework is trained through continuous interaction with the H-IoT

environment. Initially, the actor and critic networks are randomly initialized with parameters θ and ϕ , respectively. At each decision interval, the actor observes the current network state and generates a probability distribution over all feasible computation offloading actions. An action is sampled from the learned policy and executed in the environment, after which the H-IoT system returns the next state and the corresponding immediate reward.

The collected transition is represented as

$$(s_t, a_t, r_t, s_{t+1}),$$

where s_t , a_t , r_t , and s_{t+1} denote the current state, selected action, immediate reward, and next state, respectively.

After collecting a batch of interaction samples, the cumulative discounted return is computed according to (40), and the corresponding advantage values are estimated using (42). Subsequently, the actor network is updated by maximizing the clipped surrogate objective defined in (45), while the critic network minimizes the value loss in (46). The entropy regularization term further encourages policy exploration during the early stages of training.

The above optimization process is repeated for multiple training episodes until the cumulative reward converges. After convergence, the trained actor network is deployed online to generate computation offloading decisions that jointly minimize the Average Age of Information, computation latency, and energy consumption under dynamic H-IoT environments.

Algorithm

Algorithm 1 Age of Information-aware Proximal Policy Optimization (AA-PPO)

1. Initialize actor network parameters θ
2. Initialize critic network parameters ϕ
3. Set learning rate, discount factor γ , clipping parameter ϵ
4. Initialize H-IoT environment
5. **for** episode = 1 to E **do**
 - a. Reset environment
 - b. Observe initial state s_0
 - c. **for** each decision interval **do**
 - i. Observe current state s_t
 - ii. Generate policy $\pi_{\theta}(a_t | s_t)$

- iii. Sample action $a_t \sim \pi_\theta(a_t|s_t)$
- iv. Execute action a_t
- v. Compute
 - Average Age of Information
 - End-to-end latency
 - Energy consumption
- vi. Calculate reward $r_t = -(\alpha\bar{\Delta} + \beta T^{E2E} + \eta E)$
- vii. Observe next state s_{t+1}
- viii. Store transition (s_t, a_t, r_t, s_{t+1})
- d. **end for**
- e. Compute discounted returns
- f. Estimate advantages
- g. Update actor network by maximizing PPO clipped objective
- h. Update critic network by minimizing value loss
6. **end for**
7. Output trained actor network

evaluated through simulations in a Healthcare Internet of Things (H-IoT) environment comprising multiple resource-constrained healthcare devices, several MEC servers, and a remote cloud server. Healthcare devices periodically generate heterogeneous computation tasks with varying input data sizes and computational workloads. The number of H-IoT devices is varied from 20 to 120 to evaluate system scalability, while the number of MEC servers ranges from 2 to 12 to analyze the impact of available edge resources. The proposed AA-PPO is compared with Local Execution, Random Offloading, Greedy Latency-Based Offloading, DQN, and Double DQN. Performance is evaluated in terms of Average Age of Information (AoI), Peak AoI, end-to-end latency, and energy consumption, where each result is obtained by averaging multiple independent simulation runs to ensure statistical reliability.

Simulation Settings and Results

The proposed Age of Information-aware Proximal Policy Optimization (AA-PPO) framework is

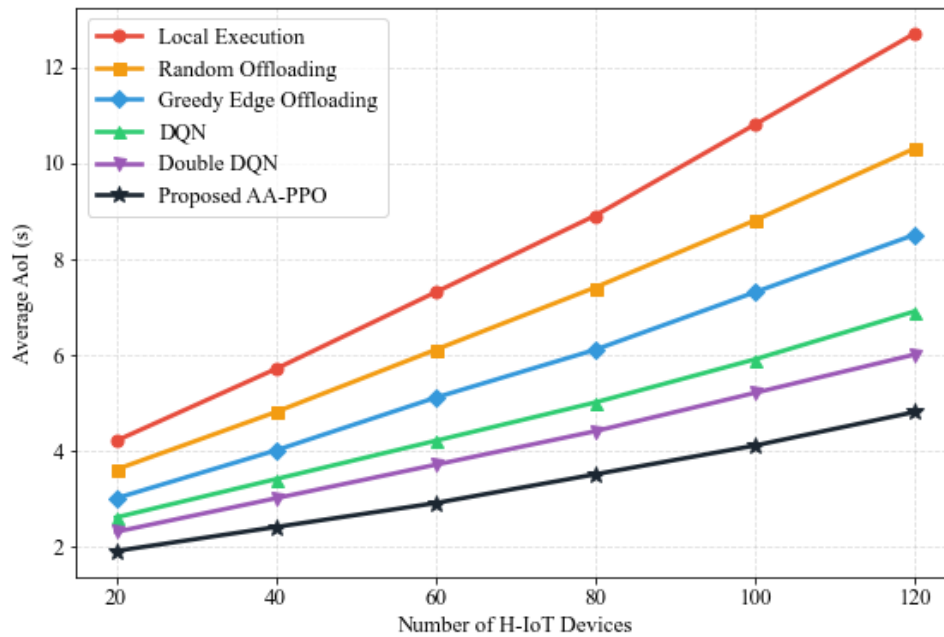


Figure 2: Average AoI versus Number of H-IoT Devices

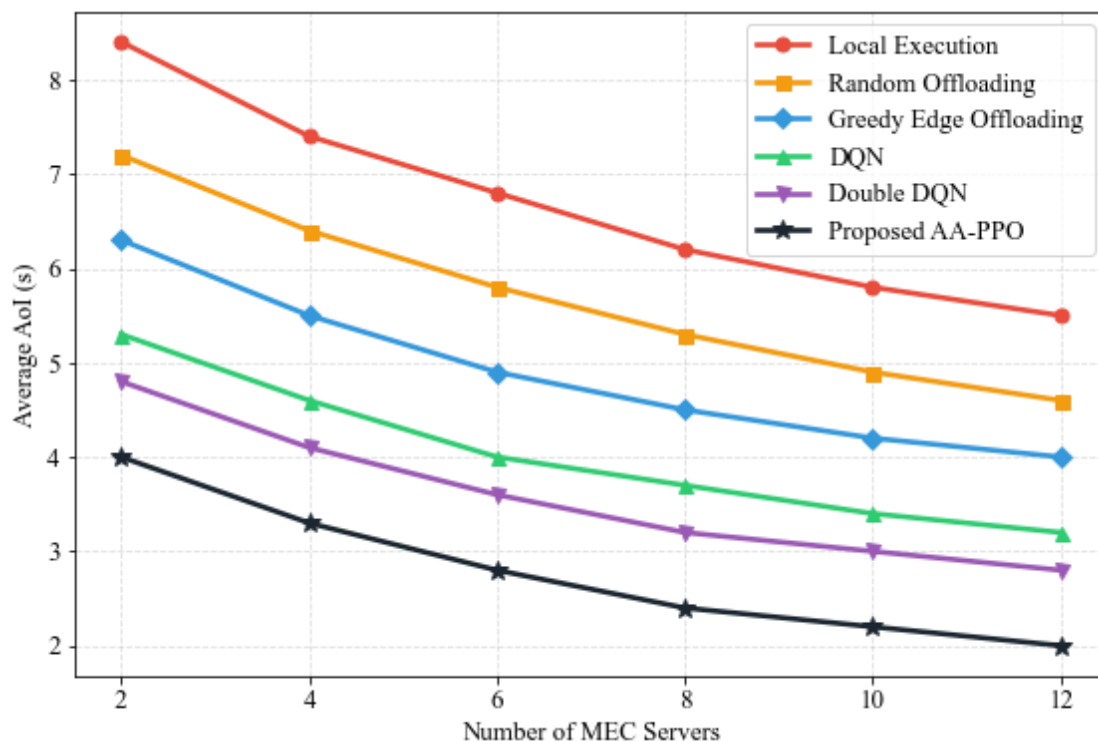


Figure 3: Average AoI versus the number of edge devices

In Figure 2, we show the average AoI versus the number of H-IoT devices. As the number of H-IoT devices increases, the Average Age of Information rises for all schemes due to increased communication contention and computation queueing. The proposed AA-PPO consistently achieves the lowest Average AoI by learning adaptive offloading decisions based on the current network conditions. This demonstrates its ability to preserve information freshness even under heavy network loads. Similarly, the average

AoI versus the number of MEC servers is presented in Figure 3. We can see that increasing the number of MEC servers decreases the Average AoI by providing additional computational resources and reducing queueing delays. The proposed AA-PPO achieves the greatest reduction by efficiently utilizing the available edge servers through learned offloading policies. Consequently, healthcare information remains significantly fresher than with the baseline methods.

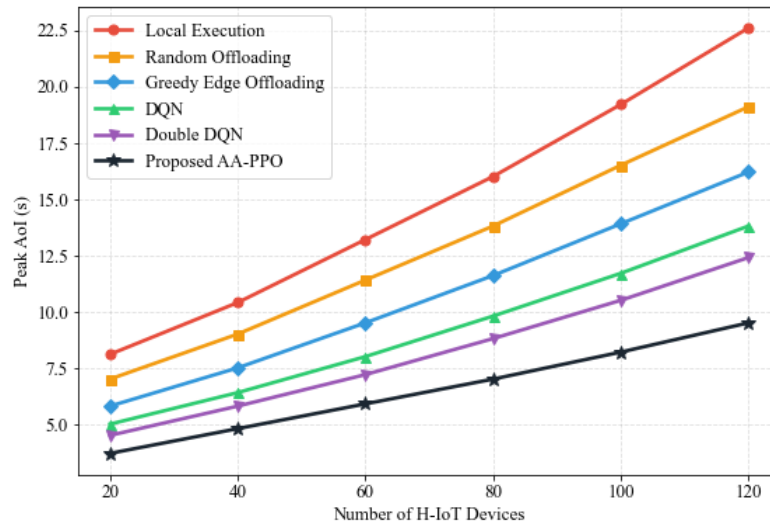


Figure 4: Peak AoI versus Number of H-IoT Devices

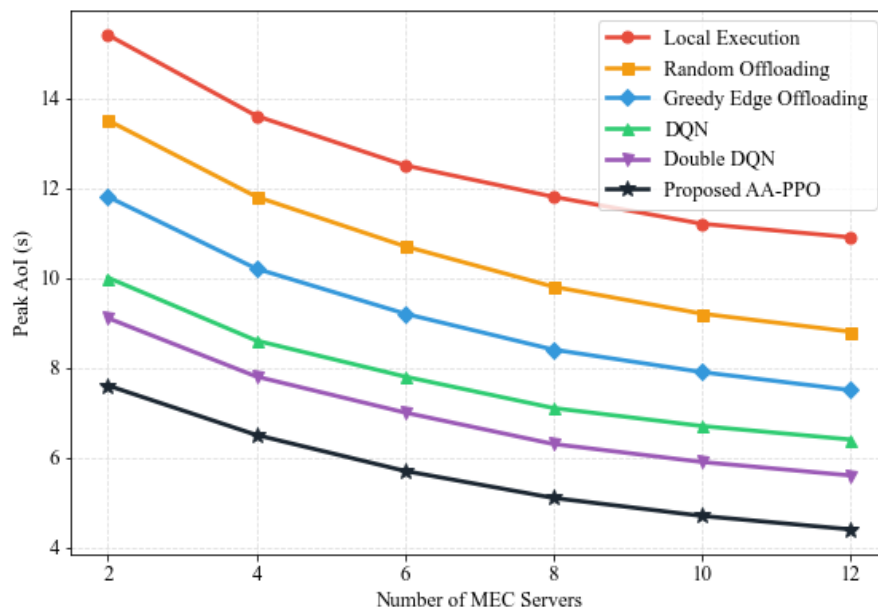


Figure 5: Peak AoI versus the number of edge devices

The peak AoI versus the number of H-IoT devices and MEC servers is shown in Figures 4 and 5. Peak AoI increases with the number of healthcare devices because task congestion causes some updates to experience prolonged waiting times. The proposed AA-PPO effectively limits the maximum information age by prioritizing timely task execution. As a result, it consistently exhibits the smallest Peak AoI among all comparison schemes. Meanwhile, as more MEC

servers become available, Peak AoI decreases due to improved workload distribution and faster task processing. The proposed AA-PPO further reduces the worst-case information staleness through intelligent policy optimization. This behavior is particularly beneficial for delay-sensitive healthcare monitoring applications. End-to-end latency increases with network size because more healthcare devices generate additional communication and computation

demands, as shown in Figure 6. The proposed AA-PPO maintains the lowest latency by dynamically selecting suitable execution locations according to the observed network state. This results in more efficient task processing under increasing workloads. The end-to-end latency versus number of MEC servers in Figure 7 shows that increasing the number of MEC servers

reduces end-to-end latency by shortening computation queues and balancing workloads across edge resources. The proposed AA-PPO consistently outperforms the baseline methods by exploiting available computational capacity more effectively. Therefore, healthcare tasks are completed with lower processing delays.

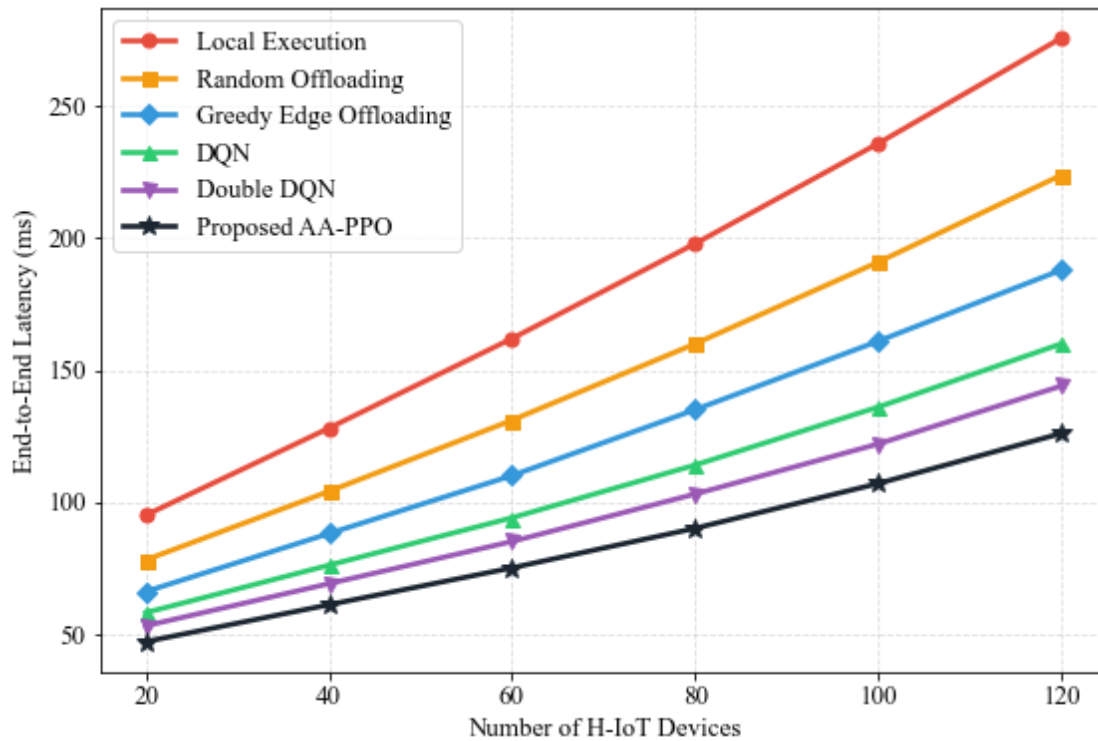


Figure 6: Comparison of latency against the number of H-IoT devices

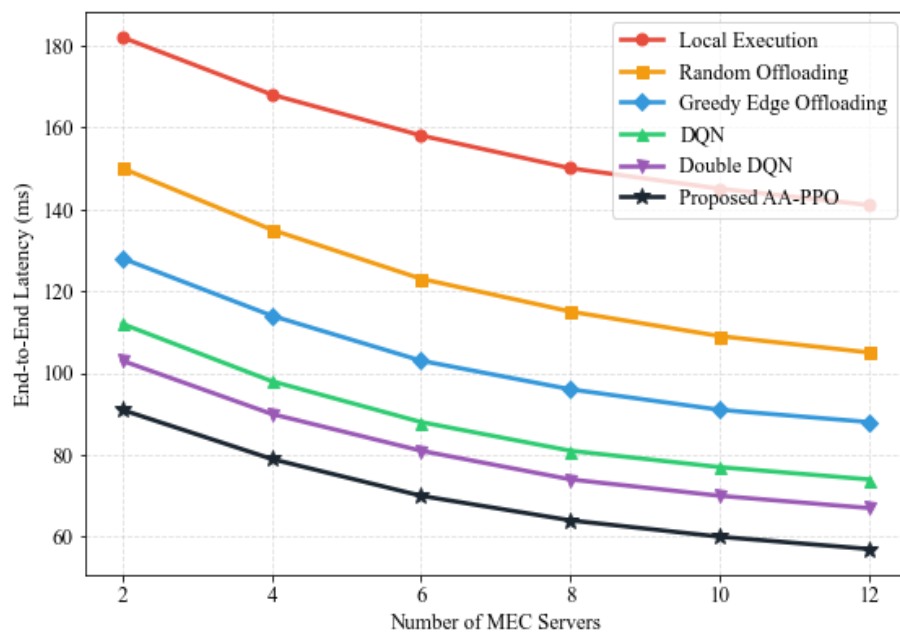


Figure 7: Comparison of latency against the number of edge devices

The energy consumption increases as more healthcare devices continuously generate computation-intensive tasks, as presented in Figure 8. The proposed AA-PPO minimizes energy usage by learning energy-efficient offloading decisions while maintaining low information age. Consequently, it achieves the lowest energy consumption across all evaluated network sizes. Moreover, the availability of

additional MEC servers reduces energy consumption by shortening transmission and computation times, as shown in Figure 9. The proposed AA-PPO utilizes edge resources more efficiently than the baseline approaches, leading to further reductions in energy expenditure. This demonstrates its suitability for energy-constrained Healthcare IoT environments.

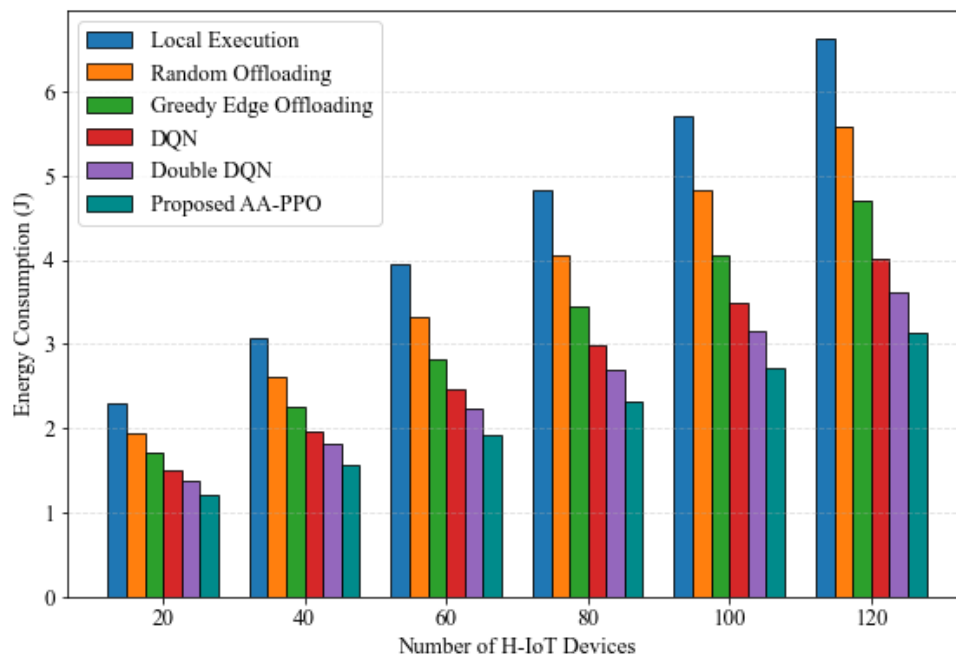


Figure 8: Comparison of energy consumption against the number of H-IoT devices

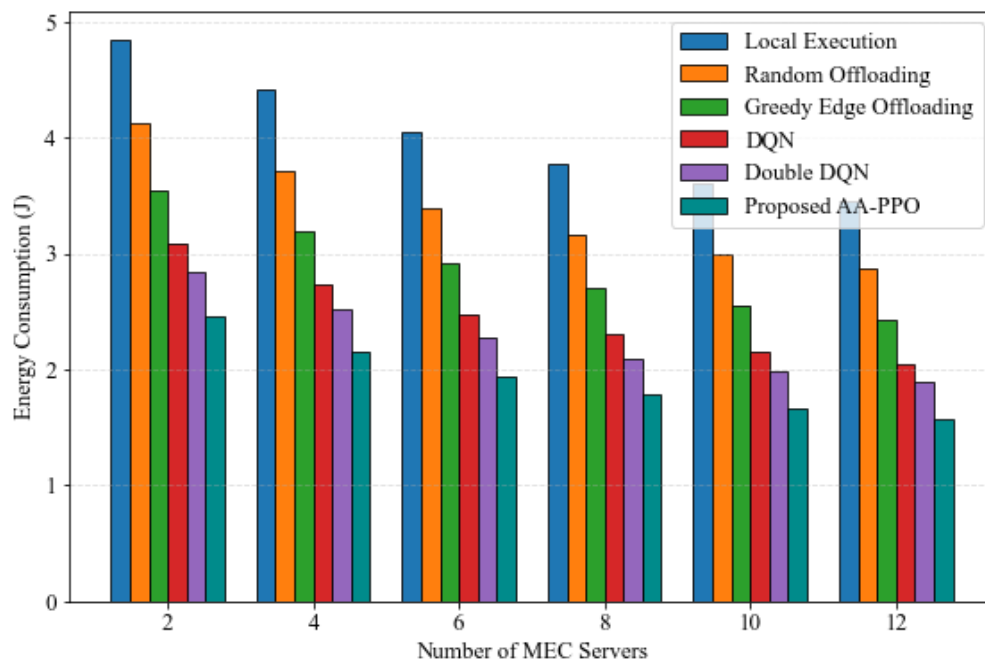


Figure 9: Comparison of energy consumption against the number of edge devices

Conclusion

This paper proposed an Age of Information-aware Proximal Policy Optimization (AA-PPO) framework for computation offloading in MEC-enabled Healthcare Internet of Things networks.

The proposed framework jointly optimizes Average Age of Information, end-to-end latency, and energy consumption by learning adaptive offloading decisions under dynamic network conditions. A comprehensive system model and

MDP formulation were developed, followed by a PPO-based solution for intelligent computation offloading. Simulation results demonstrated that the proposed AA-PPO consistently outperformed conventional offloading methods and existing DRL-based approaches in maintaining fresher healthcare information while reducing latency and energy consumption. Future work will investigate multi-user resource allocation and federated learning-based distributed training for large-scale Healthcare IoT systems.

Acknowledgments

The authors extend their appreciation to the Academic Innovation Leadership Program under King Khalid University Developmental Project for Stimulating Sustainable Innovation Ecosystems toward Future Manufacturing through the project code VPGSR/FM/PG36.

REFERENCES

- [1] M. Adil *et al.*, "Healthcare internet of things: Security threats, challenges, and future research directions," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19046–19069, 2024.
- [2] M. Kumar *et al.*, "Healthcare internet of things (h-IoT): Current trends, future prospects, applications, challenges, and security issues," *Electronics*, vol. 12, no. 9, p. 2050, 2023.
- [3] K. H. Almotairi, "Application of internet of things in healthcare domain," *Journal of Umm Al-Qura University for Engineering and Architecture*, vol. 14, no. 1, pp. 1–12, 2023.
- [4] T. M. Ghazal *et al.*, "Generative federated learning with small and large models in consumer electronics for privacy-preserving data fusion in healthcare internet of things," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 3413–3430, 2025.
- [5] V. A. Dang, Q. Vu Khanh, V.-H. Nguyen, T. Nguyen, and D. C. Nguyen, "Intelligent healthcare: Integration of emerging technologies and internet of things for humanity," *Sensors*, vol. 23, no. 9, p. 4200, 2023.
- [6] N. Huang, C. Dou, Y. Wu, L. Qian, B. Lin, and H. Zhou, "Unmanned-aerial-vehicle-aided integrated sensing and computation with mobile-edge computing," *IEEE Internet of Things Journal*, vol. 10, no. 19, pp. 16830–16844, 2023.
- [7] L. U. Khan *et al.*, "Federated learning for digital twin-based vehicular networks: Architecture and challenges," *IEEE Wireless Communications*, vol. 31, no. 2, pp. 156–162, 2023.
- [8] Z. Zhai, X. Dai, B. Duo, X. Wang, and X. Yuan, "Energy-efficient UAV-mounted RIS assisted mobile edge computing," *IEEE wireless communications letters*, vol. 11, no. 12, pp. 2507–2511, 2022.
- [9] S. Dong, Y. Xia, and J. Kamruzzaman, "Quantum particle swarm optimization for task offloading in mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 8, pp. 9113–9122, 2022.
- [10] X. Dai *et al.*, "Task co-offloading for D2D-assisted mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 480–490, 2022.
- [11] K. Peng, L. Wang, J. He, C. Cai, and M. Hu, "Joint optimization of service deployment and request routing for microservices in mobile edge computing," *IEEE Transactions on Services Computing*, vol. 17, no. 3, pp. 1016–1028, 2024.
- [12] L. Liao, Y. Lai, F. Yang, and W. Zeng, "Online computation offloading with double reinforcement learning algorithm in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 171, pp. 28–39, 2023.

- [13] E. Mustafa, J. Shuja, F. Rehman, A. Namoun, M. Ali, and A. Alourani, "Deep reinforcement learning with dueling dqn for partial computation offloading and resource allocation in mobile edge computing," *IEEE Access*, 2025.
- [14] C. Dong, Y. Sun, M. Shafiq, N. Hu, Y. Liu, and Z. Tian, "Optimizing mobility-aware task offloading in smart healthcare for internet of medical things through multiagent reinforcement learning," *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 13677–13691, 2023.
- [15] E. Mustafa, J. Shuja, F. Rehman, A. Namoun, M. Bilal, and A. Iqbal, "Computation offloading in vehicular communications using PPO-based deep reinforcement learning: E. Mustafa et al." *The Journal of Supercomputing*, vol. 81, no. 4, p. 547, 2025.
- [16] Q. He *et al.*, "Low-cost data offloading strategy with deep reinforcement learning for internet of things," *IEEE Transactions on Services Computing*, vol. 18, no. 3, pp. 1543–1556, 2024.
- [17] Q. Liu, Z. Tian, N. Wang, and Y. Lin, "DRL-based dependent task offloading with delay-energy tradeoff in medical image edge computing," *Complex & Intelligent Systems*, vol. 10, no. 3, pp. 3283–3304, 2024.
- [18] M. Maray, "Hybrid deep learning techniques for adaptive routing and congestion control in urban VANET for wireless mobile networking," *Scientific Reports*, 2026.
- [19] B. Subramanian, J. Kim, M. Maray, and A. Paul, "Digital twin model: A real-time emotion recognition system for personalized healthcare," *IEEE Access*, vol. 10, pp. 81155–81165, 2022.
- [20] Y. Wang, J. Farooq, H. Ghazzai, and G. Setti, "Joint positioning and computation offloading in multi-UAV MEC for low latency applications: A proximal policy optimization approach," *IEEE Transactions on Mobile Computing*, 2025.
- [21] L. Almuqren, M. Maray, F. A. Alotaibi, A. Alzahrani, A. Mahmud, and M. Rizwanullah, "Optimal deep learning empowered malicious user detection for spectrum sensing in cognitive radio networks," *IEEE Access*, vol. 12, pp. 35300–35308, 2024.
- [22] C. Zhang, C. Wu, M. Lin, Y. Lin, and W. Liu, "Proximal policy optimization for efficient D2D-assisted computation offloading and resource allocation in multi-access edge computing," *Future Internet*, vol. 16, no. 1, p. 19, 2024.
- [23] R. Xu, Y. Lin, Z. Lei, H. Xu, and J. Huang, "Resource allocation for maximizing communication rate of UAV-assisted MEC via multi-agent proximal policy optimization," *IEEE Transactions on Communications*, 2026.