Age-Aware Scheduling for Federated Learning with Caching in Wireless Computing Power Networks

Xiaochong Zhuang , Chuanbai Luo , Zhenghao Xie , Yu Li , Li Jiang *

Posted Date: 21 January 2025

doi: 10.20944/preprints202501.1482.v1

Keywords: Federated learning (FL); Wireless Computing; Power Network (WCPN); cache mechanism; parametric age; resource scheduling

Article

Age-Aware Scheduling for Federated Learning with Caching in Wireless Computing Power Networks

Xiaochong Zhuang [1], Chuanbai Luo [2], Zhenghao Xie [2], Yu Li [3] and Li Jiang [4],*

[1] Guangdong–Hong Kong–Macao Joint Laboratory for Smart Discrete Manufacturing and the Key Laboratory of Intelligent Detection and Internet of Manufacturing Things, Ministry of Education, Guangdong University of Technology, Guangzhou 510006, China
[2] Guangdong Provincial Key Laboratory of Intelligent Systems and Optimization Integration, Guangdong University of Technology, Guangzhou 510006, China
[3] Chongqing Key Laboratory of Intelligent Perception and BlockChain Technology, Chongqing Technology and Business University, Chongqing 40067, China
[4] 111 Center for Intelligent Batch Manufacturing Based on IoT Technology and the Key Laboratory of Intelligent Information Processing and System Integration of IoT, Ministry of Education, Guangdong University of Technology, Guangzhou 510006, China
* Correspondence: jiangli@gdut.edu.cn

Abstract: With the rapid development of the Wireless Computing Power Network (WCPN), the urgent need for data privacy protection and communication efficiency led to the emergence of the Federated learning (FL) framework. However, the time delay leads to dragging problems and reduces the convergence performance of FL in the training process. In this article, we propose a FL resource scheduling strategy based on information age perception in WCPN, which can effectively reduce the time delay and enhance the convergence performance of FL. Moreover, a data cache buffer and a model cache buffer are set up at the user and the central server respectively. Next, we formulate the parametric age-aware problem to minimize the age of global parameter, energy consumption and FL service latency simultaneously. Taking account of the dynamic WCPN environment, the optimization target is modeled as a Markov decision process (MDP), and Proximal Policy Optimization (PPO) algorithm is used to achieve the optimal solution. Finally, a large number of numerical simulation results prove the superiority and outstanding performance of the proposed method.

Keywords: Federated learning (FL), Wireless Computing Power Network (WCPN), cache mechanism, parametric age, resource scheduling

1. Introduction

With the rapid development of science and technology, wireless services such as cloud-based extended reality, holographic communication and intelligent interaction are becoming more prevalent. To provide real-time immersive user experience, the Wireless Computing Power Network (WCPN), as a highly integrated intelligent framework, is capable of delivering intense and adaptive computing services for consumers through the deep integration and utilization of distributed reachable computing resources[1–3]. Morever, WCPN can effectively cope with the challenge of load imbalance by decomposing computing tasks and coordinating between computing nodes [2,3], fully utilizing available computing resources, and providing fast and accurate computing services. However, WCPN largely relies on data, and the correctness of its decisions depends on the diversity of data used for training. Therefore, relying only on local data from devices to train is far from enough. And it is necessary to combine data from other devices to provide sufficient amounts and diversity of data. However, local data on devices is the privacy of users, sharing this data may be used for other malicious purposes, which deviates from the original intention of improving user experience. Recognizing these potential risks of privacy leakage, a distributed machine learning model, Federated learning (FL), has emerged [4–7]. Unlike traditional machine learning methods that run in data centers, FL only

 doi:10.20944/preprints202501.1482.v1

needs to send machine learning models to various intelligent devices for training. After training, smart devices upload trained models to central nodes, which then aggregate the models uploaded by various smart devices. The aggregated model is a comprehensive model that combines data from all smart devices. There is no interaction between device original data during this process, only the uploading of device machine learning models, which protects user data privacy security while achieving the goal of collaborative machine learning with multiple devices. Therefore, FL has been applied to enable distributed intelligence of WCPN, which guarantees the security of computing nodes and the privacy of computing devices [8].

In FL, the central node needs to connect with a large number of user devices and receive the distribute machine learning models in each global aggregation [5,9]. With the rapid development of WCPN, the number of intelligent devices is overwhelming and still increasing. The performance requirements of these devices for machine learning will certainly make the coverage of FL wider and the scale of machine learning models larger, which will exert tremendous pressure on the computing architecture. However, due to the limited spectrum resources, there are always some user devices that do not have enough spectrum resources to transmit machine learning models, resulting in slow or even failed transmission tasks. These users are called draggers or laggards, and their transmission latency has a decisive impact on the efficiency of FL. Therefore, dragger problems are indispensable in the research of FL [6,7,10]. To improve the efficiency of FL, some studies based on different design principles propose various device scheduling strategies, including reducing draggers and minimizing convergence time [11,12]. In addition, due to the heterogeneity of smart devices' computing capabilities , communication conditions and imbalanced service demands, it is usually possible for a single node to provide FL services, which may lead to device overloading and communication congestion [13,14]. To address this massive computational demand, W. Sun et al. combined asynchronous FL WCPN framework by jointly optimizing the computation strategy of individual computing nodes and their collaborative learning strategies to minimize the total energy consumption of all computing nodes [15]. This algorithm performs excellently in terms of learning accuracy, convergence speed, and energy conservation.

The architecture of WCPN enables users to receive dense and adaptive computing tasks without knowing the exact service deployment, they only need to provide their own service demands.The service delay caused by insufficient computing power, poor communication conditions, or other factors is extremely lethal for time-sensitive tasks, especially for those that require real-time monitoring and control. The interval between data generation and its invocation is a key factor affecting task performance, which is referred to as the Age of Information (AoI) [16]. AoI has become an important metric for measuring data freshness. There have been several studies on the impact of AoI on different optimization objectives, and they have sought to enhance data utilization efficiency and reliability by minimizing the AoI defined in this specific scenario [17–24]. To minimize the average age of critical information (AoCI) of mobile clients, X. Wang et al. built a system model based on request response communication. [25]. They proposed an information sensing heuristic algorithm based on Dynamic Programming (DP), which showed advantages in terms of average AoCI under various network parameters and had a short convergence time. Furthermore, W. Dai et al. studied the convergence behavior of extensive distributed machine learning models and algorithms with delayed update [26]. Through numerous experiments, it was demonstrated that delayed updates significantly reduce the convergence speed of FL global models due to the influence of Gradient Coherence. In other words, the staleness of each local model in FL affects the convergence speed of the global model, and outdated local models may negatively impact global model convergence when they update in opposite directions from the current global model.

Caching originates from computer systems, where copies of frequently used commands and data are stored in memory[27]. To enhance the commnucation efficiency of FL, several researches proposed the cache scheme to improve FL, which can effectively reduce the training time of FL by reducing per-iteration training time[28–30]. However, Existing research on FL with caching lacks in-depth

 doi:10.20944/preprints202501.1482.v1

exploration of the AoI. Besides, although excellent AoI can significantly improve the performance of time-sensitive machine learning models, the cost of pursuing excellent AoI in the FL framework is often to bring about significant service delays. Therefore, to improve the quality and models in WCPN, this paper proposed a FL with caching resource scheduling strategy, then focuses on optimizing resource coordination, improving service quality and reducing parameter age related to machine.

1.1. Relate Works

In many time-sensitive FL tasks, both the quality of data used for local training and the quality of uploaded models are closely related to their age. Outdated parameter ages can result in slower convergence rates for FL and reduce the performance of the global model. To tackle these issues, several recent studies have focused on optimizing the data parameter age during the device scheduling process.

In [31], the parameter age is incorporated into the device scheduling process in FL, and an age-aware wireless network FL communication strategy is proposed to prevent the devices from exiting training using energy collection technology. The fast and accurate model training can be achieved by jointly considering the aging of parameters and the heterogeneous capabilities of terminal devices. In [32], the authors propose a theoretical framework for statistical AoI provisioning in mobile edge computing (MEC) systems based on Stochastic Network Calculus (SNC) to support the tail distribution analysis of AoI. Additionally, they design a dynamic joint optimization algorithm based on block coordinate descent to solve the energy minimization problem. In [33], the authors focus on the timeliness of MEC systems, believing that the freshness of data and computing tasks is very important, and establish an age-sensitive MEC model, where the AoI minimum problem is defined. A new hybrid policy-based multimodal deep Reinforcement Learning (RL) framework with edge FL mode is proposed, which not only outperforms the baseline system in terms of average system age but also improves the stability of the training process. In [34], considering the time-sensitive IoT system with UAV assistance, the research on information freshness in the system is conducted, and a node data acquisition algorithm based on DDQN is proposed to effectively reduce the average information age of the system by optimizing the flight trajectory and transmission sequence of sensors. In [35], we propose FedAoI, an AoI based client selection policy. FedAoI ensures fairness by allowing all clients, including stragglers, to submit their model updates while maintaining high training efficiency by keeping round completion times short. In [36], the authors formulate the problem of multi-UAV trajectory planning and data collection as a Mixed Integer NonLinear Programming (MINLP), aiming to minimize the AoI and the energy consumption of IoT devices. The formulation takes into account the flight constraints of UAVs, the limitations of device data collection, and interference coordination conditions. In [37], considering the trade-off between data sampling frequency and long-term data transmission energy consumption while maintaining information freshness, a two-stage iterative optimization framework based on FL is proposed to minimize the global mean and variance of transmission energy, saving a large amount of execution energy with relatively small performance loss.

In addition to considering the impact of parameter age in federated machine learning, these works have achieved good optimization results in terms of service delay or energy consumption. However, they have not conducted in-depth research on time-sensitive data itself and have not fully considered the potential reference value of lagging user models, stopping at resource scheduling optimization strategies for shortening service delay. Based on these considerations, this paper designs a WCPN FL framework with dual caches that comprehensively considers parameter age, energy consumption, and service delay, and optimizes the iteration strategy. Besides, a WCPN FL resource scheduling strategy based on information age perception is proposed, which designs a dynamic scheduling strategy to enable as many users as possible to participate in FL in a timely manner by jointly considering the parameter freshness and heterogeneity of terminal devices.

1.2. Contributions and Paper Organization

The main contributions of this work can be summarized as follows.

 doi:10.20944/preprints202501.1482.v1

- To consider the trade-off between parameter age and service delay, we designed a FL resource scheduling strategy based on information age perception in WCPN, which takes advantage of data caching mechanisms. This strategy optimizes device's data collection frequency, computation frequency, and spectral resources to improve the global model performance in FL.
- In this paper, we comprehensively consider the aging of parameters, time-varying channels, random FL request arrivals, and heterogeneous computing capabilities among participating devices. We model the high-dimensional dynamic multi-user centralized FL framework system as a MDP, using the Proximal Policy Optimization (PPO) algorithm to solve for the minimization of global parameter age, energy consumption, and FL service delay.
- Sufficient comparative simulation experiments are designed to verify the correctness and superiority of our scheme, and the reasons are analyzed and demonstrated.

The remainder of this article is organized as follows. Section II proposes a dual-cache FL framework with local data caching and server model caching, and present the problem of the WCPN FL resource scheduling strategy based on information age perception in this work. Section III formulates the MDP for the target problem and then construct an age-aware PPO double-cache FL scheduling algorithm to solve the corresponding optimal solution. Section IV demonstrates the performance of our proposed approach through extensive simulation studies, providing simulation results and theoretical analysis. Finally, we summarize this paper in Section V.

2. System Model and Problem Formulation

In related research on conventional FL, it is usually assumed that intelligent terminals collect data in advance for model training. However, this can lead to a certain degree of aging of the data. This case of higher AoI is very unfriendly for some tasks that are sensitive to AoI. In addition, collecting data only when receiving FL requests may ensure sufficient freshness, but the service delay during data collection may be unbearable. Therefore, it is necessary to make a balance between service delay and information age and coordinate their needs reasonably.

Our system model, as shown in Figure 1, consists of a central server and K smart devices which have data collection and local computing capabilities. These devices share limited spectrum resources to interact with the central server for local model upload. The model owner initiates a FL task involving a set of smart devices $\mathcal{K} = \{1, ..., k, ..., K\}$. During the FL task, there can be multiple instances of model training requests initiated by the model owner, and each participating intelligent device uses its local latest data to train the model together to maintain the global model. This paper assumes that each instance arriving at the request follows a Poisson process [38]. First, FL-based model training is initiated based on the request from the model owner, with each model training taking place in N iterations to minimize the global loss $F(w^N)$, where N is specified by the model owner and $n = \{1, ..., n, ..., N\}$. Each iteration is composed of three steps in sequence:

 doi:10.20944/preprints202501.1482.v1

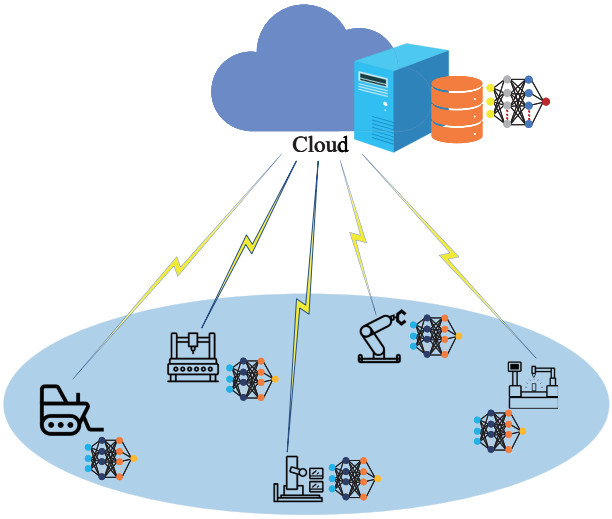

Figure 1. Decentralized edge federated modeling framework in WCPN.

Step 1: Local Train. The smart device k uses the locally collected and processed data to train the received global model w^n to obtain the local model w_k^n.

Step 2: Model Upload. The smart device transmits the model parameters w_k^n to the central server.

Step 3: Global Model Updates. All the model parameters received from K smart devices are aggregated to obtain a new global model w^{n+1}, which is then sent to each intelligent device for $n+1$ iterations.

In the synchronous FL scheme, the time taken for each iteration is limited by the slowest device. The central server needs to wait for all smart devices to complete local training, and then aggregate global model. Smart devices also need to obtain new global models to carry out the next round of iteration. To effectively reduce the service delay of FL (i.e., the delay from responding to FL to completing model uploading), this paper designs an age-aware dual-cache FL framework. Each smart device has a data cache buffer to pre-collect local training data. Meanwhile, the central server sets up a cache buffer to save models uploaded by lagging devices.

2.1. AoI and Service Latency Model

Since the diversity of uploaded parameters is crucial for training performance with non-independent distributed data, we utilize a bandwidth dynamic scheduling strategy to enable as many users as possible to participate in FL in a timely manner. In this paper, we propose a wireless network FL resource scheduling strategy based on information age awareness, which enables rapid and accurate model training by jointly considering the parameter freshness and heterogeneous capabilities of end devices. We jointly optimize AoI and resource allocation strategies to achieve a trade-off between FL accuracy, training time, and user energy consumption. Considering the potential impact of current scheduling on subsequent training and available resources, we describe the optimization problem as a MDP.

2.1.1. Local Delay and Age of Local Data

In Figure 2, the time period during which device k waits for FL training requests to complete local training includes idle time, data collection and processing time, and local model parameter training time. In traditional FL, users only start collecting training data and performing local training after receiving FL training requests, as shown in Figure 2(a). The time delay for user k to perform local model training is represented by $T_k^{tra}(t)$, while the time delay for device k to collect required data samples is represented by $T_k^{col}(t)$. Starting from the beginning of data collection and completing it before starting local training can ensure absolute data freshness, but it will result in significant delays, with a corresponding local delay of $T_k^{tra}(t) + T_k^{col}(t)$.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 January 2025 doi:10.20944/preprints202501.1482.v1

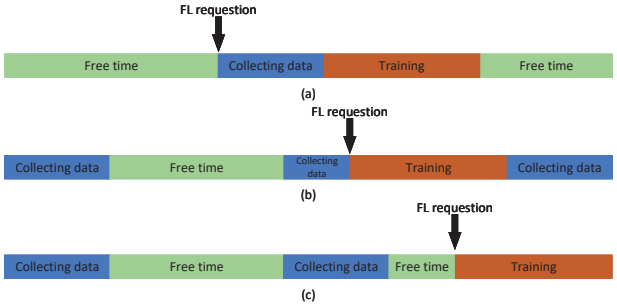

Figure 2. FL service delay in different cases.

Suppose the total Floating Point Operations(FLOPs) needed to process one data sample is G_k^{per}, and the FLOPs of each CPU cycle of device k is C_k. Then, the time for device k to process one data sample can be denoted by:

$$T_k^{per} = \frac{G_k^{per}}{C_k f_k^{com}} \tag{1}$$

In this paper, we consider designing a data buffer mechanism for FL processes. Devices should adjust the data collection frequency f_k^{col} based on intelligent algorithms and collect data in intervals of time $f_k^{col} T_k^{per}$ during idle periods. As shown in Figure 2(b) and Figure 2(c), this approach can save the data collection process and quickly respond to FL while ensuring data freshness, with a local delay of $T_k^{tra}(t)$.

Assuming that device k needs to collect N_k^{per} units of data samples for local training, data collection requires continuously collecting data and sending each group of data sequentially into the data buffer. If device k has completed data collection processing and still hasn't received FL training requests after waiting for $f_k^{col} T_k^{per}$ time, then new data will be collected to remain the freshness of data in the buffer. According to the First In First Out (FIFO) principle, the latest collected unit of data is sent to the tail of the buffer queue, and the unit of data at the head of the buffer queue will be discarded. As shown in Figure 2(b), during the new data collection process, there are always enough data samples available in the local system to quickly respond to FL requests during the data collection period.

To quantify the freshness of each unit of data, device k ages its data every time it passes through a time period T_k^{per} during idle time, which means:

$$A_{k,n}^{loc}(t+1) = A_{k,n}^{loc}(t) + 1 \tag{2}$$

where $A_{k,n}^{loc}(t)$ represents the freshness of device k's nth group of data at time t. After receiving an FL request and completing local model training, device k uploads the latest local model parameters ω_k to the central server. In the process of uploading model parameters, the parameter upload rate cannot remain efficient and stable due to the dynamic changes in the geographical location and communication environment of device k. Moreover, due to the instability of communication conditions, the probability that many devices fall behind during parameter upload increases significantly. Therefore, it is necessary to set a receive waiting time threshold T_a to limit the waiting time for the central server to receive device k's model. Let $T_k^{ser}(t) = T_k^{tra}(t) + T_k^{up}(t)$. For device k, its model can only be successfully aggregated in this round if $T_k^{ser}(t) T_a$. Assuming that device k receives an FL request at time t_k^{fl}, the sum of all local data ages of device k at this moment is $\sum_{n \in N_k^{per}} A_{k,n}^{loc}(t_k^{fl})$, and the model age will be calculated based on this basis, the optimization objective can be denoted as:

$$A_k^{mdl}\left(t_k^{fl}\right) = \sum_{n \in N_k^{per}} A_{k,n}^{loc}\left(t_k^{fl}\right) \tag{3}$$

 doi:10.20944/preprints202501.1482.v1

2.1.2. Transmission Delay and Age of Model

Considering the diversity of data samples is beneficial for improving model performance, while the freshness of device $k's$ local model also affects the convergence speed of the FL global model. Therefore, it is necessary to make reasonable use of limited bandwidth resources to enable as many models as possible that meet the age requirements to participate in global aggregation. For each device who responds to an FL request, they will upload their trained local model to the central server for aggregation of new global models. Due to dynamic changes in devices' available computing resources and communication conditions, there is uncertainty in the delay between responding to an FL request and successfully uploading the local model to the central server. Therefore, due to the impact of local training and model upload delays, devices may risk missing multiple rounds of global aggregation. The Age of user k's local model $A_k^{mdl}(t)$ increases with the increase in global iteration times until the model is used for global aggregation or discarded due to being too old, which is denoted as:

$$A_k^{mdl}(t + T^{fl}) = A_k^{mdl}(t) + 1 \tag{4}$$

where T^{fl} is the time interval between two global iterations. Considering the impact of model utilization on its age, this paper sets a model freshness threshold A_a^{mdl} to limit models that exceed this age threshold and will be discarded from global model aggregation. Combining the constraints on service latency mentioned above, only when both conditions $T_k^{ser}(t)T_a$ and $A_k^{mdl}(t)A_a^{mdl}$ are satisfied at the same time, will $C_k(t) = 1$ be set, otherwise $C_k(t) = 0$, can be denoted as:

$$C_k(t) = \begin{cases} 1, \ t_k^{loc} + t_k^{up} \leq T_a \& A_k^{mdl}(t) \leq A_a^{mdl}, \\ 0, \ else. \end{cases} \tag{5}$$

Based on the above analysis, the latest global model age for global aggregation is:

$$A^g(t) = \frac{1}{K} \sum_{k \in K} C_k(t) A_k^{mdl}(t) \tag{6}$$

2.2. Energy Consumption Model

Considering that the central server has sufficient resources, we ignore energy consumption during the global model's downlink process. In our model, energy consumption is mainly reflected in the local computation and model upload processes of smart devices. Given the heterogeneous computing resources and communication conditions of different smart devices, the time taken by each device for the processes of local computation and model upload is different, resulting in different energy consumption. In the following, we will model the energy consumption of these two processes separately.

2.2.1. Energy Consumption in Local Train

The calculation frequency of device k can be approximately represented as:

$$f_k^{com}(t) = \frac{1}{\varepsilon_{1,k}} \cdot v_k(t) \tag{7}$$

where $\varepsilon_{1,k}$ is the coefficient determined by the chip structure of device k, and v_k is the instantaneous voltage applied to the chip by device k, and a threshold protection is applied to prevent damage to the chip due to high voltages for the purpose of preserving its lifetime. That is, $v_k v_m$. The power consumed by this device under the given calculation frequency f_k^{com} can be calculated as follows:

$$P_k^{loc}(t) = \varepsilon_{2,k} v_k^2(t) f_k^{com}(t) = \varepsilon_k \left(f_k^{com}(t)\right)^3 \tag{8}$$

where $\varepsilon_{2,k}$ can be considered as a constant determined by physical factors such as process and architecture of the chip, and is denoted as $\varepsilon_k = \varepsilon_{2,k} * \varepsilon_{1,k}^2$. The number of FLOPs for local training by devices

 doi:10.20944/preprints202501.1482.v1

k during one iteration is represented by $G_k^{tra}(t)$. The latency of local training during one FL iteration when device k participates with data buffering mechanism can be calculated as follows:

$$T_k^{tra}(t) = \frac{G_k^{tra}(t)}{C_k(t) f_k^{com}(t)} \tag{9}$$

In addition, the computation frequency $f_k^{com}(t)$ of device k during data collection will be consistent with that during local training. Let n_k^{fl} be the total number of collected data samples within the time interval between two global iteration T^{fl} for device k. Then, the local-phase energy consumption (including data collection and model training processes) of this devices can be calculated as follows:

$$E_k^{col}(t) = \left(n_k^{fl} T_k^{per} + T^{tra}(t) \right) P_k^{loc}(t) \tag{10}$$

2.2.2. Energy Consumption in Model Upload

Assuming that the maximum available bandwidth within the coverage area of the central server is B_m. If device k is assigned with the bandwidth $B_k(t)$, its data upload rate can be calculated as follows:

$$R_k(t) = B_k(t) \log_2(1 + \tau_k(t)) \tag{11}$$

where $\tau_k(t)$ is the signal-to-noise ratio from device k to the central server, and $D_k(t)$ bits is the size of the local model of device k. The latency for uploading the model of device k can be calculated as follows:

$$T_k^{up}(t) = \frac{D_k(t)}{R_k(t)} \tag{12}$$

Let the transmission power during upload process be $p_k^{up}(t)$. Then, the energy consumption of device k uploading its local model can be calculated as follows:

$$E_k^{up}(t) = p_k^{up}(t) \frac{D_k(t)}{R_k(t)} \tag{13}$$

The global iteration of the current round will only aggregate models that meet the requirements. The aggregated model is only related to the required devices, so the energy consumption generated during this iteration can be simply calculated by summing up the energy consumption of the corresponding devices, represented as:

$$E^g(t) = \frac{1}{K} \sum_{k \in K} C_k(t) E_k^{col}(t) E_k^{up}(t) \tag{14}$$

2.3. Problem Formulation

We have set up data buffer and model buffer in both the device and server sides respectively. To ensure the performance of the models, we can ensure the data in each buffer is fresh enough by reducing the service latency of FL [39]. In addition to age of parameter and FL service latency, system energy consumption is also an important factor that demands consideration. Therefore, can be denoted as:

$$U(t) = A^g(t) + E^g(t) + T^{ser}(t) \tag{15}$$

To design frequency adjustment strategies for devices' data collection, local computation, and bandwidth allocation, and to coordinate the global parameter age, energy consumption, and FL service latency. Therefore, the problem of equtation (15):

 doi:10.20944/preprints202501.1482.v1

$$\begin{aligned} &\min_{f_k^{col}, f_k^{com}, B_k} \quad U(t) \\ &s.t. C1: B_k(t) > 0, \forall k \in \mathcal{K} \\ &C2: \sum_{k=1}^{K} B_k(t) \leq B_m \\ &C3: f_k^{com}(t) \leq f_k^m(t), k \in \mathcal{K} \end{aligned} \tag{16}$$

where $C1$ ensures that each device has available spectrum resources for uploading their local models. $C2$ requires that the sum of global allocated spectrum resources to devices does not exceed the maximum available value globally. $C3$ requires that the adjusted computation resources of devices at current time cannot exceed the maximum available computation resources at the same time.

3. Algorithm

Considering the dynamic and high-dimensional nature of scenarios, it is very difficult to solve problem (16) using traditional methods. In this section, the optimization problem (16) is modeled as a MDP and deep reinforcement learning models are introduced for solving it. Q-learning and DQN are popular value-based RL methods that learn the action value function Q(s, a) related to the system's rewards or penalties. However, when the action space becomes too large, finding the best action becomes extremely difficult. To overcome the complexity of the action space, a policy-based approach is introduced.

PPO is an on-policy RL algorithm that improves the Policy Gradient (PG) algorithm by retaining its excellent performance in continuous state and action spaces. The PG algorithm is updated by calculating the policy gradient estimate and maximizing the policy value using gradient ascent methods [40]. PPO is parameterized by a set of parameters that parametrize the policy, replacing deterministic policies in value-based reinforcement learning with probabilistic distributions. Different actions are sampled from the return action probability list and optimized using neural network methods. It was developed from the Additive Advantage Actor-Critic (A3C) algorithm in 2017. The core idea of PPO is to limit the policy update range within a certain offset to avoid unstable problems caused by large or small updates, effectively balancing simplicity, sample complexity, and tuning difficulty.

Considering the complexity and high dynamics of the scenario in this article, PPO algorithm will be used to solve the MDP problem and optimize the objective. The algorithm flow is shown in Figure 3. In the designed algorithm, FL resources in WCPN will be scheduled and optimized. The algorithm flow is as follows: **Step 1:** WCPN sends the environment state to the PPO agent; **Step 2:** The PPO agent outputs the optimal decision and evaluation value based on the current state, and returns the decision to the WCPN environment for execution; **Step 3:** The WCPN environment returns the reward after executing the decision; **Step 4:** The PPO agent saves relevant data to the experience replay buffer; **Step 5:** When the amount of data in the experience replay buffer reaches a preset value, the parameters of the Actor and Critic networks in the PPO agent are updated.

 doi:10.20944/preprints202501.1482.v1

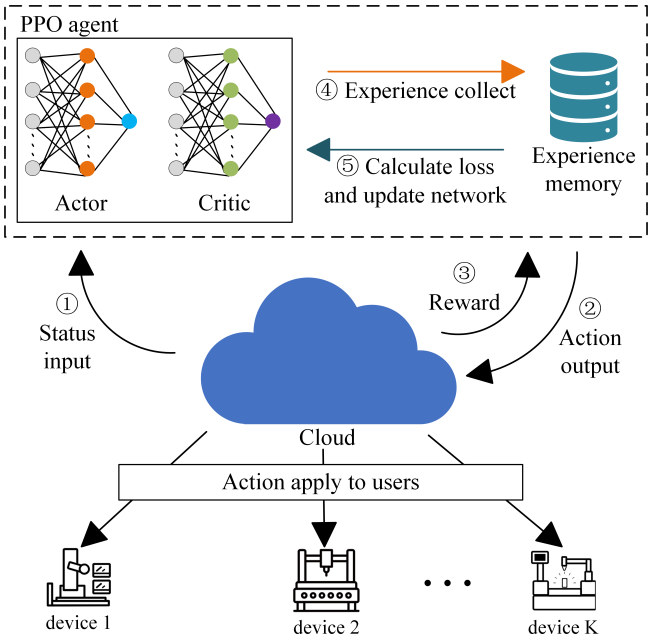

Figure 3. Algorithm working flow in system.

3.1. MDP Modeling for Optimization Problems

The MDP process is a commonly used model for designing environmental interaction systems, and often applied for reinforcement learning and dynamic. In an MDP, an agent learns the optimal strategy through interaction with the environment to maximize its expected return.

The core idea of MDP is to model the problem as a dynamic process in which an agent needs to take actions in a constantly changing environment and adjust its behavior based on feedback from the environment. However,the environment in an MDP can be uncertain, meaning that an agent may need to take different actions to respond to different situations.

3.1.1. State

The state $S(t)$ includes the amount of data that each device still needs to collect upon arrival of an FL request, the amount of data collected by each device since their last response to an FL request, the signal-to-noise ratio (SINR) at the current moment, and the time interval between the last two FL requests. Therefore, the state space at decision epoch t can be expressed by

$$S(t) = \{w_k^n, w^n, \tau_k(t), T^{fl}\} \tag{17}$$

3.1.2. Action

Based on the current environmental state $S(t)$, the action $A(t)$ adjusts the frequency of data collection for all device $k \in \mathcal{K}$, the local computation frequency, and the spectrum resources allocated for uploading the local model, i.e., $f_k^{col}(t), f_k^{com}(t), B_k(t), k \in \mathcal{K}$. Similarly, the action of the decision epoch t can be denoted as

$$A(t) = \{f_k^{col}(t), f_k^{com}(t), B_k(t)\} \tag{18}$$

3.1.3. Reward

Considering the age of data samples and model parameters during the FL iteration process, as well as the tradeoff between computational energy consumption, transmission energy consumption, and service latency, we aim to minimize the age while ensuring that energy consumption and latency are within acceptable ranges. We cannot sacrifice computing resources and spectrum resources for reducing parameter age at the expense of a large number of data samples and model parameters. In combination with our optimization objective (16), the goal of the reinforcement learning agent is to

 doi:10.20944/preprints202501.1482.v1

maximize the expected reward $r(t)$. Therefore, the reward of the agent can be described in a negative form of (15), which is:

$$r(t) = -U(t) \tag{19}$$

3.2. Double Cache FL Scheduling Algorithm of Age of Parameter Base on PPO

The PPO algorithm is a reinforcement learning algorithm based on policy gradient, aiming to optimize the policy by finding the optimal set of parameters θ^* and making the best decisions to maximize expected returns. The neural network of PPO consists of Actor network and Critic network. The Critic network is a value function network which estimates the value of each state, while the Actor network calculates the probability distribution of actions under different states.

To prevent overshooting, where the policy falls into a worse-performing region and fails to provide sufficient effective information in the future, the PPO algorithm also quantifies the optimization of the policy to ensure that each update has some degree of effectiveness. By updating the policy parameters through calculating the advantage function, a new policy can become more excellent. According to the advantage estimation method proposed by Mnih et al. [41], the Actor is run within a minibatch of logical decision slots and samples are collected for advantage estimation, which can be denoted as:

$$\widehat{A}(t) = \delta(t) + \gamma\delta(t+1) + ... + \gamma^{T-t+1}\delta(T-1) \tag{20}$$

where γ is the discount factor, $V(t) = Q(S(t); \theta_c)$ is the state value function, and $\delta(t) = R(t) + \gamma V(t+1) - V(t)$ is the advantage function. To make up for the shortcomings of the PG algorithm, PPO algorithm introduces off-policy in parameter updates to ensure that a set of data obtained through sampling can be reused. At the same time, using a clipped alternative objective ensures the change magnitude between new and old policies. Let $r_t(\theta) = \frac{\pi_\theta(A(t)|S(t))}{\pi_{\theta_{old}}(A(t)|S(t))}$ represent the difference between the policy $\pi_\theta(A(t)|S(t))$ before and after updating. The objective function of PPO can be expressed as:

$$L(\theta_a) = \widehat{\mathrm{E}}_t[\min(r_t(\theta_a)\widehat{A}(t), clip(r_t(\theta_a), 1-\varepsilon, 1+\varepsilon)\widehat{A}(t))] \tag{21}$$

where ε is a hyperparameter used to limit the magnitude of policy updates. The clip function, which clips $r_t(\theta_a)$ between $[1-\varepsilon, 1+\varepsilon]$, can be used to restrict the update magnitude of the policy. Additionally, the optimization objective of the Critic network is to minimize the difference between the estimated value and the actual reward, and its loss function can be represented as:

$$L(\theta_c) = \widehat{\mathrm{E}}_t\left(||V(t) - R(t)||^2\right) \tag{22}$$

During the training process of PPO algorithm, the agent obtains the initial state $S(0)$ from the environment. Based on the current state, it generates a random policy distribution and samples an action to apply to the FL of WCPN and the new state and action transition sequence are obtained. With each minibatch step of data collected, the alternative loss is constructed using these data, and the Actor-Critic network parameters θ_a and θ_c are updated using the Adam optimizer to obtain the optimal policy parameters θ^*. Algorithm 1 provides pseudocode for the parameter age-aware PPO double buffer FL scheduling algorithm.

 doi:10.20944/preprints202501.1482.v1

Algorithm 1: Parameter age-ware PPO Double Buffer FL Scheduling Algorithm

Initialization state $S(0)$;
Initializes the Actor neural network parameters θ_a and Critic neural network parameters θ_c for PPO agents;
for *Each episode* **do**
  Reset the system environment;
  for *Each decision epoch t* **do**
    Get the latest $S(t)$, and input it to the agent, which outputs the corresponding action $A(t)$;
    Execute the action $A(t)$ to obtain the next state $S(t+1)$ and reward $r(t)$;
    Store $\{S(t), A(t), r(t), S(t+1)\}$ in the experience replay buffer;
    if $t\%minibatch == 0$ **then**
      Calculate the advantage value $\hat{A}(t)$ according to equation (20);
      Calculate the gradients and update θ_a and θ_c according to equations (21) and (22);
      Update θ_{old}: $\theta_{old} \leftarrow \theta_a$;
    end
  end
end

4. Experiment

This section evaluates the performance of the proposed parameter-aware PPO double buffer FL scheduling algorithm through experimental testing, and sets up multiple comparison schemes to verify the correctness and superiority of the proposed scheme.

4.1. Experiment Setting

In this paper, the PPO algorithm is built using the TensorFlow neural network framework. The Actor network consists of an input layer, three hidden layers, and one output layer, while the Critic network consists of an input layer, two hidden layers, and one output layer. The PPO algorithm is set with the clipping parameter $\epsilon = 0.2$, the learning rate for the Actor network $lr\_actor = 1e-4$, the learning rate for the Critic network $lr\_critic = 2e-4$, and the discount factor $\gamma = 0.99$. The maximum number of training rounds in RL is $MAX\_EPOSIDE = 3000$, each round has an iteration of $epoch = 64$, and the minibatch size in PPO algorithm is 64.

In addition, we also validates the FL performance of the age-aware PPO double buffer scheduling algorithm using a real dataset MNIST. MNIST dataset consists of handwritten digit images and labels from 0 to 9. It includes 60,000 training images and 10,000 testing images. During simulation validation, devices use convolutional neural networks as the machine learning model for FL. They randomly collect data samples from MNIST data and conduct local training. Following the introduction in Section 2, this paper assumes that the arrival rate of FL requests follows a Poisson distribution. Considering a scenario where a central server and $K = 6$ devices jointly complete the FL task, the main environment parameters defined in Table 1 are shown below.

 doi:10.20944/preprints202501.1482.v1

Table 1. MAIN PARAMETER SETTINGS FOR SIMULATIONS

Parameter	Description	Value
K	Devices number	6
N	Local data buffer length	1000
A_a^{mdl}	Threshold age of model	150
T_a	Threshold delay for sever receive model	30s
ε_k	Device k's chip structure coefficient	[0.38988, 0.60998]
$C_k(t)$	Device k's chip clock frequency	[20, 50] FLOPS
$f_k^m(t)$	The maximum available computing resources of device k at t	[20, 50] GHz
$\tau_k(t)$	SINR between device k and server at t	[120, 130]
B_m	Global maximum available bandwidth	100M

4.2. Numerical Results

We set up multiple comparison schemes to verify the correctness and superiority of the proposed solution from different perspectives in simulation. There are a total of six different comparison schemes, and the meanings of these comparison schemes are as follows: (i) OnlyModelCache: Set only the model cache buffer in the central server to save models for dropped devices; (ii) OnlyLocalCache: Set only the local data cache buffer for each device to pre-cache data samples needed for local training; (iii) WithoutAnyCache: Do not set any cache buffers, which is the case of FL in a general scenario; (iv) FixFcol: Fix the device's data collection frequency and optimize only the device's computation frequency and allocated bandwidth size; (v) FixFcom: Fix the ratio of the device's computation frequency used for data collection and local computing to the maximum available computation frequency, and optimize only the device's data collection frequency and allocated bandwidth size; (vi) FixBandwidth: Fix the device's allocated bandwidth size and optimize only the device's data collection frequency and computation frequency.

As shown in Figure 4, as the number of iterations increases, all four algorithms can effectively reduce the FL service delay. From the reward equation(19), it can be seen that the FL delay is also an important objective in optimization. As the PPO agent solves for maximizing the reward, it also considers minimizing the FL delay, which is consistent with the goal of minimizing FL delay set in this paper. Two schemes with local data caching buffers have the lowest FL service delay. The performance of only model caching and no caching scheme is almost the same, indicating that model caching does not have a negative impact on FL service delay. The simulation results demonstrate the effectiveness of data caching buffers in reducing FL service delay. Although adding a model caching buffer will not significantly affect FL service delay, on the other hand, it can provide more device machine learning models for the global model, resulting in better global model performance and stronger generalization ability.

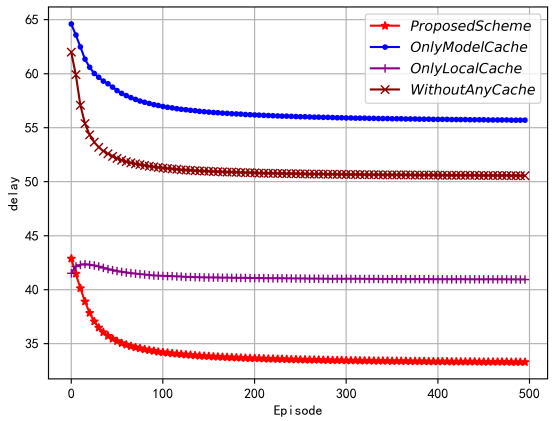

Figure 4. Service latency in different cache schemes.

 doi:10.20944/preprints202501.1482.v1

Figure 5 shows the relationship between the agent's reward and the number of iterations for the four schemes. Since the proposed scheme and the local data caching scheme can significantly reduce the service delay of FL, these two schemes also have obvious advantages in terms of rewards. Although only model caching and no caching schemes collect data after receiving FL requests, their data is the freshest, and the age of parameters involved in local training will be very low. However, the performance of these two schemes in terms of FL service delay is worse, as fresh data cannot offset the negative impact caused by high service delay, so their rewards are also lower.

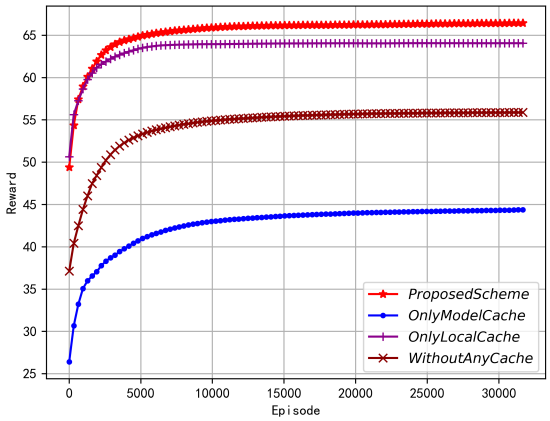

Figure 5. RL reward in different cache schemes.

As shown in Figure 6, as the number of iterations increases, the rewards for all four optimization schemes gradually increase and tend to converge. According to the definition in equation(19), the optimization objective defined by equation(15), is a negative exponential term of the reward, which means that the maximum reward of the agent corresponds to the minimization of equation(15). This is consistent with the goal of minimizing FL delay set in this paper. Compared with the other three schemes, the proposed scheme can obtain the largest reward. As the optimization variables in the proposed scheme are relatively more, its convergence speed is slightly slower, but its reward value has a significant advantage relative to other schemes. Other schemes lack optimization of necessary variables, although they can also converge to good rewards, their final results are far worse than those proposed in this paper. In conclusion, the joint optimization scheme of the three variables proposed in this paper is reasonable, correct and significantly superior.

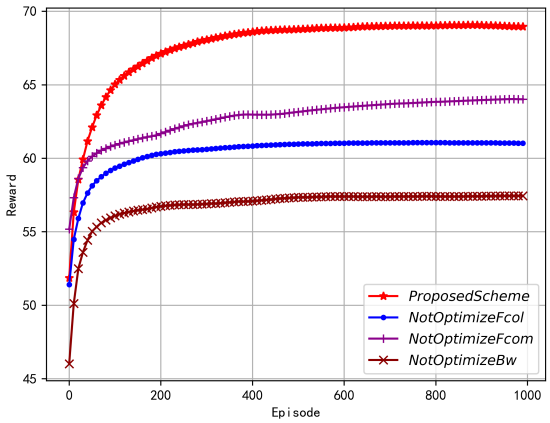

Figure 6. RL reward in different optimization variables.

 doi:10.20944/preprints202501.1482.v1

As shown in Figure 7, with the increase of iteration number, all four optimization schemes can gradually reduce global parameter age and converge. Parameter age is an important indicator of the convergence performance of the agent as an optimization objective in equation(15). Similarly, due to the relatively more optimization variables in the proposed scheme, its corresponding parameter age optimization declines at a slower convergence speed, but it can still reduce global parameter age to a very small value. Fixed collection frequency at a larger value can maintain smaller local data parameter age, but this scheme cannot make adaptive adjustments to the randomness of FL requests, so its parameter age will be slightly smaller than that of the proposed scheme. In addition, this will also bring great energy consumption, which is why its parameter age is smaller than that of fixed computation frequency but reward is very close to it. Fixed device's computation frequency makes the time spent by each device collecting unit data relatively fixed, but FL requests arrive randomly and require comprehensive consideration of the overall system energy consumption. The scope of adaptive adjustment of data collection frequency is limited and can only be controlled to a lower collection frequency, resulting in larger data sample age. Overall, adjusting either data collection frequency or computation frequency alone cannot effectively reduce local sample parameter age, and both need to be dynamically adjusted together to achieve optimal results. Furthermore, fixing the bandwidth size allocated to each device is influenced by the dragger under high dynamic communication conditions, causing the central server to wait for the dragger to complete FL service before aggregating, during which all device's parameter ages are aging together, leading to extremely poor overall parameter age performance.

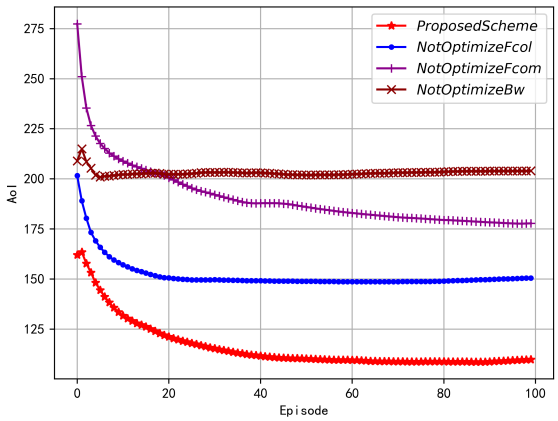

Figure 7. Global AoI in different optimization variables.

Figure 8 compares the boxplot of global parameter age under four schemes, including the proposed scheme, the fixed data collection frequency scheme, the fixed device computation frequency scheme, and the fixed device bandwidth scheme. From the figure, it can be seen that the proposed scheme performs the best in terms of global parameter age minimization, and its overall stability is relatively good with a smaller range for the upper and lower bounds of global parameter age. This proves that the proposed scheme not only performs the best in terms of global parameter age optimization but also has good stability, further validating the correctness and superiority of the scheme.

 doi:10.20944/preprints202501.1482.v1

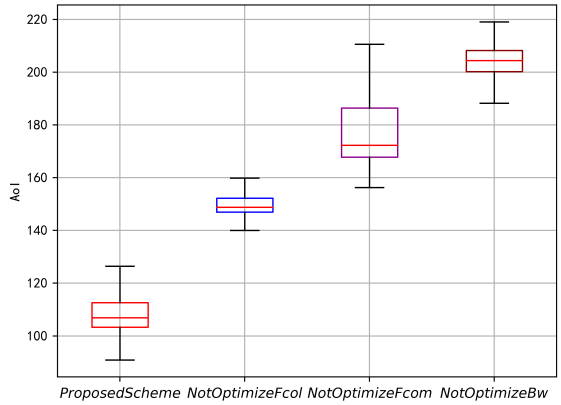

Figure 8. Boxplot global AoI in different optimization variables.

Figures 9 and 10 compare the accuracy and error of FL under four schemes, including the proposed scheme, the only model cache scheme, the only local data cache scheme, and the no-cache scheme. From the figures, it can be seen that these four schemes can gradually increase their FL accuracy and converge as the global iteration number increases. The convergence trend of these schemes is slightly different due to outdated data reducing the global model's convergence speed. As analyzed in [26], due to gradient coherence, if the current gradient is consistent with the gradient direction of the recent period, then the current update is valid; if the current update direction is opposite to the historical direction, then one of the updates (current or historical) is a wrong update, hindering model convergence. The proposed scheme converges the fastest because its global parameter age performs best, which is conducive to global model convergence. The other three comparison schemes converge relatively slowly, but with increasing iterations, these schemes can all converge to roughly the same level.

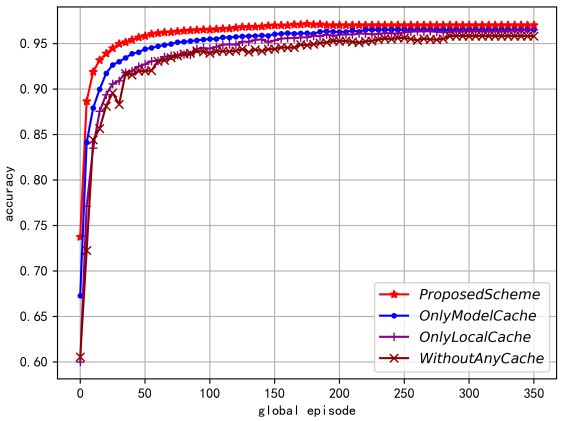

Figure 9. FL accrucy in different optimization variables.

 doi:10.20944/preprints202501.1482.v1

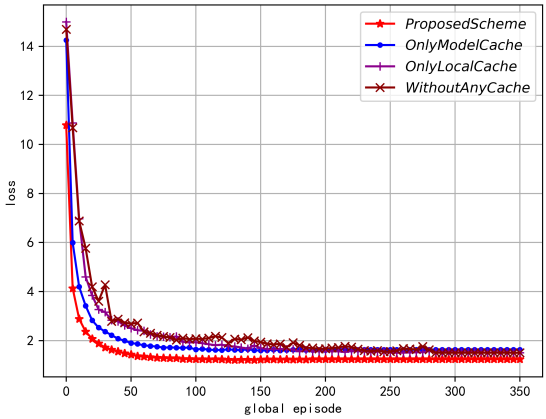

Figure 10. FL loss in different optimization variables.

5. Conclusion

In this paper, we consider the scenario of random FL tasks with high computational demands and service quality requirements and propose a parametric age-aware WCPN FL resource scheduling strategy. Considering various service demands of FL, we design a data caching mechanism to improve the global model performance of FL by optimizing device's data collection frequency, computation frequency, and spectrum resources. We comprehensively consider the parameters' staleness, time-varying channels, random FL requests arrival, and heterogeneous computing capabilities among participating devices, and use PPO algorithm to solve the optimization objective. The numerical results show that our proposed scheme can effectively balance the system benefits in terms of global parameter age, energy consumption, and service latency, resulting in maximum system efficiency.

Funding: This work was supported in part by the NSFC Programs under Grant 62371142 and Grant 62273107; and in part by the GuangDong Basic and Applied Basic Research Foundation under Grant 2024A1515010404.

References

1. Y. Lu and X. Zheng, " 6G: A survey on technologies, scenarios, challenges, and the related issues," *Journal of Industrial Information Integration*, vol. 19, 2020, pp. 100158.
2. W. Sun, S. Lei, L. Wang, Z. Liu, and Y. Zhang, " Adaptive federated learning and digital twin for industrial internet of things," *IEEE Transactions on Industrial Informatics* ., vol. 17, no. 8, pp. 5605-5614, 2020.
3. Y. Qu, C. Dong, J. Zheng, H. Dai, F. Wu, S. Guo, and A. Anpalagan, " Empowering edge intelligence by air-ground integrated federated learning," *IEEE Network*., vol. 35, no. 5, pp. 34-41, 2021.
4. H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, D. Bacon, et al., " Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492* ., 2016.
5. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, " Communication-efficient learning of deep networks from decentralized data," *Available as ArXiv:1602.05629*., 2016.
6. Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, " Deep gradient compression: Reducing the communication bandwidth for distributed training," *Int. Conf. Learn. Represent. (ICLR), Vancouver, Canada* ., May 2018.
7. K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y. J. A. Zhang, " The roadmap to 6G:CAI empowered wireless networks," *IEEE Commun. Mag* ., vol. 57, no. 8, pp. 84-90, Aug. 2019.
8. P. Wang, W. Sun, H. Zhang, W. Ma, and Y. Zhang, " Distributed and secure federated learning for wireless computing power networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 7, 2023, pp. 9381–9393.
9. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, " A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys and Tutorials* ., vol. 19, no. 4, pp. 2322-2358, Aug. 2017.
10. S. Ha, J. Zhang, O. Simeone, and J. Kang, " Coded federated computing in wireless networks with straggling devices and imperfect CSI," *Available as ArXiv:1901.05239*., 2019.
11. A. Reisizadeh, I. Tziotis, H. Hassani, A. Mokhtari and Ramtin Pedarsani, " Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity," *arXiv preprint arXiv:2012.14453*., 2020.

12. M. Chen, H. V. Poor, W. Saad and S. Cui, " Convergence Time Minimization of Federated Learning over Wireless Networks," *ICC 2020 - 2020 IEEE International Conference on Communications (ICC).*, pp. 1-6, Dublin, Ireland, Jul. 2020.
13. X. Huang, S. Leng, S. Maharjan, and Y. Zhang, " Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks," *IEEE Trans. Veh. Technol..*, vol. 70, no. 9, pp. 9282-9293, Sep. 2021.
14. V. Balasubramanian, M. Aloqaily, and M. Reisslein, " FedCo: A federated learning controller for content management in multi-party edge systems," *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN).*, 2021, pp. 1-9.
15. W. Sun, Z. Li, Q. Wang and Y. Zhang, " FedTAR: Task and Resource-Aware Federated Learning for Wireless Computing Power Networks," *IEEE Internet of Things Journal.*, vol. 10, no. 5, pp. 4257-4270, 1 March1, 2023, doi: 10.1109/JIOT.2022.3215805.
16. Y. Liu, Z. Chang, G. Min, and S. Mao, " Average age of information in wireless powered mobile edge computing system," *IEEE Wireless Communications Letters*, vol. 11, no. 8, 2022, pp. 1585–1589.
17. G. Zhang, Y. Zheng, Y. Liu, J. Hu, and K. Yang, " Resource Scheduling for Timely Wireless Powered Crowdsensing with the Aid of Average Age of Information," *ICC 2024 - IEEE International Conference on Communications*, 2024, pp. 4161–4166,doi: 10.1109/ICC51166.2024.10622387.
18. J. Zhu and J. Gong, " Optimizing Peak Age of Information in MEC Systems: Computing Preemption and Non - Preemption," *IEEE/ACM Transactions on Networking*, vol. 32, no. 4, 2024, pp. 3285–3300,doi: 10.1109/TNET.2024.3384706.
19. B. S. Vineeth and R. C. Thomas, " On the Average Age-of-Information for Hybrid Multiple Access Protocols," *IEEE Networking Letters.*, vol. 4, no. 2, pp. 87-91, June 2022, doi: 10.1109/LNET.2022.3159567.
20. M. Moltafet, M. Leinonen and M. Codreanu, " Average Age of Information for a Multi-Source M/M/1 Queueing Model with Packet Management and Self-Preemption in Service," *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT).*, Volos, Greece, 2020, pp. 1-5.
21. Y. Zheng, J. Hu and K. Yang, " Average Age of Information in Wireless Powered Relay Aided Communication Network," *IEEE Internet of Things Journal.*, vol. 9, no. 13, pp. 11311-11323, 1 July1, 2022, doi: 10.1109/JIOT.2021.3128358.
22. E. Tan Zheng Hui and A. S. Madhukumar, " Mean Peak Age of Information Analysis of Energy - Aware Computation Offloading in IIoT Networks," *2024 IEEE 99th Vehicular Technology Conference (VTC2024 - Spring)*, 2024, pp. 1–6,doi: 10.1109/VTC2024-Spring62846.2024.10683346.
23. Y. Zhang, Y. Jiang, X. Zhu, J. Cao, and S. Sun, " Optimized Age of Information for Relay Systems with Resource Allocation," *2024 IEEE 99th Vehicular Technology Conference (VTC2024 - Spring)*, 2024, pp. 1–5,doi: 10.1109/VTC2024-Spring62846.2024.10683008.
24. J. Zhu and J. Gong, " Optimizing Peak Age of Information in Mobile Edge Computing," *2023 35th International Teletraffic Congress (ITC - 35)*, 2023, pp. 1–9,doi: 10.1109/ITC-3560063.2023.10555735.
25. X. Wang, Z. Ning, S. Guo, M. Wen, and V. Poor, " Minimizing the Age of-critical-information: An imitation learning-based scheduling approach under partial observations," *IEEE Trans. Mobile Comput..*, early access, Jan. 21, 2021, doi: 10.1109/TMC.2021.3053136.
26. W. Dai, Y. Zhou, N. Dong, H. Zhang, and E. P. Xing, " Toward understanding the impact of staleness in distributed machine learning," *Int. Conf. Learn. Represent. (ICLR), New Orleans, Louisiana.*, May 2019, pp.1-6.
27. D. Buck and M. Singhal, "An analytic study of caching in computer systems," *Journal of Parallel and Distributed Computing*, vol. 32, no. 2, pp. 205–214, 1996. Elsevier.
28. F. Fu, X. Miao, J. Jiang, H. Xue, and B. Cui, "Towards communication-efficient vertical federated learning training via cache-enabled local updates," *arXiv preprint arXiv:2207.14628.*, 2022.
29. Z. Wu, S. Sun, Y. Wang, M. Liu, K. Xu, W. Wang, X. Jiang, B. Gao, and J. Lu, "Fedcache: A knowledge cache-driven federated learning architecture for personalized edge intelligence," *IEEE Transactions on Mobile Computing.*, 2024. IEEE.
30. Y. Liu, L. Su, C. Joe-Wong, S. Ioannidis, E. Yeh, and M. Siew, "Cache-Enabled Federated Learning Systems," *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing.*, pp. 1–11, 2023.
31. X. Liu, X. Qin, H. Chen, Y. Liu, B. Liu and P. Zhang, " Age-aware Communication Strategy in Federated Learning with Energy Harvesting Devices," *2021 IEEE/CIC International Conference on Communications in China (ICCC), Xiamen, China.*, 2021, pp. 358-363, doi: 10.1109/ICCC52777.2021.9580240.
32. Q. Meng, H. Lu, and L. Qin, " Energy Optimization in Statistical AoI - Aware MEC Systems," *IEEE Communications Letters*, vol. 28, no. 10, 2024, pp. 2263–2267,doi: 10.1109/LCOMM.2024.3450127.

33. Z. Zhu, S. Wan, P. Fan and K. B. Letaief, " Federated Multiagent Actor-Critic Learning for Age Sensitive Mobile-Edge Computing," *IEEE Internet of Things Journal*., vol. 9, no. 2, pp. 1053-1067, 15 Jan.15, 2022, doi: 10.1109/JIOT.2021.3078514.
34. J. Xu, X. Jia and Z. Hao, " Research on Information Freshness of UAV-assisted IoT Networks Based on DDQN," *2022 IEEE 5th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Chongqing, China*., 2022, pp. 427-433, doi: 10.1109/IMCEC55388.2022.10019931.
35. L. Dong, Y. Zhou, L. Liu, Y. Qi, and Y. Zhang, " Age of Information Based Client Selection for Wireless Federated Learning With Diversified Learning Capabilities," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, 2024, pp. 14934–14945, doi: 10.1109/TMC.2024.3450549.
36. X. Xiao, X. Wang, and W. Lin, " Joint AoI-Aware UAVs Trajectory Planning and Data Collection in UAV-Based IoT Systems: A Deep Reinforcement Learning Approach," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 4, 2024, pp. 6484–6495, doi: 10.1109/TCE.2024.3440406.
37. Y. -L. Hsu, C. -F. Liu, H. -Y. Wei and M. Bennis, " Optimized Data Sampling and Energy Consumption in IIoT: A Federated Learning Approach," *IEEE Transactions on Communications*., vol. 70, no. 12, pp. 7915-7931, Dec. 2022, doi: 10.1109/TCOMM.2022.3216353.
38. S. Zhang, J. Li, H. Luo, J. Gao, L. Zhao, and X. S. Shen, " Towards fresh and low-latency content delivery in vehicular networks: An edge caching aspect," *Proc. 10th Int. Conf. Wireless Commun. Signal Process.*., 2018, pp. 1-6.
39. W. Dai, Y. Zhou, N. Dong, H. Zhang, and E. P. Xing, " Toward Understanding the Impact of Staleness in Distributed Machine Learning," *arXiv preprint arXiv:1810.03264*., 2018.
40. Bie, T., Zhu, X.Q., Fu, Y., et al, " Safety priority path planning method based on Safe-PPO algorithm," *Journal of Beijing University of Aeronautics and Astronautics*., 2021, 1-15, DOI:10.13700/j.bh.1001-5965.2021.0580.
41. V. Mnih, A. Puigdomenech Badia, M. Mirza, A. Graves, T. Lillicrap, TimHarley, D. Silver, and K. Kavukcuoglu, " Asynchronous methods for deep reinforcement learning," *In Proceedings of The 33rd International Conference on Machine Learning, Proceedings of Machine Learning Research, New York, USA*., 2016. PMLR.