Computer Science and Information Systems 20(3):1037–1060 https://doi.org/10.2298/CSIS220930026C

A Hierarchical Federated Learning Model with Adaptive Model Parameter Aggregation

Zhuo Chen[1], Chuan Zhou[1] and Yang Zhou[2]

[1] College of Computer Science and Engineering, Chongqing University of Technology
Chongqing, China
chenzhuo@cqut.edu.cn
czhou@2020.cqut.edu.cn

[2] Department of Computer Science and Software Engineering, Auburn University
Auburn, USA
yangzhou@auburn.edu

Abstract. With the proposed Federated Learning (FL) paradigm based on the idea of "data available but invisible", participating nodes which create or hold data can perform local model training in a distributed manner, then a global model can be trained only by continuously aggregating model parameters or intermediate results from different nodes, thereby achieving a balance between data privacy protection and data sharing. However, there are some challenges when deploying a FL model. First, there may be hierarchical associations between participating nodes, so that the datasets held by each node are no longer independent of each other. Secondly, due to the possible abnormal delay of data transmission, it can seriously influence the aggregation of model parameters. In response to the above challenges, this paper proposes a newly designed FL framework for the participating nodes with hierarchical associations. In this framework, we design an adaptive model parameter aggregation algorithm, which can dynamically decide the aggregation strategy according to the state of network connection between nodes in different layers. Additionally, we conduct a theoretical analysis of the convergence of the proposed FL framework based on a non-convex objective function. Finally, the experimental results show that the proposed framework can be well applied to applications in different network connections, and can achieve faster model convergence efficiency while ensuring the accuracy of the model prediction.

Keywords: Parameter Aggregation, Federated Learning, Internet of Things, Privacy Computing.

1. Introduction

Traditional machine learning models usually need to collect data generated in distributed locations into a central storage point (e.g., a cloud data center) for model training. However, with the increase in the number of mobile terminals and Internet of Things (IoT) sensors, the data generated is not only more diverse in data types and formats, but also the scale of data is also proliferating. While larger-scale data can help train better machine learning models, transferring large amounts of data consumes more network resources [1]. In addition, there is a non-negligible risk of information leakage in the process of data collection, transmission and storage [2]. Furthermore, data holders are increasingly

reluctant to transfer data to other uncontrolled locations for the purpose of privacy protection. These issues pose challenges for building machine learning models in an environment where data is increasingly fragmented and isolated. In recent years, service nodes deployed close to users have been greatly improved in terms of computing power, storage resources, and network transmission capabilities, which have laid the foundation for building distributed machine learning models based on these distributed nodes [3, 4]. In particular, Federated Learning (FL) [5] proposed based on the concept of "data does not move while model moves", enables collaborative learning among multiple participating nodes without the data leaving the place they are generated, which is called as FedAvg. One global model is trained only by continuously aggregating model parameter or intermediate results, thereby achieving a balance between data privacy protection and data sharing. This new type of machine learning paradigm has recently received continuous attention from academic community.

A typical FL model can be regarded as a two-layers FL framework[6] composed of a parameter server (PS) with sufficient computing power (e.g., a server deployed in cloud data center) and multiple clients with acceptable computing capability (e.g., edge service nodes) . The operation process of a typical FL model is demonstrated in Fig.1. The client independently performs local model training based on local dataset, and global model is optimized through the exchange of model parameters under the encryption mechanism. Then, the global model is transferred to the clients for facilitating local training next time. The whole process continues until the model converges, or reaches the maximum number of iterations. Since PS obtains model gradients or model parameters rather than raw data from clients, the purpose of protecting the privacy can be achieved [6, 7]. However, in the practical scenarios, there is often a more complex hierarchical relationship between the data holders, and the parameters exchange between clients and PS may be fulfilled by IoT network with unstable transmission quality. When faced with such kind of scenario, the existing work lacks some considerations on two issues: 1) The data generated by multiple clients may have explicit or implicit correlations. This means that the data features generated by nodes at the lower layers of the hierarchical relationship will affect data distribution at higher layers, which no longer makes the dataset on each participating node completely independent. 2) Model quality and model convergence may be affected by the quality of network transmission. Specifically, when the network transmission is abnormal, the model parameters cannot be transferred between PS and the clients in time. If PS uses the synchronous aggregation method [8], the training time will be prolonged. In contrast, if the PS adopts a purely asynchronous aggregation method [9], although the training time can be reduced, the convergence of the model is unsatisfactory.

To address the above issues, a hierarchical federated learning (HFL) framework is proposed in this paper. In this framework, multiple nodes participating in collaborative learning are logically divided into multiple layers. There is an association relationship between the data generated by the nodes at the lower layer and the data generated by the nodes at the upper layer. In addition, the nodes in the middle layer (named Intermediate Aggregation Node, IAN) will not only aggregate the model parameters passed by its lower-layer nodes, but also perform local training based on the data generated by itself. We continually propose an adaptive parameter aggregation strategy. Based on this strategy, the IAN can adaptively adjust the aggregation method according to the quality of IoT-based data transmission to improve the model convergence performance and reduce

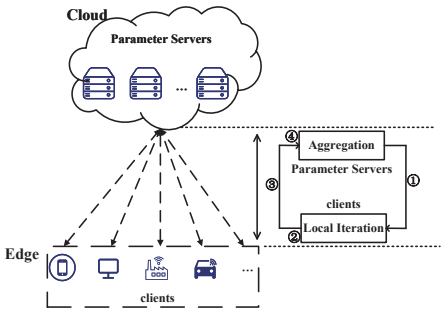

Fig. 1. The operation procedures for a typical FL model

the training time. Furthermore, we conduct a theoretical analysis of the convergence for the proposed HFL model with non-convex objective function, demonstrating that the convergence of the proposed HFL framework depends on aggregation frequencies and the total number of training rounds. Finally, using the HFL model, we achieve water demand forecasting in sub-regional and hierarchical water supply scenarios involving multiple water supply companies. The main contributions of this paper are summarized as follows.

- We propose a HFL model that integrates adaptive parameter aggregation algorithm. Under different network transmission delay, this framework can help multiple nodes that have hierarchical relationships and participate in joint learning to achieve better model training performance.
- We conduct an in-depth analysis of the model convergence and examine key parameters that have important impact on convergence.
- We establish the proposed HFL model to realize the prediction of urban water demand. Furthermore, through comparing with similar work, we finally verify the effectiveness and efficiency of the model.

The rest of the paper is organized as follows. Related work is discussed in Section 2. Section 3 proposes the HFL framework and presents an adaptive aggregation strategy. In Section 4, we conduct an in-depth convergence analysis of the proposed HFL framework. Section 5 describes the case of realizing urban water demand prediction based on the HFL, and the experimental results of the model are presented. Finally, the paper is concluded in section 6.

2. Related Work

In this section, we will discuss the representative HFL frameworks and the existing efforts on performance improvement of HFL.

Some HFL models have recently been proposed for the so-called "Terminal-Edge-Cloud" network service architecture. Reference [10] builds a layered FL model by relying on terminals, edge nodes and cloud servers as participants in joint learning. By taking advantage of the respective advantages of edge nodes and cloud server, the rational use of

computing and communication resources can be realized. Z. Wang et al.[11] treat cloud servers and edge nodes as global aggregator and cluster aggregator, then an asynchronous aggregation method and a synchronous aggregation method are adopted to achieve parameter aggregation, respectively. This work only performs a convergence analysis based on a convex objective function. In addition, W. Lim et al.[12] design a resource allocation mechanism and an incentive mechanism for a hierarchical FL architecture. The above-mentioned HFL frameworks do not consider the correlation between the data generated by multiple nodes participating in joint learning when designing the model. Although the above mentioned works proposed multiple hierarchical structure based FL models to balance the local iterations and runtime, it has not taken into account the potential relationship between the data held by the participating nodes.

Different from the centralization-based machine learning model, the training datasets of different scales held by participating nodes, the non-independent and identically distributed (non-IID) characteristics between different datasets, and the unstable network transmission quality are all potential factors that may affect the performance of FL models. To address these challenges, McMahan et al. [5] proposed the FedAvg algorithm, which enables participants to perform gradient descent independently, and finally the aggregation node averages the staged gradient values of clients to achieve model aggregation. Furthermore, Li Tian et al. [13] proposed an algorithm called FedProx, which can be effectively applied to highly heterogeneous environments and obtained satisfactory convergence. In [14], the synchronous aggregation mechanism is adopted to realize the parameter interaction between PS and clients, but it is difficult to applied to the scenarios with unstable network environment. In addition, for IoT networks with unstable network links, Chen et al. [15] proposed a lightweight node selection strategy based on an asynchronous FL model, which can improve the model training efficiency. H. Zhu et al. [16] considered availability and fairness in the client nodes scheduling process, and designed an asynchronous aggregation algorithm to improve the convergence of the model. C. Chen et al. [17] proposed an adaptive parameters transmission algorithm. The model parameters that are temporarily stable will not participate in the network transmission process, thereby reducing network bandwidth consumption. J. Liu et al. [18] combined Deep Reinforcement Learning (DRL) to propose an adaptive algorithm that adjusts the number of nodes participating in joint learning, and intelligently adjusts the local updates delivered to the PS according to the network state during each round of aggregation. J. Jin et al. [19] applied an adaptive optimization algorithm to FL to accelerate model convergence. These existing works are trying to improve the performance of HFL from the selection of nodes involved in the aggregation, the improvement of model training efficiency, the transmission of training parameters, and the design of the local update mechanism, etc. However, as far as we know, they have not considered adaptive parameters aggregation in the case of complex associations between multiple participating nodes in the collaborative learning.

3. A Hierarchical Federated Learning Framework

In this section, we first introduce the proposed HFL framework in detail, and then propose an adaptive parameter aggregation algorithm for HFL.

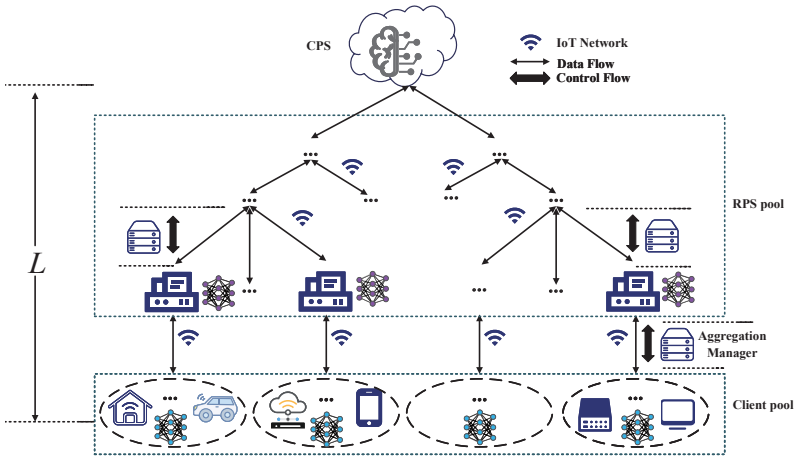

Fig. 2. The architecture of the proposed HFL

3.1. The description of the proposed HFL Model

As shown in Fig.2, we propose a newly designed HFL framework. In this framework, nodes are logically divided into multiple layers, and nodes between two adjacent layers can be interconnected through IoT-based network to realize data transmission. These participating nodes, from the role of performing FL, consist of Central Parameter Server (CPS), Regional Parameter Server (RPS) and clients. Since the RPS is located in the middle layer (or layers) of the proposed framework, it can connect CPS and clients at the same time. Therefore, a RPS actually plays the role of an IAN. Since then, we will use RPS to replace IAN to make further description. Specifically, the CPS is usually one cloud server with powerful computing capability. The CPS trains the global model and can interact with multiple RPSs. One RPS is typically one edge server with IoT connectivity that not only generates or collects data, but also trains regional model. At the same time, the RPS can aggregate the parameter updates of other RPSs or clients. A client is usually acted as an IoT terminal or a light-wight edge service node with acceptable computing capabilities. A client mainly performs local model training and interacts with RPSs with model parameters. These three types of participating nodes cooperate with each other to aggregate the model parameters and complete the FL model training. In addition, the framework also includes an Aggregation Manager (AM), which can periodically check the quality of the current IoT network. AM is the basis for the adaptive aggregation strategy with CPS and RPS. In a practical scenario, AM may be monitoring nodes managed by telecom operators responsible for operating the IoT network.

We assume that the area where HFL will be deployed can be logically divided into S sub-regions, that is, the HFL framework includes S RPSs. There are K clients in each sub-region. For the distinction in description, the parameter aggregation on the RPS, the model obtained by RPS aggregation, the local training on the RPS and the corresponding training model are called regional aggregation, regional model, regional training and regional updated model, respectively. Additionally, the aggregation on the CPS and the model aggregated are called global aggregation and global model, respectively. The local

Table 1. Summary of main notations

Notation	Description
$\boldsymbol{K}, K$	The set of clients associated with a RPS, the number of clients associated with a RPS
$\boldsymbol{S}, S$	The set of all RPSs, the number of RPSs
$\mathcal{D}_k$, D_k	Local dataset of client k, the size of $\mathcal{D}_k$
$\mathcal{D}_s$, D_s	Local dataset of RPS s, the size of $\mathcal{D}_s$
$\mathcal{D}$, D	The whole dataset, the size of $\mathcal{D}$
Γ^r	The size of $\bigcup_{k\in\alpha_c \boldsymbol{K}} D_k$ on round r
B	Global aggregation interval
K_1	The whole number of training rounds for a client ($K_1 = B\kappa_1\kappa_2$)
H	The number of performing iterations per round
K_2	Thw whole number of training rounds for a RPS ($K_2 = B\kappa_3$)
r	Index of local training rounds
t	Index of regional training rounds
α_c	A certain fraction of $\boldsymbol{K}$
α_s	A certain fraction of $\boldsymbol{S}$
f	Global loss function on dataset $\bigcup_{s\in\alpha_s \boldsymbol{S}} D_s$
F_s	Edge loss function on dataset $\mathcal{D}_s$
f_s	Edge loss function on dataset $\bigcup_{k\in\alpha_c \boldsymbol{K}} D_k$
f_k	client loss function on dataset D_k
Q	The true transmission delay to evaluate the network quality
T	The accepted transmission delay

training on the client and the corresponding model are called local training and local updated model, respectively. The global model w_0 with traditional FL is usually initialized in a random manner and broadcast by central server to others. However, in the HFL, the initial global model w_0 is learned from the common features of each node's dataset by CPS, and w_0 is broadcast to RPSs and clients. RPSs and clients start the training of local models based on the local dataset and initial weights. The training of the local model is performed in a parallel and distributed manner. The way of local update is performed as follows

$$
\begin{aligned}
H \text{ iterations } &\begin{cases} w_{r,1}^k = w_0 - \eta \nabla f_1^k\left(w_0\right) \\ w_{r,2}^k = w_{r,1}^k - \eta \nabla f_2^k\left(w_{r,1}^k\right) \\ \cdots \end{cases} \\
&\Rightarrow w_{r,H}^k = w_0 - \eta \sum_{i=1}^{H} \nabla f_i^k\left(w_{r,i}^k\right),
\end{aligned} \tag{1}
$$

where $w_{r,H}^k$ is the local update obtained by node k in the round r after H local iterations, and η is the learning rate. In particular, when $r = H = 0$, $w_{r,H}^k = w_0$. Then, integrating the iterative results of H times, the final equation in Eq.(1) can be obtained. The optimization method adopted in Eq.(1) is SGD, and Adam optimizer can also be applied [20]. After the local updates from clients are obtained by RPSs , FedAvg is used to obtain the regional model w_r^s, and the aggregation method can be represented as follows

$$w_r^s = \sum_{k=1}^{\alpha_c K} \frac{D_k}{\Gamma_s^r} w_{r,H}^k, \tag{2}$$

where $\alpha_c (\alpha_c \in \left\{\frac{1}{K}, \frac{2}{K}, \ldots, 1\right\})$ represents the proportion of clients selected to participate in the aggregation, the dataset on client k is represented by $\mathcal{D}_k$, the size of $\mathcal{D}_k$ is D_k ($D_k \triangleq |\mathcal{D}_k|$, where $|\cdot|$ denotes the cardinality). The size of the whole dataset in sub-region s with $\alpha_c K$ clients is $\Gamma_s^r = \left|\bigcup_{k \in \alpha_c K} \mathcal{D}_k\right|$.

After clients and RPS have completed $\kappa_1 \kappa_2$ rounds of local training and κ_2 times of regional aggregations respectively, then RPS performs iterative training based on its own dataset and generates a regional update w_s. The iterative process is the same as Eq.(1). Unlike clients, the number of iteration in one RPS is needed to execute κ_3 rounds to end. After each round of iterative execution, the RPS uploads w_s to the upper-layer RPS or CPS for aggregating a wider range of regional model or global model. In the proposed HFL model, the evolution of local weight $w_{r,i}^k$ of client k can be represented as follow

$$w_{r,i}^k = \begin{cases} w_{r,i-1}^k - \eta \nabla f_i^k \left(w_{r,i-1}^k\right), \text{if } i \geq 1 \text{ , } r \mid \kappa_1 \neq 0 \\ \frac{\sum_{k \in \alpha_c K} D_k \left(w_{r,H-1}^k - \eta \nabla f_H^k \left(w_{r,H-1}^k\right)\right)}{\Gamma_s}, \text{if } r \mid \kappa_1 = 0 \\ \frac{\sum_{s \in \alpha_s S} D_s \left(w_{t,H-1}^s - \eta \nabla f_H^s \left(w_{t,H-1}^s\right)\right)}{D}, \text{if } r \mid \kappa_1 \kappa_2 = r \mid \kappa_3 = 0 \end{cases} \tag{3}$$

The update process of the weight w_s of the RPS in the sub-region s is similar to Eq.(3). According to the description of Fig.2, the parameter aggregation and training process of the proposed HFL is shown in Algorithm 1. Especially, the loss function at CPS is

$$\min_{w \in \mathbb{R}} f(w) = \sum_{s=1}^{\alpha_s S} \frac{D_s}{D^t} F_s(w), \tag{4}$$

where $F_s(w) = 1/D_s \cdot \sum_p^{D_s} \mathcal{F}_s(w_s, \zeta_p)$, $D_s \triangleq |\mathcal{D}_s|$ and $D^t = \left|\bigcup_{s \in \alpha_s S} \mathcal{D}_s\right|$. Especially, $\mathcal{F}_s(w_s, \zeta_p)$ is the loss function of the p-th data sample. Furthermore, as shown in Eq.(5), there is a weight w_r^s that minimizes the regional loss function.

$$w_r^s = \arg\min f_s = \arg\min \sum_{k=1}^{\alpha_c K} \frac{D_k}{\Gamma_s^r} f_k. \tag{5}$$

3.2. An adaptive parameter aggregation method for HFL

In this part, an adaptive parameter aggregation algorithm is proposed, which dynamically integrates synchronous and asynchronous aggregation method into the proposed HFL framework, so as to enable different nodes in the HFL framework (i.e., between CPS and RPSs, between RPS and RPS, and between RPS and clients) can perform adaptive parameter aggregation according to current connection state of the wireless IoT network. The monitoring of the connection status is performed by the AM. The AM informs the corresponding nodes of the connection status information of the IoT network to dynamically adjust the aggregation strategy adopted between the corresponding nodes (i.e.,

Algorithm 1 The parameter aggregation and training process of HFL

Input: the initial global model w_0, the number of clients that belong to one RPS:K, the number of RPS:S, the learning rate :η, other parameters are related to training rounds:B,κ_1,κ_2.
Output: the final global model w.

for each round $r = 1, 2, 3, ..., B\kappa_1\kappa_2$ **do**
  for each client $k = 1, 2, ..., K$ in parallel **do**
    $w_r^k = w_{r-1}^k - \eta\nabla f_k$ /* Processing at the clients*/.
  end for
  if $r|\kappa_1 = 0$ **then**
    Send w_r^k back to related RPS by client k.
    for each RPS $s = 1, 2, ..., S$ in parallel **do**
      Aggregate the local models by order in which clients arrive according to Eq.(2) for getting w_r^s /*Processing at the RPS*/.
      Send w_r^s to related clients again.
    end for
  end if
  if $r|\kappa_1\kappa_2 = 0$ **then**
    for each RPS $s = 1, 2, ..., S$ in parallel **do**
      Aggregate the local models by order in which clients arrive to get w_s^0.
      for $j = 1, 2, ..., \kappa$ **do**
        $w_s^j \leftarrow w_s^{j-1} - \eta\nabla f_s$.
      end for
      Send w_s^j back to CPS by RPS.
    end for
    Aggregate the regional models by order in which RPS arrive to get w /*Processing at the CPS*/.
    Send w to all edge devices(RPS,client) as the new w_0 for the next round.
  end if
end for

between CPS and RPS, between RPS and RPS, and between RPS and clients). We use Fig.3 to represent the complete training process performed by different types of nodes in HFL. First, a specific client will complete κ_1 rounds of local training, and then send the local model parameters to the corresponding RPS, so that the RPS can complete the aggregation of the regional model. When the above process is performed κ_2 times, the RPS will use the aggregated regional model parameters and its own private data as input, and then continue to complete κ_3 rounds of regional training. The obtained regional model parameters are then uploaded to the CPS through the IoT network to complete the training of the global model. Until the end of the global model training, the total number of iterations of the process is B, in which the total number of local training rounds in each client is $K_1(K_1 = B\kappa_1\kappa_2)$, and the total number of regional training rounds in each RPS is K_2 ($K_2 = B\kappa_3$). Furthermore, the threshold T is defined to represent the minimum acceptable transmission delay during the model training. Before RPS and CPS perform parameter aggregation, they will obtain the information of transmission delay between nodes located in two adjacent layers periodically detected by AM. The delay is represented by the parameter Q. When $Q > T$, it means that the current communication quality is unsatisfactory. Therefore, an asynchronous aggregation mechanism is adopted

to reduce the overall training time of the model. On the contrary, a synchronous aggregation strategy is adopted to ensure the convergence stability of the global model.

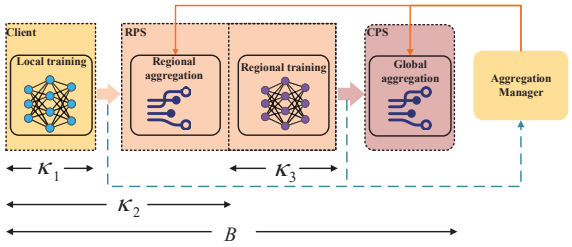

Fig. 3. The training process of the proposed HFL Framework

If we take a three-layer HFL model as an example, the adaptive synchronous and asynchronous aggregation decision will produce four aggregation combinations, i.e., "synchronous-synchronous", "asynchronous-asynchronous", "synchronous-asynchronous", "asynchronous-synchronous". For the synchronous aggregation mechanism, when $\alpha_c = 1$ or $\alpha_s = 1$, it means that it is necessary to wait for all nodes at the corresponding level to complete training and upload to the CPS (or the RPS) before triggering parameter aggregation. In contrast, if the asynchronous aggregation strategy is adopted, when the number of nodes that complete model training and upload the model reaches the specified threshold, the server (CPS or RPS) can be triggered to perform parameter aggregation, but the server only broadcasts the model aggregated to the nodes contributing to current model aggregation for next training. It is worth noting that asynchronous aggregation strategy must take into account: the server receives the local model parameters of node in the r_c-th round, while the node receives the model aggregated from the server in the r_s-th round, and they are often inconsistent [21], i.e., $\lambda = r_s - r_c \neq 0$. Therefore, this paper defines a parameter v_r^k to measure the staleness of node k in the r-th round, where $v_r^k = \rho^\lambda$, $(0 < \rho < 1)$ and ρ is a constant. In particular, if there is no staleness for model update, that is, it is equivalent to a synchronous aggregation mechanism at this time. Therefore, for the asynchronous aggregation strategy, each model parameter $w_{r,H}^k$ owned by node k received by the server will be processed according to Eq.(6) to reduce the impact of nodes with poor staleness on the aggregation model, and then participate in the aggregation process.

$$w_{k,H}^r = v_r^k w_{k,H}^r + \left(1 - v_r^k\right) \overline{w}_s^{r^*}, 0 < v_r^k < 1, \tag{6}$$

where $\overline{w}_s^{r^*} = \frac{1}{\Gamma^{r^*}} \sum_k^{\alpha_c K} D_k \cdot w_{k,H}^{r^*}$ is the regional model after the r^*-th round for aggregation. The modification for $w_{k,H}^r$ will be completed based on $\overline{w}_s^{r^*}$ to participate in the model aggregation at $(r^* + 1)$-th round. If the transmission delay of IoT connection is serious, v_r^k will decrease significantly with the increase of λ, resulting in Eq.(6), the weight $w_{k,H}^r$ of the model participating in the aggregation tends to be close to the aggregation result $\overline{w}_s^{r^*}$ of the previous round, so as to maintain the stability of the global model. The process between CPS and RPS is similar to the above process. Theoretically, the proposed adaptive parameter aggregation algorithm can be extended to one HFL framework with L-layer $(L > 3)$ correlation, and correspondingly 2^L kinds of synchronous

and asynchronous parameter aggregation schemes can be formed. The proposed adaptive aggregation method is shown in Algorithm 2.

Algorithm 2 The adaptive parameter aggregation method

Input: parameter received time t_r, parameter sent time t_s
Output: one aggregation strategy, the number of client for aggregation $\alpha_c K$ in RPS, the number of RPS for aggregation $\alpha_s S$ in CPS, the model parameters corrected $w_{k,H}^r$.

```
The AM calculates the actual transmission delay Q according to t_r and t_s periodically.
if Q is larger than T then
    Telling RPS or CPS to use asynchronous solution.
    α_c K = K − 1 in RPS or α_s S = S − 1 in CPS.
    The RPS or the CPS calculates the update delay λ and records it.
    The RPS corrects related model parameters by Eq.(6) to reduce the impact of staleness, so do CPS.
else
    Telling RPS or CPS to use synchronous solution.
    α_c K = K in RPS or α_s S = S in CPS.
end if
```

4. The Analysis of Convergence

For ease of convergence analysis, we denote the number of local training rounds and regional training rounds as $r(1 \leq r \leq B\kappa_1\kappa_2)$ and $t(1 \leq t \leq B\kappa_3)$, respectively. We assume that an unbiased estimate of $\nabla f_k(w)$ is $g_j(w, \zeta_k^r)$, i.e., $\nabla f_k(w) = \mathbb{E}_{\zeta_k^r \sim \mathcal{D}_k} g_j\left(w; \zeta_k^r\right)$. Also, we assume the loss function is non-convex and smooth. Then we introduce the following assumptions.

Assumption 1:(Lipschitz Gradient). The function f_k, f_s, F_s, f are L-smooth, i.e., $\|\nabla f_k(w) - \nabla f_k\left(w'\right)\| \leq L\|w - w'\|$, $\|\nabla f_s(w) - \nabla f_s\left(w'\right)\| \leq L\|w - w'\|$, $\|\nabla F_s(w) - \nabla F_s\left(w'\right)\| \leq L\|w - w'\|$, $\|\nabla f(w) - \nabla f\left(w'\right)\| \leq L\|w - w'\|$.

Assumption 2:(Bounded Variance). The divergences satisfy: $\|\nabla F_s(w) - \nabla f(w)\|^2 \leq \epsilon_s^2$, $\|\nabla f_k(w) - \nabla f_s(w)\|^2 \leq \epsilon_k^2$, $\|g_k(w, j) - \nabla f_k(w)\|^2 \leq \sigma^2, \forall s, k, j, w$.

The above assumptions are widely used in non-convex optimization theory [20]. Particularly, the parameter ϵ_s^2 and ϵ_k^2 can quantify the similarity of objective functions. Note $\epsilon_s^2 = 0$ or $\epsilon_k^2 = 0$ corresponds to the IID setting.

Theorem 1: Given the learning rate $\eta \leq \frac{1}{L}$, $1 - 3\eta^2 L^2 \geq 0$ and the optimal global model and regional model are respectively $\overline{w}^*$, $\overline{w}_s^*$. When the synchronous aggregation method is adopted, the upper bound of the average regional gradient deviation is given as

follows

$$\frac{1}{B\kappa_1\kappa_2}\sum_{r=1}^{B\kappa_1\kappa_2}\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\right\|^2 \leq \frac{2}{B\kappa_1\kappa_2\eta}\left(\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^0\right)-f_s\left(\overline{\mathbf{w}}_s^*\right)\right)+\eta\sigma^2LK^2$$
$$+\frac{KL^2\eta^2}{B\kappa_1\kappa_2\left(1-3\eta^2L^2\right)\left\|\Gamma^r\right\|^2}\sum_{k\in V_K}\left\|D_k\right\|^2(\Phi_1+\Phi_2) \tag{7}$$
$$\text{where } \Phi_1=2^{B\kappa_1\kappa_2+4}\sigma^2\left(1+K\sum_{k\in V_K}\left\|D_k\right\|^2\right), \Phi_2=3*2^{B\kappa_1K_2+3}\epsilon_k^2$$

Proof. Due to the proposition of Lipschitz smooth, the expectation of f_s can be expressed as

$$\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^r\right)=\mathbb{E}f_s\left[\overline{\mathbf{w}}_s^{r-1}-\eta\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\right]=\mathbb{E}f_s\left[\overline{\mathbf{w}}_s^{r-1}-\eta\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right]$$
$$\leq \mathbb{E}f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\underbrace{-\eta\mathbb{E}\left\langle\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right),\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\rangle}_{A_1}+\frac{\eta^2L}{2}\underbrace{\mathbb{E}\left\|\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla g_k\left(w_k^{r-1}\right)\right\|^2}_{A_2} \tag{8}$$

We further express the bound of A_1 as follows:

$$-\eta\mathbb{E}\left\langle\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right),\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\rangle=\frac{\eta}{2}\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2$$
$$-\frac{\eta}{2}\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\right\|^2-\frac{\eta}{2}\mathbb{E}\left\|\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2 \tag{9}$$

Then the bound of A_2 can be represented as

$$\mathbb{E}\left\|\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kg_k\left(w_k^{r-1}\right)\right\|^2=\mathbb{E}\left\|\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2$$
$$+\mathbb{E}\left\|\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\left(g_k\left(w_k^{r-1}\right)-\nabla f_k\left(w_k^{r-1}\right)\right)\right\|^2\leq\frac{K\sigma^2}{\left\|\Gamma^r\right\|^2}\sum_{k\in V_K}D_k^2+\mathbb{E}\left\|\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2 \tag{10}$$

By replacing A_1 and A_2 in (8) with (9) and (10) respectively, then we can get

$$\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^r\right)\leq\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)+\frac{\eta^2L}{2}\left(\frac{K\sigma^2}{\left\|\Gamma^r\right\|^2}\sum_{k\in V_K}D_k^2\right)-\left(\frac{\eta}{2}-\frac{\eta^2L}{2}\right)\mathbb{E}\left\|\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2$$
$$+\frac{\eta}{2}\left(\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2-\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\right\|^2\right)$$
$$\overset{(a)}{\leq}\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)+\frac{\eta^2L}{2}\left(\frac{K\sigma^2}{\left\|\Gamma^r\right\|^2}\sum_{k\in V_K}D_k^2\right) \tag{11}$$
$$+\frac{\eta}{2}\left(\underbrace{\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2}_{A_3}-\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\right\|^2\right)$$

Since $\eta \leq \frac{1}{L}$ is assumed, then (a) is obtained. Additionally, the following inequality can be derived based on Assumption 1.

$$\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2\leq\frac{KL^2}{\|\Gamma^r\|^2}\sum_{k\in V_K}D_k^2\underbrace{\mathbb{E}\left\|\left(\overline{\mathbf{w}}_s^{r-1}-w_k^{r-1}\right)\right\|^2}_{A_4} \tag{12}$$

According to SGD and FedAvg, the bound of A_4 can be further represented as

$$\begin{aligned}
&\mathbb{E}\left\|\overline{w}_s^{r-1}-w_k^{r-1}\right\|^2=\mathbb{E}\|\overline{w}_s^{r-2}-w_k^{r-2}+\eta g_k\left(w_k^{r-2}\right)-\eta\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kg_k\left(w_k^{r-2}\right)\|^2\\
&=\mathbb{E}\left\|\left(\overline{w}_s^{r-2}-w_k^{r-2}\right)+\eta g_k\left(w_k^{r-2}\right)-\eta\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kg_k\left(w_k^{r-2}\right)\right\|^2\\
&\overset{(b)}{\leq}2\mathbb{E}\left\|\eta g_k\left(w_k^{r-2}\right)-\eta\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kg_k\left(w_k^{r-2}\right)\right\|^2+2\mathbb{E}\left\|\overline{w}_s^{r-2}-w_k^{r-2}\right\|^2
\end{aligned} \tag{13}$$

The above inequality (b) can be obtained from the mean value inequality. After expanding (13), we obtain

$$\begin{aligned}
&\mathbb{E}\left\|\overline{w}_s^{r-1}-w_k^{r-1}\right\|^2\leq 2\left[\mathbb{E}\left\|\eta g_k\left(w_k^{r-2}\right)-\eta\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kg_k\left(w_k^{r-2}\right)\right\|^2+\mathbb{E}\left\|\overline{w}_s^{r-2}-w_k^{r-2}\right\|^2\right]\\
&\leq\eta^2\underbrace{\sum_{i=1}^{r}2^i\mathbb{E}\left\|g_k\left(w_k^{r-i-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kg_k\left(w_k^{r-i-1}\right)\right\|^2}_{A_5}
\end{aligned} \tag{14}$$

We continue to drive the bound of A_5 as follows

$$\begin{aligned}
&\sum_{i=1}^{r}2^i\mathbb{E}\left\|g_k\left(w_k^{r-i-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kg_k\left(w_k^{r-i-1}\right)\right\|^2\\
&\leq\sum_{i=1}^{r}2^{i+1}\mathbb{E}\left\|g_k\left(w_k^{r-i-1}\right)-\nabla f_k\left(w_k^{r-i-1}\right)+\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kf_k\left(w_k^{r-i-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kg_k\left(w_k^{r-i-1}\right)\right\|^2\\
&+\sum_{i=1}^{r}2^{i+1}\mathbb{E}\left\|\nabla f_k\left(w_k^{r-i-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kf_k\left(w_k^{r-i-1}\right)\right\|^2
\end{aligned} \tag{15}$$

We now further bound the term A_5 by mean inequality, we get

$$\begin{aligned}
A_5&\leq\sum_{i=1}^{r}2^{i+2}\mathbb{E}\left\|g_k\left(w_k^{r-i-1}\right)-\nabla f_k\left(w_k^{r-i-1}\right)\right\|^2+\sum_{i=1}^{r}2^{i+1}\mathbb{E}\left\|\nabla f_k\left(w_k^{r-i-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kf_k\left(w_k^{r-i-1}\right)\right\|^2\\
&+\sum_{i=1}^{r}\frac{2^{i+2}K}{\|\Gamma^r\|^2}\sum_{k\in V_K}\|D_k\|^2\mathbb{E}\left\|\left(g_k\left(w_k^{r-i-1}\right)-f_k\left(w_k^{r-i-1}\right)\right)\right\|^2\\
&\leq\sigma^2\left(\sum_{i=1}^{r}2^{i+2}+\sum_{i=1}^{r}2^{i+2}K\sum_{k\in V_K}\|D_k\|^2\right)+\underbrace{\sum_{i=1}^{r}2^{i+1}\mathbb{E}\left\|f_k\left(w_k^{r-i-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K}D_kf_k\left(w_k^{r-i-1}\right)\right\|^2}_{A_6}
\end{aligned} \tag{16}$$

Similarly, the upper bound of A_6 can be derived as follows

$$\sum_{i=1}^{r} 2^{i+1}\mathbb{E}\left\|\nabla f_k\left(w_k^{r-i-1}\right)-\frac{1}{\Gamma^r}\sum_{k\in V_K} D_k f_k\left(w_k^{r-i-1}\right)\right\|^2 \leq 3\sum_{i=1}^{r} 2^{i+1}\left[\mathbb{E}\left\|\nabla f_k\left(w_k^{r-i-1}\right)-\nabla f_k\left(\overline{w}_s^{r-i-1}\right)\right\|^2+\epsilon_k^2\right] \tag{17}$$

With the results of (14), (16) and (17), we can obtain (18),

$$\mathbb{E}\left\|\overline{w}_s^{r-1}-w_k^{r-1}\right\|^2 \leq 2^{r+3}\eta^2\sigma^2\left(1+K\sum_{k\in V_K}\|D_k\|^2\right)+3\eta^2L^2\sum_{i=1}^{r}2^{i+1}\mathbb{E}\left(\left\|w_k^{r-i-1}-\overline{w}_s^{r-i-1}\right\|^2\right)+3\eta^2\epsilon_k^2 2^{r+2} \tag{18}$$

By averaging the results of $B\kappa_1\kappa_2$ trainings, we can get

$$\begin{aligned}&\frac{1}{B\kappa_1\kappa_2}\sum_{r=1}^{B\kappa_1\kappa_2}\mathbb{E}\left\|\overline{w}_s^{r-1}-w_k^{r-1}\right\|^2 \leq \frac{2^{B\kappa_1\kappa_2+4}}{B\kappa_1\kappa_2}\eta^2\sigma^2\left(1+K\sum_{k\in V_K}\|D_k\|^2\right)+\frac{3^*2^{B\kappa_1\kappa_2+3}\eta^2\epsilon_k^2}{B\kappa_1\kappa_2}\\&+\frac{3\eta^2L^2}{B\kappa_1\kappa_2}\sum_{r=1}^{B\kappa_1\kappa_2}\sum_{i=1}^{r}2^{i+1}\mathbb{E}\left(\left\|w_k^{r-i-1}-\overline{w}_s^{r-i-1}\right\|^2\right)\leq \Phi_1+\Phi_2, 1>1-3\eta^2L^2\geq 0\end{aligned} \tag{19}$$

Through (11), (12), (18) and (19), we can obtain **Theorem 1**, which completes the proof.

$$\begin{aligned}&\frac{1}{B\kappa_1\kappa_2}\sum_{r=1}^{B\kappa_1\kappa_2}\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\right\|^2 \leq \eta\sigma^2LK^2+\frac{2}{B\kappa_1\kappa_2\eta}\left(\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^0\right)-f_s\left(\overline{\mathbf{w}}_s^*\right)\right)\\&+\frac{KL^2\eta^2}{B\kappa_1\kappa_2\left(1-3\eta^2L^2\right)\left\|\Gamma^r\right\|^2}\sum_{k\in V_K}\|D_k\|^2\left(\Phi_1+\Phi_2\right)\end{aligned} \tag{20}$$

where $\eta\leq\frac{1}{L}$, $1-3\eta^2L^2\geq 0$.

Following **Theorem 1**, the similar result can be obtained for the upper bound of the average global gradient deviation and the upper bound of the average regional gradient deviation when synchronous FL is used.

Furthermore, if asynchronous aggregation method is adopted, **Theorem 2** can be obtained under the premise of the above assumptions.

Theorem 2: Given the learning rate $\eta\leq\frac{1}{L}$, $1-6\eta^2L^2\geq 0$. The upper bound of the average regional gradient deviation is given as follows,

$$\begin{aligned}&\frac{1}{B\kappa_1\kappa_2}\sum_{r=1}^{B\kappa_1\kappa_2}\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\right\|^2 \leq \frac{2}{\eta}\left(\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^0\right)-\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^r\right)\right)+\eta L\sigma^2\alpha_c^2K^2\\&+\alpha_cKL^2\sum_{k\in M_{K,r}}\frac{2\eta^2\sigma^2\left(1+\alpha_c^2K^2\right)+6\eta^2L^2\epsilon_k^2}{B\kappa_1\kappa_2\left(1-6\eta^2L^2\right)\left(1-2v_k^{\lambda_k}\right)}\left(2B\kappa_1\kappa_2v_k^{\lambda_k}-\frac{\left(2v_k^{\lambda_k}\right)^2\left(1-\left(2v_k^{\lambda_k}\right)^{B\kappa_1\kappa_2+1}\right)}{1-2v_k^{\lambda_k}}\right)\end{aligned} \tag{21}$$

Proof. Since some proof procedure can be found in **Theorem 1**, we only show the differences from **Theorem 1**: Firstly, we assume that in the r^th round, the server receives the model from the set $M_{K,r}$ which is denoted as the clients sending local model to the

server. We have

$$\begin{aligned}\mathbb{E}f_s\left(\overline{\mathbf{w}}_s^r\right) \leq \mathbb{E}_s\left(\overline{\mathbf{w}}_s^{r-1}\right) + \frac{\eta^2 L}{2}\left(\alpha_c K\sigma^2 \sum_{k\in M_K,}\left\|\frac{D_k}{\Gamma^r}\right\|^2\right) \\ + \frac{\eta}{2}\left(\underbrace{\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right) - \sum_{k\in M_{K,r}}\frac{1}{\Gamma^r}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2}_{A_7} - \mathbb{E}\left\|f_s\left(\overline{\mathbf{w}}_s^{r-1}\right)\right\|^2\right)\end{aligned} \tag{22}$$

Through the use of smoothness, the upper bound of A_7 can be expressed as follows

$$\begin{aligned}\mathbb{E}\left\|\nabla f_s\left(\overline{\mathbf{w}}_s^{r-1}\right) - \sum_{k\in M_{K,r}}\frac{1}{\Gamma^r}D_k\nabla f_k\left(w_k^{r-1}\right)\right\|^2 \leq \\ \alpha_c K L^2 \sum_{k\in M_{K,r}}\left\|\frac{D_k}{\Gamma^r}\right\|^2 \underbrace{\mathbb{E}\left\|\left(\overline{\mathbf{w}}_s^{r-1} - w_k^{r-1}\right)\right\|^2}_{A_8}\end{aligned} \tag{23}$$

Similar to the method adopted in (14) and (16), we can further obtain

$$\begin{aligned}\mathbb{E}\left\|\overline{w}_s^{r-1} - w_k^{r-1}\right\|^2 \leq 2\left\|v_k^{\lambda_k}\right\|^2\left(\mathbb{E}\left\|\overline{w}_s^{r-2} - w_k^{r-2}\right\|^2 + \mathbb{E}\left\|\eta g_k\left(w_k^{r-2}\right) - \eta\sum_{k\in M_{K,r}}\frac{1}{\Gamma^r}D_k g_k\left(w_k^{r-2}\right)\right\|^2\right) \\ \leq \eta^2\underbrace{\sum_{i=1}^{r}2^i\left\|v_k^{\lambda_k}\right\|^{2i}\mathbb{E}\left\|g_k(w_k^{r-i-1}) - \left\|\frac{1}{\Gamma^r}\right\|^2\sum_{k\in M_{K,r}}D_k g_k\left(w_k^{r-2}\right)\right\|^2}_{A_9}\end{aligned} \tag{24}$$

Similarly, we can get

$$\begin{aligned}A_9 \leq \sigma^2\left(\sum_{i=1}^{r}2^{i+1}\left\|v_k^{\lambda_k}\right\|^{2i} + \sum_{i=1}^{r}2^{i+1}\left\|v_k^{\lambda_k}\right\|^{2i}\alpha_c K\sum_{k\in M_{K,r}}\left\|\frac{D_k}{\Gamma^r}\right\|^2\right) \\ + \underbrace{\sum_{i=1}^{r}2^{i+1}\left\|v_k^{\lambda_k}\right\|^{2i}\mathbb{E}\left\|\nabla f_k(w_k^{r-i-1}) - \frac{1}{\Gamma^r}\sum_{k\in M_{K,r}}D_k f_k\left(w_k^{r-i-1}\right)\right\|^2}_{A_{10}}\end{aligned} \tag{25}$$

Finally, based on (22)-(25), the conclusion of **Theorem 2** can be obtained.

5. Performance Evaluation

In this section, we experimentally evaluate the effectiveness and performance of the proposed HFL framework. We firstly describe experimental settings and evaluation metric. Then, we conduct two experiments to illustrate the influence of key parameters on model convergence and demonstrate the efficiency of the proposed model, respectively.

Table 2. The statistics of datasets in the experiment

Area	Nodes	Min_O(divergence)	Max_I(divergence)	Min_I(divergence)
RPS 1	6	0.12412	0.02429	0.01558
RPS 2	4	0.26799	0.07153	0.01426
RPS 3	7	0.16471	0.07686	0.03633
RPS 4	4	0.20628	0.09204	0.03170
RPS 5	5	0.22823	0.09762	0.02344

5.1. Experimental Setup

The HFL framework proposed in this paper is formulated on the basis of analyzing the hierarchical relationship between water supply regions and the historical water supply datasets in Chongqing, China. The datasets used for experiments include the data of daily water supply in different water supply regions from July 2019 to July 2021. We firstly performed noise reduction and smoothing on the data, which is to reduce the impact of noisy data on subsequent model training to a certain extent. Also,the datasets used for training not only includes water supply data at different time points in different water supply areas, but also includes holiday information and environmental data in each water supply area, such as: air temperature, air humidity, rainfall, wind speed and direction, which are taken into account. At the same time, due to the fact that the IoT-based data collection terminals in various regions may lose packets or be interfered by communication during the actual network transmission process, there is a small amount of data missing in the 2-year datasets. According to the relationship between the water supply regions and the HFL structure shown in Fig.2, the entire water supply area includes one CPS for training and aggregating global model, and then the whole region is divided into 5 sub-regions, each sub-region includes one RPS and multiple clients, RPS is responsible for model aggregation in the sub-region, and client is responsible for its local model training. It is worth noting that, we refer to the method adopted in [22, 23] to further divide the original data set into multiple sub-datsets, and increase the scale of participating nodes while ensuring that the original distribution of the data is not changed. The deployment of CPS, RPSs and clients is shown in Fig.4. Furthermore, we use the divergence to measure the similarity between different regions. If the larger the divergence, the smaller the similarity. The data distribution for each region is combined in pairs, and then the corresponding divergence values are then calculated. The statistics datasets and the results on divergence between them are summarized in Table 2. Specifically, *Nodes* represents the number of clients included in the area covered by an RPS. In addition, *Min_O* represents the minimum divergence between the RPS and the clients outside the RPS area. *Max_I* represents the maximum divergence between the clients and RPS in the same RPS area, and *Min_I* indicates the minimum divergence in a specific RPS area. It can be seen from Table 2 that the data similarity in the same region is high, while the data similarity is low outside the area. Finally, according to the ratio of 4:1, each dataset is divided into training dataset and test dataset. In addition, in order to intuitively measure the accuracy of the model, we introduce a commonly used indicator explained variance score(EVC). When the indicator is close to 1, it means that the fitting effect of the model will be better, other-

wise the opposite. The experiments are deployed on a deep learning workstation equipped with NVIDIA GeForce RTX 3090 GPU, and the HFL model proposed is built based on TensorFlow [24].

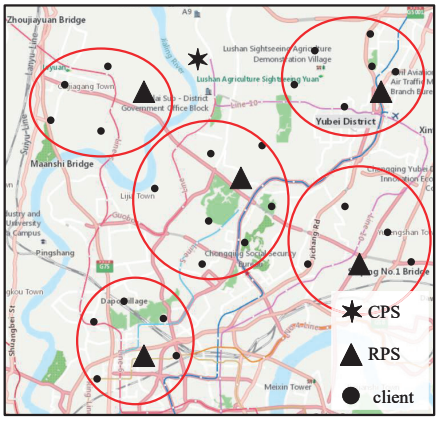

Fig. 4. Deployment of CPS, RPSs and clients in an entire water supply area

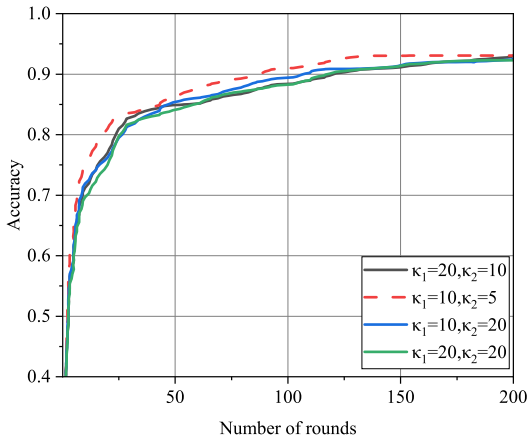

Fig. 5. Convergence under different κ_1 and κ_2

5.2. The Analysis of Convergence

During the training process of the client and the RPS with K_1 and K_2 rounds respectively, the convergence of the global model can be compared ($K_1 = B\kappa_1\kappa_2$, $K_2 = B\kappa_3$) by adjusting the value of κ_1 and κ_2. When the current number of training rounds r satisfies $r|\kappa_1\kappa_2 = 0$, RPS will aggregate local models belonged to the clients in the covered sub-region and obtain the regional model $\overline{w}_s^r$, and further train the regional model of the RPS. The initial model parameter of the RPS is $\overline{w}_s^r$, that is, $w_0^s = \overline{w}_s^r$. Finally, the regional

model of the RPS participates in the global model aggregation at the CPS. It can be seen that the convergence of the global model is directly affected by the RPS from Algorithm 1, while the client indirectly affects the convergence of the global model by affecting the initial model weight of the RPS. Additionally, we assume the minimum acceptable delay $T \to +\infty$ at this time. It can be also seen from **Theorem1** and **Theorem2** that the smaller the values of κ_1 and κ_2, the better the convergence of the model, which is not only applicable to the regional model but also the global model. As shown in Fig.5, the global model convergence is evaluated under different combinations of κ_1 and κ_2. It can be seen that the global model converges best when $\kappa_1 = 10$ and $\kappa_2 = 5$, mainly because the values of these two parameters are the smallest. In contrast, when $\kappa_1 = 20$ and $\kappa_2 = 20$, the model convergence performance is the worst.

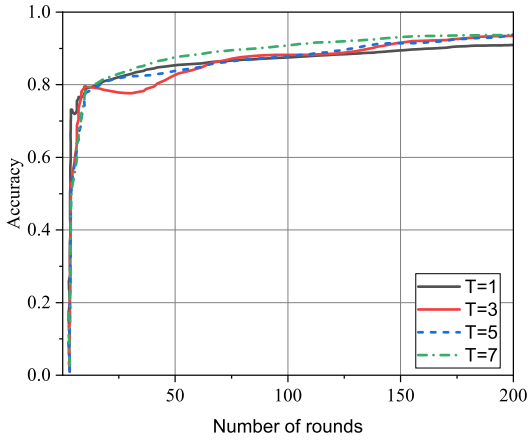

Fig. 6. The prediction accuracy of the model under different values of T

5.3. Performance Under Different Network Delay

In order to observe the impact of the heterogeneity of the devices and the data on model performance, we first construct the ideal case for network connection. Under the conditions of $\kappa_1 = 10$ and $\kappa_2 = 5$, we design four different T to compare the influence of system heterogeneity on the model convergence. The values of T are 1, 3, 5, and 7, respectively. The number of training rounds for both the RPS and the client is 200, that is, $B\kappa_1\kappa_2 = B\kappa_3 = 200$. The results on model prediction accuracy with different T are shown in Fig.6. It can be found that due to the difference in computing capability of each node participating in the HFL, different convergence trends of the global model will appear. In particular, when T is set to a small value, it can simulate the situation that the node cannot tolerate the existing delay being large during the parameter transmission process. In Fig.6, when $T = 1$, nodes will select asynchronous aggregation for model training more time. In addition, since the number of nodes participating in asynchronous aggregation during a certain round of training is less than the total number of nodes, in the early stage of model training, the accuracy curve corresponding to $T = 1$ fluctuates relatively significantly.

Table 3. Model prediction accuracy and switching times after 200 rounds of training under different T

	HFL(T=1)	HFL(T=3)	HFL(T=5)	HFL(T=7)
Running Time(sec)	7869.408	11079.721	12547.851	13093.137
EVC	0.909	0.934	0.937	0.936
Async_Freq	577	50	8	0
Sync_Freq	464	915	973	987

Table 4. Prediction accuracy of RPS under different T

	RPS 1	RPS 2	RPS 3	RPS 4	RPS 5
	EVC				
T = 1	0.827	0.931	0.846	0.887	0.918
T = 3	0.825	0.930	0.842	0.892	0.921
T = 5	0.830	0.934	0.844	0.892	0.923
T = 7	0.831	0.934	0.844	0.890	0.921

Under the above ideal network connection, we record running time, *EVC* for the global model, and switching times of synchronous aggregation method (*Sync_Freq*) and asynchronous aggregation method (*Async_Freq*) corresponding to four sets of T. The results are shown in Table 3. It can be observed that *Sync_Freq* and *Async_Freq* computed by CPS with $T = 1$ are relatively close. Indicating that in ideal circumstances, the delay caused by other influencing factors (such as heterogeneous computing power) averages close to 1. With the increase of T, *Sync_Freq* for CPS will be also increased. Correspondingly, the time to complete the entire training task will be also increased, because the synchronous aggregation needs to wait for all participants to complete their local task. In particular, when $T = 7$, then *Async_Freq* $= 0$, it means that arbitrary delay can be tolerated, so asynchronous aggregation will not be adopted, that is, the HFL is equivalent to the traditional synchronous FL. Conversely, the HFL is more sensitive to delay when $T \to 1$. More extremely, the HFL is similar to traditional asynchronous FL when $T \to 0$. Similarly, RPS also has its *Sync_Freq* and *Async_Freq*, as shown in Fig.7, where Ri (*Async)* and Ri (*Sync)* are represented as *Async_Freq* and *Sync_Freq* of RPS i respectively. The global model after convergence with different T in Fig.6 are applied to the test datasets of all RPSs respectively, then corresponding prediction accuracy (i,e, *EVC*) can be calculated, as shown in Table 4. It can be found that the difference in terms of test accuracy on the same dataset for different T is almost small.

However, most of the nodes actually participating in FL are mobile devices with limited network bandwidth resources, so there will be a certain communication delay in parameter transmission process, and the delay for uploading parameters is generally larger than that for downloading parameters [25].

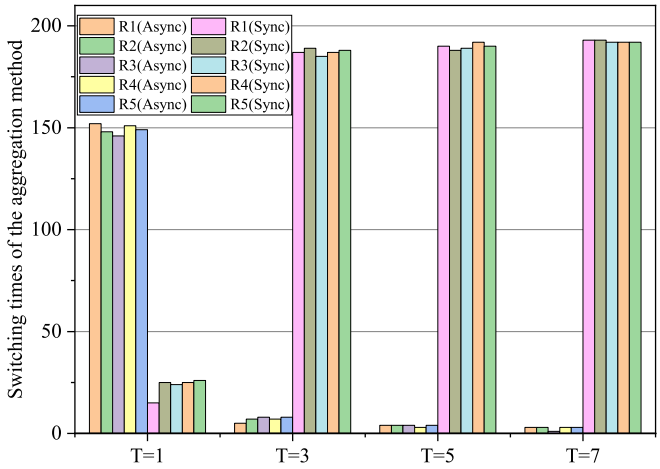

Fig. 7. Switching times of the aggregation method (async or sync) for RPS(1,2,3,4,5) under different T

The specific communication delay is measured as follow [26],

$$T_0 = \frac{\Theta_i}{b_i^r \log\left(1 + o_i \left|G_i^r\right|^2 / \psi\right)} \tag{26}$$

where b_i^r is the allocated bandwidth for node i at round r from the total bandwidth B, i.e. $\sum_i b_i^r = B$, o_i is the transmit power of node i, G_i^r is the average channel gain between node i and its upper server, ψ is the background noise and Θ_i is the size of local model of node i. To simplify the calculation, we assume $B = 20MHz$, $\psi = 10^{-19}$, $o_i = 20dBm$ and $G_i^r = 10$[27].

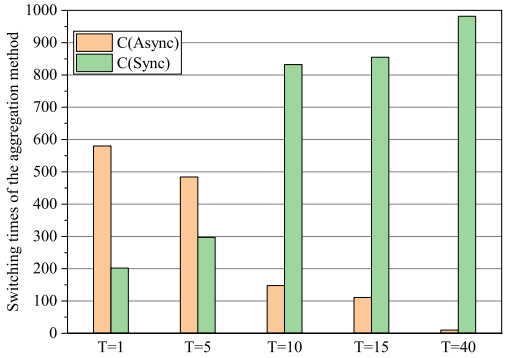

Fig. 8. Switching times of the aggregation method (Async or Sync) for CPS under different T

We further set five different T ($T = 1$, $T = 5$, $T = 10$, $T = 15$, $T = 40$) to verify the validity of the HFL. After 200 rounds local training on clients, multiple *Async_Freq*

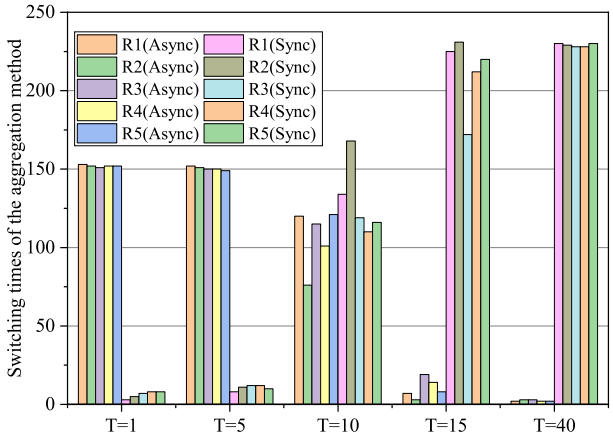

Fig. 9. Switching times of the aggregation method (Async or Sync) for RPS(1,2,3,4,5) under different T

and *Sync_Freq* can be obtained by CPS and RPS respectively, as shown in Fig.8 and Fig.9. And "C(Async)" and "C(Sync)" respectively represent asynchronous(Async_Freq) or synchronous(Sync_Freq) aggregation performed by CPS. When T becomes larger, *Sync_Freq* will be increased synchronously, but *Async_Freq* will be the opposite. In particular, if $1 < T < 40$, CPS will use both synchronous and asynchronous aggregation method during the entire training process, that is, the operation for adopting synchronous or asynchronous aggregation is a dynamic procedure, instead of statically adopting either one approach among them. Similarly, with the increase of T, RPS will also undergo similar synchronous and asynchronous aggregation strategy adjustment during the whole training process. Particularly, *Async_Freq* is close to 0 at $T = 40$ in Fig.9, indicating that the sum of communication and calculation delay is closed to 40, and the overall training process of the HFL will be affected hardly by the delay.

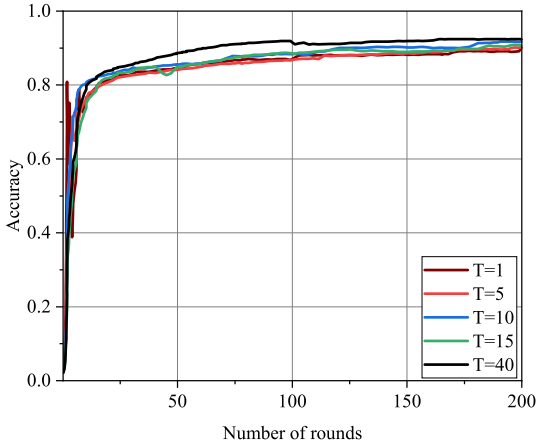

Fig. 10. The prediction accuracy of the model under different T

The prediction accuracy curves of global model corresponding to the above five T are shown in Fig.10. When T is increased, it will reduce the impact of communication delay on the HFL and increase *Sync_Freq*, which makes the model convergence more stable and faster, such as the accuracy curve with $T = 40$, but the time for training model will be also increased. However, the gap of prediction accuracy among them after model convergence is between 0.7% and 2.4%.

Furthermore, the framework proposed in this paper (denoted by Adaptive-HFL) is also compared with some representative models (i.e. ECHFL[10], ADFO[20], AHFL[26]) to demonstrate the its effectiveness. Among them, ECHFL and AHFL are both belong to hierarchical federated learning models, and ADFO is a federated version of adaptive optimizer. When $T = 10$, the comparison result about prediction accuracy is shown in Fig.11, due to the small number of local models of a single server in the hierarchical-based learning structure, there will be certain fluctuations for training, and the curve corresponding to ADFO is relatively smoother because ADFO is designed based on "server-client" network service architecture and adaptive optimization algorithm. However, in general, the hierarchical structure can achieve better prediction effect than ADFO after convergence. At the same time, since the hierarchical correlation and adaptive parameter aggregation scheme are considered in Adaptive-HFL, and more constraints are considered by AHFL model, so the higher prediction accuracy can be achieved. In addition, the accuracy of Adaptive-HFL is about 3.4% higher than that of ADFO. Finally, under the same rounds of training, the results of time required for the four models are shown in Fig.12. It can be seen that the time required for Adaptive-HFL has decreased by 12.6%, 8.4%, and 9.8% than ADFO, AHFL and ECHFL, respectively. Among them, because Adaptive-HFL can adjust the aggregation method between the different layers according to factors such as network communication delay, Adaptive-HFL realizes the model accuracy close to AHFL, but saves 8.4% of the training time required for convergence. This results show that Adaptive-HFL has a significant advantage in convergence performance compared to other three models.

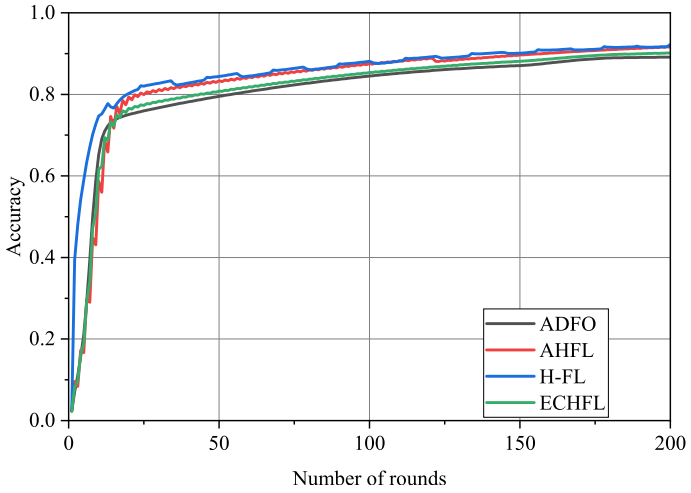

Fig. 11. The accuracy of different algorithms

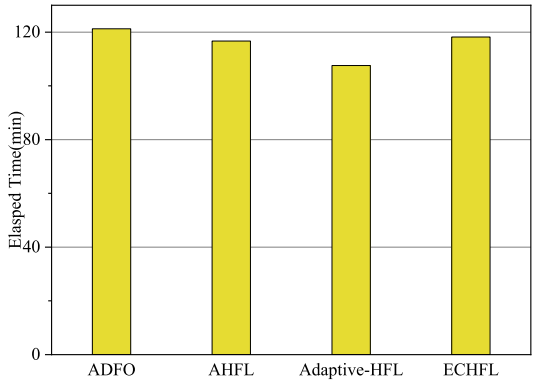

Fig. 12. The comparison of time required for model convergence

6. Conclusion

This paper first proposes a hierarchical federated learning framework, which realizes the joint learning with hierarchical associations between multiple data holders. Meanwhile, we propose a model parameter aggregation algorithm for selecting dynamically asynchronous aggregation/synchronous aggregation to improve the efficiency for training global model. This paper not only conducts a theoretical analysis on the convergence of the proposed model, but also comprehensively evaluates the effectiveness and performance of the proposed framework by taking the prediction of water demand in the urban multi-regional water supply scenario as an experiment. In our future work, we will investigate the matching problem between the efficiency of model training and node selection based on the HFL proposed framework.

Acknowledgments. This work is supported in part by Scientific and Technological innovation project of scientific research institutions of Chongqing (No.cstc2021jxjl20010), in part by Chongqing Technology Innovation and Application Development Key Project under Grant 2022TIAD-KPX0048 and Grant 2022TIAD-KPX0053, in part by Banan District Scientific and Technological Achievements Transformation and Industrialization Project, in part by Graduate Student Innovation Program of Chongqing University of Technology under Grant (No.gzlcx20222061).

References

1. K. M. Ahmed, A. Imteaj, M. H. Amini, Federated deep learning for heterogeneous edge computing, in: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 2021, pp. 1146–1152.
2. M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan, H. V. Poor, Distributed learning in wireless networks: Recent progress and future challenges, IEEE Journal on Selected Areas in Communications 39 (12) (2021) 3579–3605.
3. L. Chettri, R. Bera, A comprehensive survey on internet of things (iot) toward 5g wireless systems, IEEE Internet of Things Journal 7 (1) (2020) 16–32.

4. W. Y. B. Lim, Z. Xiong, J. Kang, D. Niyato, C. Leung, C. Miao, X. Shen, When information freshness meets service latency in federated learning: A task-aware incentive scheme for smart industries, IEEE Transactions on Industrial Informatics 18 (1) (2022) 457–466.
5. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.
6. A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, D. Ramage, Federated learning for mobile keyboard prediction, arXiv preprint arXiv:1811.03604 (2018).
7. J. Domingo-Ferrer, A. Blanco-Justicia, J. Manjón, D. Sánchez, Secure and privacy-preserving federated learning via co-utility, IEEE Internet of Things Journal 9 (5) (2022) 3988–4000.
8. X. Li, K. Huang, W. Yang, S. Wang, Z. Zhang, On the convergence of fedavg on non-iid data, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.
9. C. Xie, S. Koyejo, I. Gupta, Asynchronous federated optimization, CoRR abs/1903.03934 (2019). arXiv:1903.03934.
10. L. Liu, J. Zhang, S. Song, K. B. Letaief, Client-edge-cloud hierarchical federated learning, in: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–6.
11. Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, Y. Zhao, Resource-efficient federated learning with hierarchical aggregation in edge computing, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, 2021, pp. 1–10.
12. W. Y. B. Lim, J. S. Ng, Z. Xiong, J. Jin, Y. Zhang, D. Niyato, C. Leung, C. Miao, Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning, IEEE Transactions on Parallel and Distributed Systems 33 (3) (2022) 536–550.
13. A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, V. Smith, On the convergence of federated optimization in heterogeneous networks, CoRR abs/1812.06127 (2018). arXiv:1812.06127.
14. B. Luo, X. Li, S. Wang, J. Huang, L. Tassiulas, Cost-effective federated learning design, in: IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, 2021, pp. 1–10.
15. Z. Chen, W. Liao, K. Hua, C. Lu, W. Yu, Towards asynchronous federated learning for heterogeneous edge-powered internet of things, Digital Communications and Networks 7 (3) (2021) 317–326.
16. H. Zhu, M. Yang, J. Kuang, H. Qian, Y. Zhou, Client selection for asynchronous federated learning with fairness consideration, in: 2022 IEEE International Conference on Communications Workshops (ICC Workshops), 2022, pp. 800–805.
17. C. Chen, H. Xu, W. Wang, B. Li, B. Li, L. Chen, G. Zhang, Communication-efficient federated learning with adaptive parameter freezing, in: 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 2021, pp. 1–11.
18. J. Liu, H. Xu, L. Wang, Y. Xu, C. Qian, J. Huang, H. Huang, Adaptive asynchronous federated learning in resource-constrained edge computing, IEEE Transactions on Mobile Computing (2021) 1–1.
19. J. Jin, J. Ren, Y. Zhou, L. Lyu, J. Liu, D. Dou, Accelerated federated learning with decoupled adaptive optimization, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, Vol. 162 of Proceedings of Machine Learning Research, PMLR, 2022, pp. 10298–10322.
20. S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, H. B. McMahan, Adaptive federated optimization, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021.
21. Y. Zhao, X. Gong, Quality-aware distributed computation for cost-effective non-convex and asynchronous wireless federated learning, in: 2021 19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt), IEEE, 2021, pp. 1–8.

22. C. Zhou, J. Liu, J. Jia, J. Zhou, Y. Zhou, H. Dai, D. Dou, Efficient device scheduling with multi-job federated learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 9971–9979.
23. Q. Li, Y. Diao, Q. Chen, B. He, Federated learning on non-iid data silos: An experimental study, in: 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, 2022, pp. 965–978.
24. Tensorflow, Tensorflow, https://www.tensorflow.org (2022).
25. S. Q. Zhang, J. Lin, Q. Zhang, A multi-agent reinforcement learning approach for efficient client selection in federated learning, Proceedings of the AAAI Conference on Artificial Intelligence 36 (8) (2022) 9091–9099. doi:10.1609/aaai.v36i8.20894.
26. B. Xu, W. Xia, W. Wen, P. Liu, H. Zhao, H. Zhu, Adaptive hierarchical federated learning over wireless networks, IEEE Transactions on Vehicular Technology 71 (2) (2022) 2070–2083.
27. W. Shi, S. Zhou, Z. Niu, M. Jiang, L. Geng, Joint device scheduling and resource allocation for latency constrained wireless federated learning, IEEE Transactions on Wireless Communications 20 (1) (2021) 453–467.

Zhuo Chen received Ph.D. degree in communication and information systems from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. He is currently an Associate Professor with the College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. His current research interests include distributed machine learning and IoT application.

Chuan Zhou received B.E. degree in Internet-of-Things engineering from Chongqing University of Education, Chongqing, China, in 2020. He currently is pursuing his master's degree in Computer science and technology at the College of Computer Science and Engineering, Chongqing University of Technology, Chongqing, China. His research interests include Federated Learning and big data.

Yang Zhou received Ph.D. degree in the College of Computing at the Georgia Institute of Technology. He is currently an Assistant Professor in the Department of Computer Science and Software Engineering at the Auburn University. His current research interests include trustworthy machine learning, parallel, distributed, and Federated Learning.

Received: September 30, 2022; Accepted: March 10, 2023.