Two-way Delayed Updates with Model Similarity in Communication-Efficient Federated Learning

Yingchi Mao[a,b], Zibo Wang[b], Jun Wu[b], Lijuan Shen[b], Shufang Xu[a,b], and Jie Wu[c]
[a] Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing, China
[b] School of Computer and Information, Hohai University, Nanjing, China
[c] Center for Networked Computing, Temple University, Philadelphia, USA

***Abstract*—The great achievement of IoT and the wide use of edge devices have brought explosive growth in data. The quality and scale of data determine the performances of machine learning models. Federated learning has attracted widespread attention for its ability to use isolated data and protect data privacy. Models can represent excellent generalization capabilities through federated training. However, the large number of devices and complex models involved in federated training exacerbate the communication costs and degrade the performance of the global model. Although existing approaches can reduce communication costs, they ignore the degradation of global model accuracy in a heterogeneous environment. To alleviate the huge communication costs in federated learning, this paper focuses on reducing upstream and downstream communication frequency while ensuring global model accuracy. We propose a Two-way Delayed Updates method with Model Similarity in Communication-Efficient Federated Learning (FedTDMS). FedTDMS employs personalized local computation to improve global model accuracy on heterogeneous data. Combining local update relevance check and global model compensation, FedTDMS reduces the communication frequency in Federated Learning. We conduct experiments on the MNIST-FL and CIFAR-10-FL datasets. Results show that FedTDMS can greatly optimize communication efficiency while maintaining good global model accuracy.**

***Index Terms*—federated learning, data heterogeneity, communication efficiency optimization, communication frequency.**

I. Introduction

With the great achievement of IoT and the wide use of edge devices, massive amounts of data for training complex models can be obtained from various edge devices [1] [2] [3] [4]. IoT technology provides intelligent applications to users [5] [6], the quality and scale of data gathered through edge devices determine the performances of intelligent applications [7]. In real-world scenarios, the scattered distribution and privacy concerns of data result in poor data quality for intelligent applications [8]. Federated learning [9] is capable of providing edge intelligent applications for it can fully and effectively utilize dispersed data while protecting data security. The heterogeneity of federated settings requires huge client-server communication costs during training, [10] which leads to network latency or data loss, slowing down the convergence speed of the global model [11].

Fig. 1 shows the frequent communication process in heterogeneous federated learning. In a federated environment, each communication round can be divided into an upstream phase of clients uploading the local model and a downstream phase of the server distributing the global model. Assuming the number of clients is N, compared with the IID data, training a global model on the Non-IID data requires more communication rounds due to the various data distribution among the clients. Assuming each communication round includes both N^2 rounds of upstream and downstream communication. If the client uploads its local model after each local update, the total communication rounds of all clients is $2N^3$.

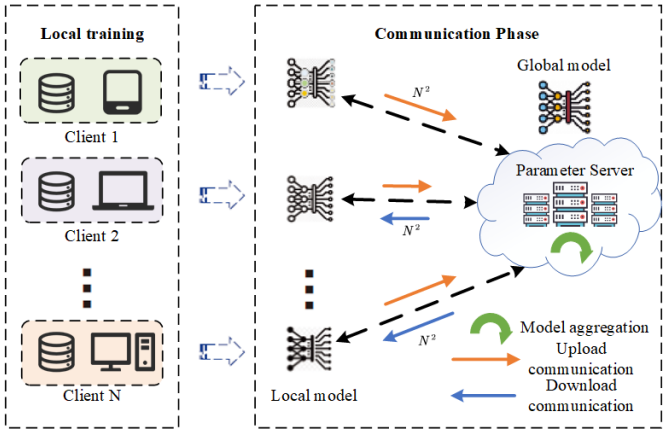

Fig. 1. The frequent communication process in heterogeneous federated learning

Related works reduce the communication frequency between clients and the server to optimize the communication efficiency in federated training [12]. During the upstream communication phase, the multi-round local update is an effective way to reduce communication frequency. FedAvg [13] reduces the frequency of upstream communication by increasing iterations of the local update. CMFL [14] proposes a check mechanism for local model updates to avoid uploading irrelevant updates to the server, thereby reducing communication frequency. However, FedAvg and CMFL only optimize the upstream communication efficiency.

During the downstream communication phase, the communication frequency can be reduced by distributing the model updates to part of the clients. PRLC [15] enables clients to discontinuously download the global model. LAG-PS [16] employs an adaptive delayed update method to skip slowly-varying gradients. However, LAG-PS is unsuitable for heterogeneous data due to the dramatic fluctuations of gradients.

In general, existing approaches only unilaterally reduce the communication frequency upstream or downstream and do not apply to heterogeneous federated learning. Therefore, this paper proposes a Two-way Delayed Updates method with Model Similarity in Communication-Efficient Federated Learning (FedTDMS), considering reducing the communication frequency in both upstream and downstream phases while ensuring the global model accuracy. FedTDMS employs personalized local computation to improve the global model accuracy in heterogeneous environments. By adding to the local update relevance check and global model compensation, FedTDMS delays the upload and download of the model updates simultaneously. Generally speaking, FedTDMS reduces the communication frequency of federated learning while ensuring global model accuracy.

The main contributions of this paper are as follows,

- The personalized local computation is utilized to determine the number of local iterations according to the computing power and data distribution of the client. The personalized local computation mitigates problems such as slow convergence and low global accuracy due to heterogeneous federated learning.
- To reduce upstream communication frequency, a local update relevance check mechanism is employed for identifying and skipping weakly relevant local model updates. Meanwhile, the local update relevance check alleviates gradient conflict and drift caused by data heterogeneity.
- To reduce downstream communication frequency, a global model compensation mechanism is introduced for selecting part of the clients to receive the global model. Furthermore, to ensure global accuracy, unselected clients utilize a local update to imitate the global updates.

The remainder of this paper is organized as follows. Section II presents relevant work on reducing communication frequency. The system model of FedTDMS is shown in Section III. Section IV provides a detailed description of FedTDMS. Section V evaluates the performance of FedTDMS. In the end, Section VI presents the conclusions of this paper.

II. Related Work

Reducing the communication frequency between the client and the server is widely used to optimize communication efficiency in federated training. During the upstream communication phase, allowing for multi-round local update is a common way to reduce communication costs [17] [18]. FedAvg [13] allows the clients to execute a fixed round of stochastic gradient descent (SGD) [19]. The frequency of global model updates in FedAvg drops significantly, thus effectively reducing the upstream communication frequency. Inspired by FedAvg, related researches [19] [20] [21] [22] utilize parallel SGD and variant algorithms based on the multi-round local updates for higher communication efficiency. However, FedAvg and related variant algorithms force each client to perform a fixed round of local update, which is unsuitable for heterogeneous federated learning. FedProx [23] assigns variable rounds of local update to each client by introducing a regularization term, making it adaptable to heterogeneous federated learning.

In addition to the multi-round local update, the delayed aggregation mechanism can also reduce the frequency of upstream communication. CMFL [14] compares the local update direction with the global update trend, preventing invalid updates from being uploaded. LAG-WK [16] utilizes the difference in loss between the local updates and the global updates to detect slowly changing gradients. LAG-WK skips the computation and upload of the slowly changing gradients to reduce upstream communication. However, local gradients of clients change dramatically under strong heterogeneity situations. For both CMFL and LAG-WK, a large amount of gradient uploads are skipped in the prophase of training, damaging global model accuracy.

Moreover, related works reduce the downstream communication frequency for less communication costs. PRLC [15] introduces a delayed update mechanism based on local compensation. Specifically, clients intermittently pull the global model, while clients that have not obtained the global model utilize a local update to imitate the global updates. However, the experiments only show that PRLC can reduce communication costs in a homogeneous environment. Chen et al. [16] proposed LAG-PS based on LAG-WK to check whether there is a slow change during the global updates process. When the changes in the global updates are small or the global model is close to convergence, LAG-PS reuses the outdated model and delays downstream communication.

In general, existing methods only unilaterally reduce the communication frequency upstream or downstream, and the heterogeneity of Federated learning is not fully considered. This paper proposes a Two-way Delayed Updates method with Model Similarity in Communication-Efficient Federated Learning (FedTDMS). FedTDMS can optimize the communication efficiency of heterogeneous federated learning while ensuring global model accuracy.

III. System model

Assuming that the clients set $C = \{c_i \,|i = 1, \cdots, c\}$ and the central server from the federated network, each of the clients owns a private training dataset D_i and an initialized local model w_0. Before each round of training, the server randomly selects S clients for participating in federated training. During the training, clients optimize their local loss function $f_i(w)$, as shown in Equation (1):

$$f_i(w) = \sum_{n=1}^{N} \ell(D_{i,n}; w), \tag{1}$$

where N is the total number of data samples owned by client i, $D_{i,n}$ is the n-th data sample in client i, and $\ell(D_{i,n}; w)$ represents the local loss function. The global model w is shared by all clients and is jointly trained by the S selected clients. Therefore, the objective of federated learning is optimizing the loss function $F(w)$ of the global model to ensure

TABLE I
MAIN SYMBOLIC PARAMETERS IN SECTION III

Symbol	Definition
M	Number of clients
P	Number of parameters of the network
S	Number of clients participating in training
η_l	Local learning rate
c_i	The i-th client participating in training
S_{push}	Set of clients for uploading model updates
V_{comm}^{push}	Upstream communication volume
V_{comm}^{pull}	Downstream communication volume
w_i^r	Local model of client c_i in r-th iteration
w^r	Global model in r-th iteration
$\triangle w_i^r$	Local model updates of client c_i in r-th iteration
$\triangle w^r$	Global model updates in r-th iteration

that the average loss of all clients is minimized, as shown in equation (2):

$$\min_{w} F(w) = \frac{1}{S}\sum_{i=1}^{S} f_i(w). \tag{2}$$

Usually, SGD is used to optimize Equation (2). Specifically, the local updates $\triangle w_i$ are calculated by participating clients based on their private data and are uploaded to the server subsequently. The server averages all local updates to update the global model, as shown in Equation (3):

$$w_{r+1} = w_r + \frac{1}{S}\sum_{i=1}^{S} \triangle w_i, \tag{3}$$

where w_r represents the global model in the t-th round of training. w_{r+1} is the global model of $r+1$-th round updated based on the average local updates.

After aggregating the local updates, the server distributes the global model of $r+1$-th round to all clients for updating their local model, as shown in Equation (4):

$$\forall i \in C \ w_{r+1}^i = w_{r+1}, \tag{4}$$

where w_{r+1}^i is the local model owned by client i of $r+1$-th round.

Assuming the federated training converges after T rounds of iterations, each round of iterations involves one upstream communication and one downstream communication. The total communication volume is V_{comm}, as shown in Equation (5):

$$V_{comm} = \sum_{r=1}^{T}\left(\sum_{i=1}^{S} V(w_r^i) + \sum_{i=1}^{S} V(w_r)\right), \tag{5}$$

where $V(w)$ is used to calculate the size of w, $\sum_{i=1}^{S} V(w_r^i)$ refers to the upstream communication volume V_{comm}^{push} and $\sum_{i=1}^{S} V(w_r)$ refers to the downstream communication volume V_{comm}^{pull}. According to Equation (5), the optimization objective is reducing the frequency of uploading or downloading model updates, furthermore reducing the total communication volume V_{comm} in federated learning.

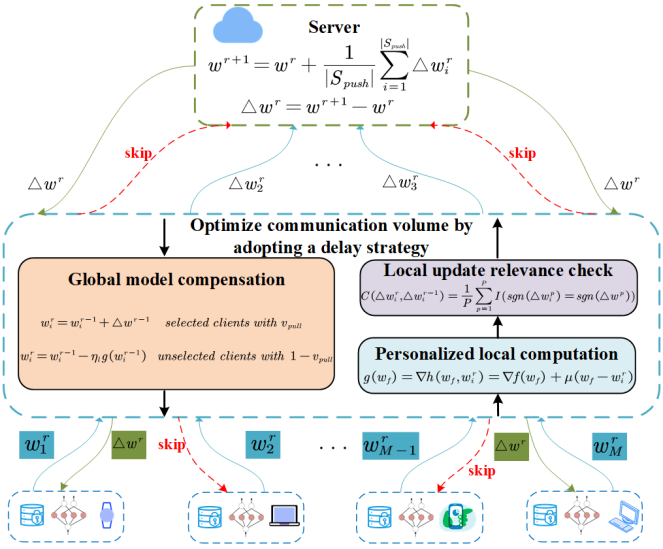

Fig. 2. The framework of FedTDMS

However, unilaterally reducing the frequency of model updates leads to missing information during global model aggregation, exacerbating global model accuracy. This phenomenon is particularly evident in the heterogeneous environment. Therefore, the overall optimization objective is to achieve efficient communication by reducing both V_{comm}^{push} and V_{comm}^{pull} while ensuring global model accuracy.

The main symbolic parameters of FedTDMS proposed in this paper are shown in Table I.

IV. The Design of FedTDMS

A. Overall Framework

Fig. 2 illustrates the framework of FedTDMS. In each training round, FedTDMS can be divided into three phases: (1) personalized local computation, (2) local update relevance check, and (3) global model compensation. Phases (1) and (2) reduce the frequency of clients uploading local model updates, thus reducing the upstream communication volume, while phase (3) reduces the frequency of the server distributing the global updates, thus reducing the downstream communication volume. Additionally, phases (1) and (3) can alleviate the decrease in global model accuracy, thereby maintaining good global model accuracy while optimizing communication efficiency.

B. Personalized Local Computation

Personalized local computation allows FedTDMS to determine the iterations of local update according to the computing power and data distribution of each client. Inspired by FedProx [23], we introduce a regularization term into the loss function to construct variable local iterations for different clients, as shown in Equation (6):

$$g(w_f) = \nabla h(w_f, w_i^r) = \nabla f(w_f) + \mu(w_f - w_i^r), \tag{6}$$

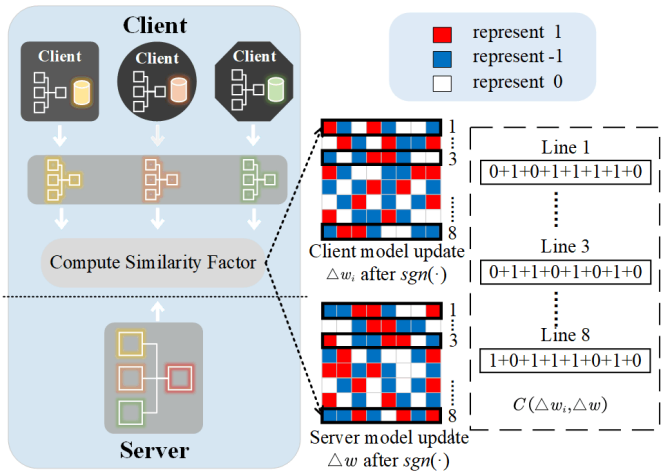

Fig. 3. The illustration of the local update relevance check

where $g(w_f) = \nabla h(w_f, w_i^r)$ is the loss function with the regularization term $\mu(w_f - w_i^r)$, μ is the regularization coefficient, and w_f is the local model during the local update process. In the first round of local iteration, it has $w_f = w_i^r$, where w_i^r is the initial model of client c_i in communication round r. As the client performs personalized local computation among iterations, w_f gradually biases towards the local data distribution of client i, and the local model update process is shown in Equation (7):

$$w_f = w_f - \eta_l g(w_f), \tag{7}$$

where η_l is the local learning rate. Equation (7) is used to perform a local model update after each local iteration.

By allowing clients to perform variable iterations of local update, personalized local computation alleviates problems such as slow convergence and low accuracy of the global model.

C. Local Update Relevance Check

During the training process, some local model updates either have the same direction with the global model updates or contribute little to the global model updates. When the global model is close to convergence, the similitude between the updates of the model on both sides is extremely high. Therefore, it is advisable to skip similar local model updates to reduce upstream communication volume. Based on the above, the local update relevance check is proposed to examine whether the update trend of the models on both sides is consistent. Local update relevance check allows clients to skip uploading the highly correlated local updates to decrease the frequency of upstream communication.

Before performing the local update relevance check, the local model updates $\triangle w_i$ is firstly obtained, as shown in Equation (8):

$$\triangle w_i = w_f - w_i^r. \tag{8}$$

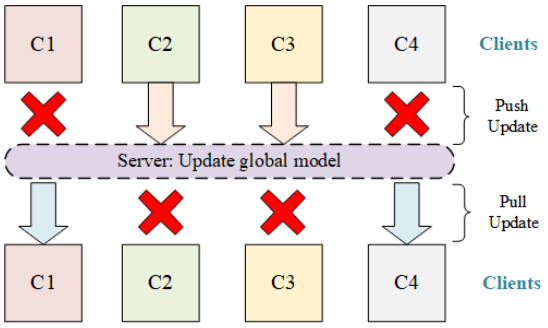

Fig. 4. The two-way delayed updates mechanism of FedTDMS

Next, the number of parameters with consistent update direction between the models on both sides is calculated to obtain the update correlation coefficient, as shown in Equation (9):

$$C(\triangle w_i, \triangle w) = \frac{1}{P}\sum_{p=1}^{P} I(sgn(\triangle w_i^p) = sgn(\triangle w^p)), \tag{9}$$

where P represents the number of updated parameters, and $sgn(\cdot)$ is the symbolic function. $I(sgn(\triangle w_i^p) = sgn(\triangle w^p)) = 1$ represents that the update direction of the p-th parameter between the models on both sides is identical. The update correlation coefficient $C(\triangle w_i, \triangle w) \in [0, 1]$ represents the proportion of the number of parameters with the same update direction to the number of total parameters. The bigger $C(\triangle w_i, \triangle w)$ is, the updates between the models on both sides are more similar. Therefore, it is necessary to set a threshold for $C(\triangle w_i, \triangle w)$ to determine whether the client meets the delayed upload standard.

We use the local similarity coefficient v_{client} as the threshold of $C(\triangle w_i, \triangle w)$. $C(\triangle w_i, \triangle w) < v_{client}$ indicates that the local updates of c_i are weakly correlated with the global model updates and needed to be uploaded; $C(\triangle w_i, \triangle w) > v_{client}$ indicates that the local updates of c_i are similar to the global model updates and the upstream communication of c_i in current round can be skipped. Fig. 3 is the illustration of the local update relevance check of FedTDMS.

By identifying and skipping weakly correlated local model updates, the local update relevance check reduces the frequency of upstream communication. Moreover, the mechanism alleviates problems such as gradient conflicts and gradient drift caused by data heterogeneity.

D. Global Model Compensation

During the downstream communication phase, the server usually distributes the global updates to a section of the clients for less communication costs. However, this operation results in differences between the models on both sides. The difference becomes more severe as the training proceeds, ultimately reducing the accuracy of the global model and even causing it to fail to converge. Therefore, global model compensation is proposed to enable clients who have not

Algorithm 1 Two-way Delayed Updates method with Model Similarity in Communication-Efficient Federated Learning (FedTDMS)

Input: local model updates $\triangle w$, local learning rate η_l, local training epochs E, global learning rate η_g, number of iterations R, number of clients N, current iteration round r, regularization coefficient μ

Output: the global model w^{R+1}

for $r = 1, 2, \cdots, R$ **do**
  Client:
  Define $S \subseteq \{1, 2 \cdots, N\}$
  for Client $i \in S$ **do**
    Download model update $\triangle w^{r-1}$
    if i *is selected with* v_{pull} **then**
      $w_f = w_i^r = w_i^{r-1} + \triangle w^{r-1}$
    else
      $w_f = w_i^r = w_i^{r-1} - \eta_l g\left(w_i^{r-1}\right)$
    end if
    for $e = 1, 2, \cdots, E$ **do**
      $g\left(w_f\right) = \nabla h\left(w_f, w_i^r\right) = \nabla f\left(w_f\right) + \mu\left(w_f - w_i^r\right)$
      $w_f = w_f - \eta_l g\left(w_f\right)$
    end for
    $\triangle w_i^r = w_f - w_i^r$
    $C\left(\triangle w_i^r, \triangle w^{r-1}\right) = \frac{1}{P}\sum_{p=1}^{P} I(sgn\left(\triangle w_i^p\right) = sgn\left(\triangle w^p\right))$
    if $C\left(\triangle w_i^r, \triangle w^{r-1}\right) < v_{client}$ **then**
      Communicate $\triangle w_i^r$
    else
      Communicate (NULL)
    end if
  end for
  Server:
  $w^{r+1} = w^r + \frac{1}{|S_{push}|}\sum_{i=1}^{|S_{push}|} \triangle w_i^r$
  $\triangle w^r = w^{r+1} - w^r$
  Communicate $\triangle w^r$ **to selected clients with** v_{pull}
end for

received the global model to compensate for the gap with the global model by performing local update.

Specifically, during the downstream communication phase, the clients selected with probability v_{pull} update their local model using global model updates, as shown in Equation (10):

$$w_i^r = w_i^{r-1} + \triangle w, \tag{10}$$

Clients that have not received the global model updates compensate for the gap with the global model by performing local update, as shown in Equation (11):

$$w_i^r = w_i^{r-1} - \eta_l g\left(w_i^{r-1}\right) \tag{11}$$

where $g\left(w_i^{r-1}\right)$ is the client that has not received the global model updates, and η_l represents the local learning rate.

As shown in Fig. 4, clients C1 and C4 delay the upstream communication by utilizing local update relevance check, and clients C2 and C3 delay the downstream communication using global model compensation, achieving two-way delayed updates.

E. Algorithm Design

During the training process, clients selected with v_{pull} update their local model according to the global updates, the rest of the clients compensate for the gap with the global model by performing personalized local computation. The Local update relevance check is performed subsequently to decide whether to upload local updates. Then the server updates the global model while delivering it to selected clients. Algorithm 1 shows the detailed steps of FedTDMS.

V. Performance Evaluation

A. Experiment Setup

1) Federated datasets and parameter settings: We select the MNIST dataset and CIFAR-10 dataset for experiments. For the purpose of imitating the data distribution among clients in a heterogeneous federated network, we employ the Dirichlet distribution $Dir\left(\alpha\right)$ to generate heterogeneous federated datasets of various degrees, named as MNIST-FL and CIFAR-10-FL. The larger α is, the more homogeneous the data distribution among clients is.

For the MNIST-FL dataset, the iteration rounds R is set to 100, the batch size $batchsize$ is set to 200, the local learning rate η_l is set to 0.1, the global learning rate η_g is set to 1, and the number of clients N is set to 100. 10 clients are randomly selected for federated training in each round. The regularization coefficient μ is set to 0.01, based on the experimental settings of FedProx. For the CIFAR-10-FL dataset, the iteration rounds R is changed to 200 and the local learning rate η_l is changed to 0.01.

2) Network models: Based on the complexity of the CIFAR-10-FL dataset in terms of class and RGB channels compared to the MNIST-FL dataset, different network models are chosen to handle image classification tasks for the two datasets. Specifically, we use Logistic Regression and AlexNet for the MNIST-FL dataset and CIFAR-10-FL dataset, respectively. So that we can verify the communication efficiency of FedTDMS on both convex and non-convex models.

3) Experimental baselines: FedAvg [13], as a typical communication-efficient approach, is selected to be one of the baselines. Additionally, we compare FedTDMS with CMFL [14], which optimizes communication efficiency according to the global model update trend and the local update direction, and PRLC [15], which utilizes the intermittently pulling and local update to reduce communication costs.

The reduction in communication volume represents the skipped communication volume due to upstream and downstream communication delays when the global model reaches a specified accuracy.

B. Analysis of Experiment Results

1) Analysis of hyperparameter selection for FedTDMS: In order to select suitable hyperparameters for subsequent

TABLE II
REDUCTION IN COMMUNICATION VOLUME (KB) WITH v_{client}

v_{client}	Dataset					
	MNIST-FL			CIFAR-10-FL		
	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$
0.60	**182.48**	**184.38**	**186.22**	**814.75**	**892.59**	**931.13**
0.65	29.74	30.03	30.28	126.76	181.07	200.18
0.70	4.66	4.96	5.18	39.02	45.32	50.00
0.75	2.17	2.52	2.78	11.05	15.11	16.66
0.80	1.73	1.75	1.77	4.75	7.77	8.83

TABLE III
GLOBAL MODEL ACCURACY (%) WITH v_{client}

v_{client}	Dataset					
	MNIST-FL			CIFAR-10-FL		
	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$
0.60	92.35	92.44	**92.61**	60.10	64.22	**67.73**
0.65	92.41	92.45	92.49	60.43	64.86	67.65
0.70	92.40	92.46	92.55	60.87	65.29	67.42
0.75	92.37	92.47	92.54	61.69	65.63	67.26
0.80	**92.43**	**92.47**	92.54	**61.85**	**65.71**	67.58

TABLE IV
REDUCTION IN COMMUNICATION VOLUME (KB) WITH v_{pull}

v_{pull}	Dataset					
	MNIST-FL			CIFAR-10-FL		
	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$
0.1	**2046.16**	**2067.26**	**2088.35**	**8754.33**	**8859.80**	**8965.27**
0.3	1591.46	1607.87	1624.27	7055.02	7137.05	7219.08
0.5	1136.76	1148.48	1160.20	5156.48	5215.08	5273.67
0.7	682.05	689.09	696.12	2847.80	2883.95	2918.11
0.9	227.35	229.70	232.04	996.14	1007.86	1019.58

TABLE V
GLOBAL MODEL ACCURACY (%) WITH v_{pull}

v_{pull}	Dataset					
	MNIST-FL			CIFAR-10-FL		
	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$
0.1	89.26	89.31	89.58	50.55	53.85	56.01
0.3	90.20	90.46	90.51	53.94	57.76	61.25
0.5	**92.33**	**92.47**	**92.55**	61.33	64.83	**67.56**
0.7	92.28	92.42	92.53	60.79	64.15	67.30
0.9	92.32	92.40	92.50	**61.98**	**64.92**	67.48

experiments, we evaluate the performance of FedTDMS on the reduction in communication volume and global model accuracy with different hyperparameters. Specifically, we conduct experiments on the MNIST-FL dataset and CIFAR-10-FL dataset under varying degrees of data heterogeneity ($\alpha = 0.5$, $\alpha = 1$, and $\alpha = 10$). Table II to Table V show the results.

In regard to local correlation coefficient v_{client}, it is found that when $v_{client} < 0.6$, almost all the participating clients have a local update correlation coefficient greater than v_{client}. This means all the clients participating in federated learning skip upstream communication, thus causing the federated training fails. When $v_{client} > 0.8$, all the local update correlation coefficients of the participating clients are less than v_{client}, which means that all clients need to upload local model updates, resulting in the reduction in communication volume equal to 0. Therefore, the value of v_{client} should be taken within $[0.6, 0.8]$ to make it meaningful. The related experiments only show the results with v_{client} in $[0.6, 0.8]$.

As shown in Table II, as the value of v_{client} increases, the reduction in communication volume decreases significantly. Table III shows that, in high data heterogeneity situation ($\alpha = 0.5$, $\alpha = 1$), as the value of v_{client} increases, the global model accuracy improves slowly. Under low data heterogeneity conditions ($\alpha = 10$), the influence of v_{client} on global model accuracy is not significant, and the global model accuracy is highest for $v_{client} = 0.6$.

Based on the above analysis, it can be concluded that when $v_{client} = 0.6$, the global model accuracy decreases slightly under conditions of high data heterogeneity. While in low data heterogeneity situation, the minimum upstream communication volume and the maximum global model accuracy can be obtained simultaneously. Therefore, setting v_{client} to 0.6 can significantly improve communication efficiency while sacrificing a little global model accuracy.

As shown in Table IV, as the value of v_{pull} increases, the reduction in communication volume decreases significantly. It can be seen from Table V that on the MNIST-FL dataset, it has the highest global model accuracy when $v_{pull} = 0.5$. On the CIFAR-10-FL dataset, in high data heterogeneity situation ($\alpha = 0.5$, $\alpha = 1$), it has the highest global model accuracy when $v_{pull} = 0.9$. In low data heterogeneity situation ($\alpha = 10$), it has the highest global model accuracy when $v_{pull} = 0.5$. To balance the global model accuracy against downstream communication volume, we choose $v_{pull} = 0.5$. In this case, FedTDMS can reduce downstream communication volume and maintain good global model accuracy.

2) Analysis of reduction in communication volume: Fig. 5 illustrates the experimental results of the reduction in communication volume for FedAvg, CMFL, PRLC, and FedTDMS on the MNIST-FL dataset.

Fig. 5(a) shows that, in high data heterogeneity situation ($\alpha = 0.5$), FedTDMS has the maximum reduction in communication volume. Compared to CMFL, which only optimizes upstream communication, and PRLC, which only optimizes downstream communication, FedTDMS improves communication efficiency by 718.4%, 641.5%, 616.5%, and 16.5%, 15.2%, 14.7%, respectively. FedTDMS reduces the upstream and downstream communication volume by utilizing local update relevance check and global model compensation, greatly improving communication efficiency in federated learning. PRLC has significantly better communication efficiency than CMFL for the mechanism of randomly selecting clients to receive global updates greatly reducing the communication costs in the downstream communication phase. Although CMFL skips communicating the model parameters unrelated to global model updates, a large number of model parameters

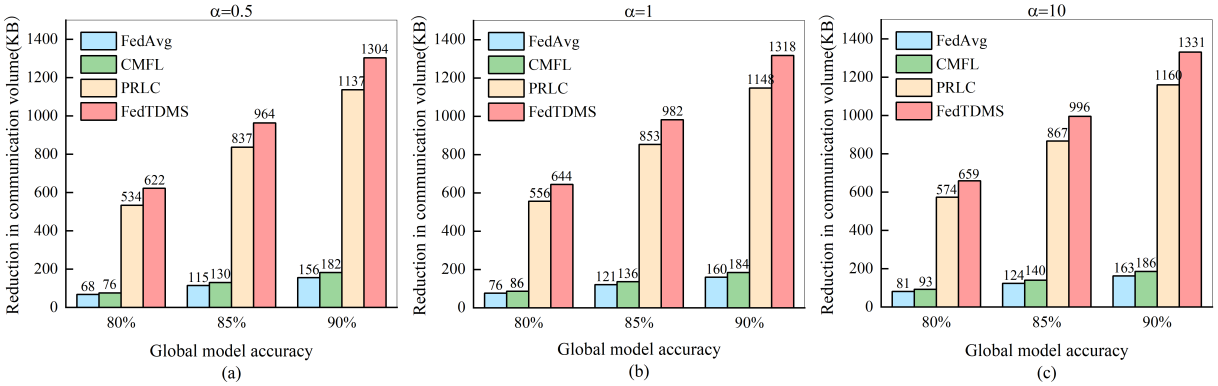

Fig. 5. Reduction in communication volume on the MNIST-FL dataset for FedAvg, CMFL, PRLC, and FedTDMS

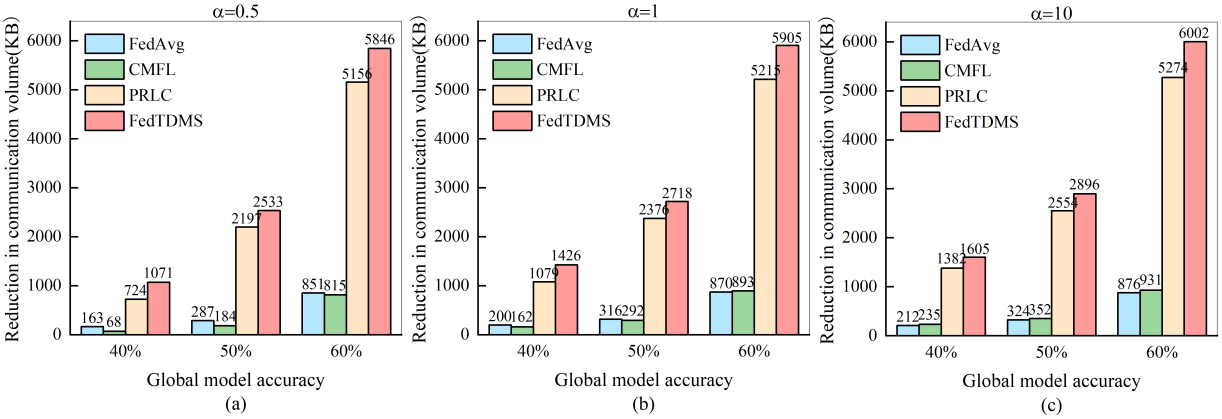

Fig. 6. Reduction in communication volume on the CIFAR-10-FL dataset for FedAvg, CMFL, PRLC, and FedTDMS

still need to be uploaded, resulting in lower communication efficiency compared to PRLC. FedAvg has the lowest reduction in communication volume, its multi-round local update mechanism needs to be improved for better communication efficiency.

The results of optimizing communication efficiency on the weakly heterogeneous MNIST-FL dataset are shown in Fig. 6(c). FedTDMS shows a more significant improvement in optimizing communication efficiency; the reduction in communication volume reaches 659KB, 996KB, and 1331KB, respectively under different model accuracies, outperforming the other three methods.

Fig. 6 illustrates the experimental results of optimizing communication volume for FedAvg, CMFL, PRLC, and FedTDMS on the CIFAR-10-FL dataset. Similar to the results obtained on the MNIST-FL dataset, FedTDMS achieved the maximum reduction in communication volume under different data heterogeneity degrees when global model accuracy reaches 40%, 50%, and 60%, respectively.

Based on the above analysis, it can be concluded that FedTDMS outperforms FedAvg, CMFL, and PRLC in optimizing communication efficiency under different degrees of data heterogeneity. Additionally, all experimental approaches show a further improvement in communication efficiency as the degree of data heterogeneity weakens. This suggests that high data heterogeneity is indeed a bottleneck for optimizing communication efficiency in federated learning. FedTDMS can significantly reduce high communication volume by reducing both upstream and downstream communication frequency.

3) Analysis of global model accuracy: Table VI demonstrates the global model accuracy of FedAvg, CMFL, PRLC, and FedTDMS on the MNIST-FL dataset and CIFAR-10-FL dataset with different heterogeneity degrees. Columns 2 to 4 in Table VI show that for the MNIST-FL dataset, the four approaches have little difference in global model accuracy. Specifically, the CMFL method achieved the highest global model accuracy of 92.58% under low data heterogeneity conditions ($\alpha = 10$); as the data heterogeneity increased ($\alpha = 1$, $\alpha = 0.5$), FedTDMS achieved the highest global model accuracy with 92.43%, and 92.35%, respectively, slightly better than CMFL; PRLC had the lowest global model accuracy under different heterogeneity degrees, and FedAvg had the second lowest global model accuracy.

As the model and dataset become more complex, the difference in global model accuracy among the four approaches is more obvious. Columns 5 to 7 in Table VI show that on the CIFAR-10-FL dataset, FedTDMS achieves the highest global model accuracy of 67.54% in low data heterogeneity

TABLE VI
GLOBAL MODEL ACCURACY OF DIFFERENT APPROACHES

Comparison approaches	Dataset					
	MNIST-FL			CIFAR-10-FL		
	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$
FedAvg	92.19	92.33	92.50	57.13	61.25	65.17
CMFL	92.25	92.42	**92.58**	**62.96**	**65.40**	66.48
PRLC	92.14	92.21	92.49	55.24	60.36	64.52
FedTDMS	**92.35**	**92.43**	92.57	61.01	64.09	**67.54**

situations ($\alpha = 10$); under high data heterogeneity conditions ($\alpha = 1$, $\alpha = 0.5$), the global model accuracy of FedTDMS is lower than that of CMFL, with a decrease of about 2.0%, and 3.1%, respectively. However, the global model accuracy of FedTDMS is still significantly higher than that of FedAvg and PRLC, with PRLC suffering a significant decrease in global model accuracy due to random updates of client models.

Analysis of the above experimental results reveals that PRLC exacerbates model drift and conflict in a heterogeneous environment, resulting in serious damage to the global model accuracy. FedAvg forces clients to perform fixed rounds of local computation, resulting in poor adaptability to the heterogeneous environment. Overall, the experimental results of FedTDMS are not as good as CMFL, as FedTDMS sacrifices some global model accuracy for communication efficiency during downstream communication. However, the personalized local computation and the global model compensation mechanism employed by FedTDMS alleviate the low accuracy of the global model due to delayed upload communication, and the global accuracy is comparable to CMFL. Thus, FedTDMS optimizes communication efficiency in heterogeneous federated learning while ensuring global model accuracy.

VI. Conclusions

This paper proposes a Two-way Delayed Updates method with Model Similarity in Communication-Efficient Federated Learning (FedTDMS). FedTDMS employs personalized local computation to alleviate the negative impact of heterogeneous data on global model accuracy. Combining local update relevance check and global model compensation, FedTDMS delays the upload and download of model updates. Experimental results show that FedTDMS can greatly optimize the communication efficiency of heterogeneous federated learning while maintaining good global model accuracy.

References

[1] W. Li, W. Meng, and L. T. Yang, "Enhancing trust-based medical smartphone networks via blockchain-based traffic sampling", in *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 122-129, Oct. 2021.

[2] A. Farley, and H. Ham, "Real time IP camera parking occupancy detection using deep learning", *Procedia Computer Science*, vol. 179, pp. 606-614, Jan. 2021.

[3] A. Heidari, N. J. Navimipour, M. Unal, and G. Zhang, "Machine learning applications in internet-of-drones: systematic review, recent deployments, and open issues", *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–45, Mar. 2023.

[4] Y. Liu, W. Yu, T. Dillon, W. Rahayu, and M. Li, "Empowering IoT predictive maintenance solutions with AI: A distributed system for manufacturing plant-wide monitoring", *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1345-1354, Feb. 2022.

[5] A. Razmjoo, A. Gandomi, M. Mahlooji, D. A. Garcia, S. Mirjalili, A. Rezvani, S. Ahmadzadeh, and S. Memon, "An investigation of the policies and crucial sectors of smart cities based on IoT application", *Applied Sciences*, vol. 12, no. 5, pp. 2672, Mar. 2022.

[6] K. Fizza, A. Banerjee, K. Mitra, P. P. Jayaraman, R. Ranjan, P. Patel, and D. Georgakopoulos, "QoE in IoT: A vision survey and future directions", *Discover Internet Things*, vol. 1, no. 1, pp. 1-14, Dec. 2021.

[7] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Artificial intelligence applications for industry 4.0: A literature-based study", *Journal of Industrial Integration and Management*, vol. 7, no. 1, pp. 83-111, Mar. 2022.

[8] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, and B. He, "A survey on federated learning systems: Vision hype and reality for data privacy and protection", *IEEE Transactions on Knowledge and Data Engineering*, Nov. 2021.

[9] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence", *arXiv:1610.02527*, 2016.

[10] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions", *High-Confidence Computing*, vol. 1, no. 1, 2021.

[11] Y. Zhou, Y. Fu, Z. Luo, M. Hu, D. Wu, Q. Z. Sheng, and S. Yu, "The role of communication time in the convergence of federated edge learning", *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3241-3254, Jan. 2022.

[12] W. Liu, L. Chen, and W. Zhang, "Decentralized federated learning: Balancing communication and computing costs", *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 131-143, Fed. 2022.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, "Communication-efficient learning of deep networks from decentralized data", in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273-1282, Apr. 2017.

[14] L. Wang, W. Wang, and B. Li, "CMFL: Mitigating communication overhead for federated learning", in *2019 IEEE 39th international conference on distributed computing systems (ICDCS)*, pp. 954-964, Jul. 2019.

[15] H. Wang, Z. Qu, S. Guo, X. Gao, R. Li, and B. Ye, "Intermittent pulling with local compensation for communication-efficient federated learning", *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 779-791, Dec. 2020.

[16] T. Chen, G. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning", *Advances in neural information processing systems*, vol. 31, Dec. 2018.

[17] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization", *International Conference on Artificial Intelligence and Statistics*, vol. 108, pp. 2021-2031, 2020.

[18] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning", *Proceedings of the National Academy of Sciences*, vol. 118, no. 17, pp. 1-8, Mar. 2021.

[19] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning", in *International Conference on Machine Learning*, vol. 119, pp. 5132-5143, 2020.

[20] H. Yuan, and T. Ma, "Federated accelerated stochastic gradient descent", *Advances in Neural Information Processing Systems*, vol. 33, pp. 5332-5344, 2020.

[21] D. A. Emre Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization", *International Conference on Learning Representations*, 2021.

[22] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization", *Advances in neural information processing systems*, vol. 33, pp. 7611-7623, 2020.

[23] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks", *Proceedings of Machine Learning and Systems*, vol. 2, pp. 429-450, 2020.