Journal of Systems Architecture 150 (2024) 103108

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Fed-MPS: Federated learning with local differential privacy using model parameter selection for resource-constrained CPS

Shui Jiang [a], Xiaoding Wang [a,c,*], Youxiong Que [c], Hui Lin [a,b]

[a] College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China
[b] Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, Fuzhou 350117, China
[c] National Key Laboratory for Tropical Crop Breeding, Institute of Tropical Bioscience and Biotechnology, Chinese Academy of Tropical Agricultural Sciences, Sanya, 572024, China

ARTICLE INFO

Keywords:
Cyber-Physical Systems
Distributed learning
Federated learning
Differential privacy

ABSTRACT

In Cyber-Physical Systems (CPS), distributed learning is essential for efficiently handling complex tasks when sufficient resources are available. However, when resources are limited, traditional distributed learning struggles to complete even simple tasks and presents a risk of privacy leakage. As a promising distributed learning paradigm, federated learning only requires the client to send the trained model to the server instead of private data, thereby preserving the client's privacy to some extent. However, with the rapid development of artificial intelligence technology, attack methods such as inference attacks still cause privacy leakage for clients participating in federated learning. Moreover, due to its distributed learning nature, federated learning cannot escape the dilemma of model accuracy being constrained by resources. To address the aforementioned problems, this paper proposes a Federated local differential privacy scheme using Model Parameter Selection, named Fed-MPS, for resource-constrained CPS. Specifically, to resolve the issue of limited CPS resources, Fed-MPS adopts an update direction consistency-based parameter selection algorithm in federated learning to extract parameters that enhance model accuracy for subsequent training, thereby improving the final model accuracy and reducing communication overhead. Furthermore, Fed-MPS applies the local differential privacy mechanism to further enhance clients' privacy. By adding noise only to the chosen parameters, the privacy budget is significantly reduced while ensuring model accuracy. Through privacy analysis, we prove that the proposed Fed-MPS scheme satisfies $(\epsilon, \delta) - DP$. Additionally, convergence analysis guarantees that Fed-MPS will converge to the global optimum with a convergence ratio of $O(\frac{1}{T^2})$ within T rounds of federated learning. Extensive experiments on prominent benchmark datasets Cifar10, Mnist, and FashionMNIST demonstrate that, compared with baseline schemes, the proposed Fed-MPS provides higher model accuracy for CPS under resource constraints.

1. Introduction

Cyber-Physical Systems (CPS) represent the next-generation intelligent architecture that seamlessly integrates computing, communication, and control [1]. This framework successfully enables interaction with physical processes through human–computer interaction interfaces and facilitates the remote, reliable, real-time, secure, and collaborative manipulation of physical entities through the network. The implementation of distributed learning in CPS [2] has led to the development of a CPS-based distributed learning architecture. As a powerful distributed learning paradigm, federated learning offers efficient model training [3] and a certain degree of privacy protection [4]. In federated learning, each client is responsible for training a local model from the initial one using its private data and then transmits the trained model to the server for model aggregation. Subsequently, the server distributes the aggregated model to all clients as the latest initial model. This iterative process continues until the convergence condition is met [5,6]. The architecture that combines CPS and federated learning involves two types of participants, namely the client and server, and is established within the three-layer network of CPS [7], comprising the perception layer, network layer, and control layer. The local device is part of the perception layer, and the transmission between the client and server traverses the network layer, while the server operates within the control layer. However, the storage and computing resources of devices in CPS are often limited. Therefore, there might be impossible

* Corresponding author at: College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China.
E-mail addresses: jiangshui87176@gmail.com (S. Jiang), wangdin1982@fjnu.edu.cn (X. Wang), queyouxiong@itbb.org.cn (Y. Que), linhui@fjnu.edu.cn (H. Lin).

https://doi.org/10.1016/j.sysarc.2024.103108
Received 10 June 2023; Received in revised form 27 January 2024; Accepted 8 March 2024
Available online 15 March 2024

for devices to complete the model training due to resource constraints. Consequently, there is an urgent need to design a more efficient federated learning mechanism tailored to the resource constraints of CPS. Therefore, this paper considers the scenario depicted in Fig. 1, where each client offloads the training task of its local model to the edge. Taking into account the limited resources, the trained models will be "tailored" according to the different edge computing capabilities, and the server will aggregate these models according to the "tailoring rules" during the aggregation process.

It is important to highlight that in federated learning [8], the client only sends its trained model to the server, making it impossible for the server to access the client's private data. This design ensures that models, rather than original data, are shared, providing clients with a certain degree of data privacy [9]. However, with the rapid development of artificial intelligence [10], hackers have demonstrated increasingly sophisticated and diverse attack methods. In particular, hackers can launch inference attacks to reverse-engineer models and infer sensitive information, posing a significant threat to the data privacy of traditional federated learning [11]. To counter such attacks, the concept of differential privacy was introduced by Dwork and Roth [12]. Subsequently, numerous privacy-preserving methods based on differential privacy have been proposed [13]. Most of these methods achieve privacy protection by adding noise to the data, with the intensity of privacy protection being determined by the amount of noise added. The combination of federated learning and differential privacy can effectively safeguard the privacy of clients in federated learning and prevent inference attacks on models. Based on the target, i.e., the client or the server, where the noise is added, the differential privacy of federated learning can be categorized into local differential privacy and global differential privacy. Given that local differential privacy is implemented locally on the client side during model training, it is more efficient in protecting clients' privacy. Although the combination of federated learning and differential privacy can effectively address the problem of data privacy leakage in federated learning, it also introduces another challenge. In differential privacy, the privacy budget increases with the model size and communication rounds, which is inversely proportional to the model accuracy. An excessively large privacy budget leads to low model accuracy, while a small privacy budget compromises data privacy protection. This issue is particularly severe in deep neural networks with a large number of parameters, which is well known by "privacy budget explosion" because of the addition of massive noise to the data for privacy protection.

To address the aforementioned challenges, this paper introduces a Federated local differential privacy scheme using Model Parameter Selection, named Fed-MPS, which significantly reduces communication overhead and privacy budget for resource-constrained CPS. Additionally, our scheme is designed to defend against black-box attacks in inference, where attackers attempt to infer the workings of a model solely through its inputs and outputs, without knowledge of its internal structure and parameters. We summarize the main contributions of this paper as follows:

- To improve the performance of distributed learning under resource constraints in CPS, we propose an update direction consistency based parameter selection algorithm in federated learning to alleviate the problems of high communication overhead and low model accuracy. Specifically, this algorithm chooses model parameters during federated learning training according to their update directions, and only the selected model parameters can participate in the subsequent training, ensuring model accuracy while reducing communication overhead.
- To protect the privacy of federated learning participants, we add Gaussian noise to the trained model to resist inference attacks. Because the noise is only added to the chosen model parameters, the problem of privacy budget explosion is also resolved.
- We conduct comprehensive privacy analysis and convergence analysis on the proposed Fed-MPS scheme. The privacy analysis proves that the proposed Fed-MPS scheme achieves $(\epsilon, \delta) - DP$, where ϵ denotes the privacy budget and δ denotes the approximate max divergence, while the convergence analysis indicates that this scheme can converge to the global optimum with a convergence ratio of $O(\frac{1}{T^2})$ within T rounds of federated learning.
- The extensive experiments are conducted on the benchmark datasets Cifar10, Mnist, and FashionMNIST to evaluate the performance of Fed-MPS by comparing with contemporary algorithms CMFL and LDP-Fed. The experimental results demonstrate that the proposed scheme outperforms the baselines by providing better model accuracy in resource constrained CPS.

The remainder of this paper is organized as follows. The related work is introduced in Section 2. Section 3 elaborates implementation details of the proposed Fed-MPS scheme. Section 4 provides both the privacy analysis and convergence analysis. Section 5 gives the performance evaluation. Section 6 concludes this paper.

2. Related work

Given the stringent resource constraints in CPS, traditional distributed learning models often fall short of meeting task requirements, thus necessitating a high degree of optimization for distributed learning models. Currently, related research mainly falls into two categories: model optimization under federated learning and model optimization under federated learning with differential privacy protection. A comparison of the current mainstream research on federated differential privacy is outlined in Table 1.

2.1. Model optimization for federated learning

Differing from the conventional federated fearning architecture, Zhou et al. have developed a multi-center aggregation structure at the global level, which learns multiple global models based on the dynamically updated local model weights [14]. At the local level, they have designed a hierarchical neural network structure that includes personalized and federated modules to address issues of data and model heterogeneity, thereby enhancing the performance of personalized federated learning in the context of Metaverse data. Zhou et al. [15] proposed a peer-to-peer based privacy-aware asynchronous federated learning framework for secure and resilient decentralized model training of modern mobile robotic systems. Additionally, they introduced a clustering-based approach to select participants for federated learning using social context data to enhance the safety and performance of federated learning [16]. Xu et al. [17] proposed a privacy-preserving federated learning method using a function-based encryption protocol, reducing training time, data transfer, and improving communication efficiency. Sattler et al. [18] developed a compression framework tailored to meet the requirements of federated learning, extending the existing top-k gradient sparsification compression technique with a mechanism for downstream compression and weight updating, resulting in improved communication efficiency by reducing the model size and privacy budget. Yang et al. [19] proposed an iterative algorithm to address energy-efficient transmission and computational resource allocation for federated learning over wireless communication networks, significantly reducing energy consumption in wireless communication networks. Meng et al. [20] introduced an efficient and privacy-enhancing federated learning scheme for industrial artificial intelligence, which is non-interactive and prevents privacy data leakage even if multiple entities collude. Reisizadeh et al. [21] proposed an efficient communication federated learning method with periodic averaging and quantization to address communication and scalability challenges in federated learning. A Hierarchical Hybrid Network model is constructed to describe the multi-type relationships between

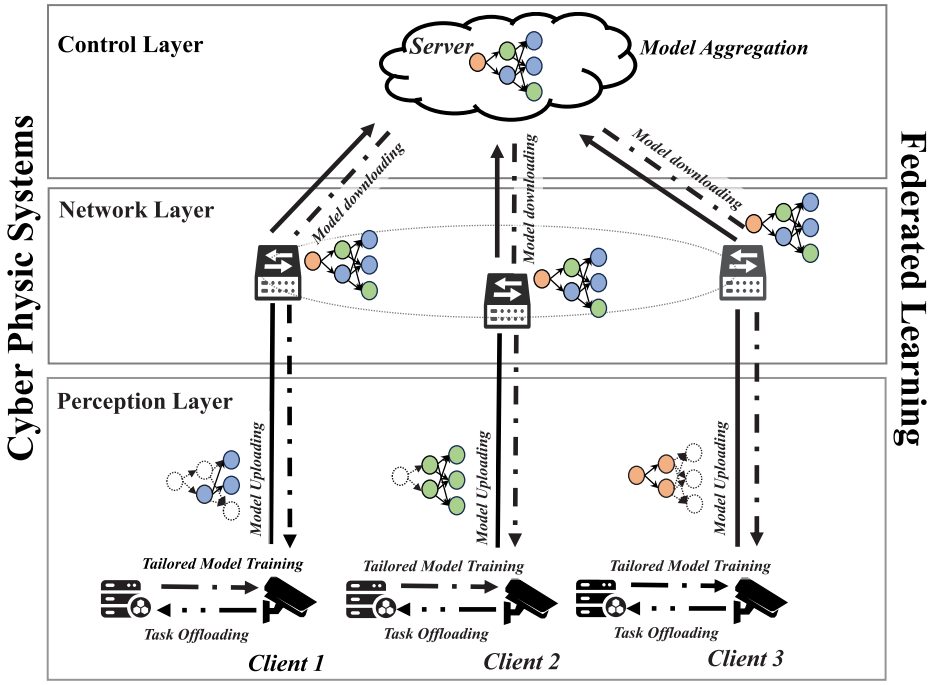


Fig. 1. The architecture combining resources constrained CPS and federated learning.

Table 1
A comparison of current mainstream privacy protection strategies in federated learning.

References	Scenes	Advantages	Limitations
[12]	DP	Quantitative protection of user privacy	Limited user privacy protection
[27]	Cross-silo FL	Ensure user-level privacy	No consideration of data availability
[28]	FL	Adaptively perturb the residual weights using LDP	Adaptive perturbation adds additional cost
[29]	LDP-FL	Reduce communication overhead while protecting privacy	Privacy budget explosion
[30]	FL	Achieves greater utility and smaller transfer rates	Limited user privacy protection
[31]	DP-FL	Protect users' privacy through DP	Privacy budget explosion
[32]	LDP-FL	Improve communication efficiency, reduce privacy budget	Compression model fails in model accuracy
[33]	DP-FL	Solve the privacy budget explosion by model shuffling	Shuffling model adds extra cost
[34]	LDP-FL	Proposed mechanism bypasses the curse of dimensionality	Model accuracy is not considered
[35]	FL	Improve model accuracy, protect client-level DP	Model sparsity cause model accuracy loss
[36]	LDA	Solve the privacy problem in LDA by LDP-FL	Privacy budget is not considered
[37]	Fed-Distillation	Noise-free DP	Supports no privacy preserving ML methods
[38]	DP-FL	Adaptive gradient descent, reduce communications costs	Privacy budget explosion
[39]	FL	Address privacy concerns in wireless IoT	Poor tradeoff between privacy and accuracy
[40]	Signds-FL	Improve model convergence and accuracy	Limited user privacy protection
[41]	Blockchained-FL	Data security and higher accuracy	Privacy budget explosion
[25]	FL	Convergence guarantee, and low communication cost	Poor trade-off between privacy and reliability
[26]	pFL	Model personalization	Poor trade-off between privacy and reliability
[24]	FML	Model personalization	Poor trade-off between privacy and reliability
[42]	LDP-FL	Low communication cost	Poor trade-off between privacy and reliability

different entities to optimize big data recommendations [22]. Ghosh et al. [23] proposed a new iterative federated clustering algorithm, which alternately estimates the clustering identity of users and optimizes the model parameters of the client through gradient descent to improve the efficiency of federated learning communication. Yang et al. proposed the Group-based Federated Meta-Learning framework (G-FML) [24] that employs a simple yet effective grouping mechanism to adaptively partition clients into multiple groups, enabling group-level meta-models to achieve personalization in highly heterogeneous environments. Wang et al. proposed the Communication-Mitigated Federated Learning (CMFL) [25] that provides clients with feedback on the global trend of model updating, reducing communication overhead by avoiding irrelevant updates to the server and ensuring learning convergence. Ma et al. proposed the dubbed Layer-wised Personalized Federated learning (pFedLA) [26] that uses a dedicated hypernetwork for each client on the server side and introduces a parameterized mechanism to update layer-wise aggregation weights for accurate model personalization.

2.2. Model optimization for federated learning with local differential privacy

Seif et al. [43] proposed a private wireless gradient aggregation scheme using LDP and federated learning to enhance privacy protection over wireless channels. Girgis et al. [44] proposed an efficient communication scheme for privacy amplification by client-side subsampling and used a privacy mashup model to reduce the privacy budget. Hu et al. [32] proposed a new federated learning framework with sparse amplification privacy, which combines random sparsization with gradient perturbation to enhance privacy assurance. Liu et al. [33] reduced the privacy budget by exploiting the privacy amplification effect in the recently proposed shuffle model with differential privacy.

Zhou et al. [31] proposed a Gaussian differential privacy-based federated learning algorithm Noisy-FL, which enabled user-level privacy protection. Wang et al. [36] proposed a local differential privacy-based federated learning framework for LDA models and provide theoretical guarantees for data privacy and model accuracy. To address the trade-off between privacy budget and model performance, Sun et al. [37] proposed a new framework that applies the noise-free differential privacy (NFDP) mechanism to the federated model distillation framework. Sun et al. [34] proposed a design for a local differential privacy mechanism for federated learning to address the issue of an exponential increase in privacy budget due to deep model iteration. Jiang et al. [40] proposed a multidimensional selection algorithm based on an exponential mechanism in federated learning to further improve the convergence and accuracy of the model. Jiang et al. [45] proposed a new hybrid differential privacy and adaptive compression for industrial data processing with a federated edge learning framework to address the problem of models subject to inference attacks. In addition, Javed et al. [41] innovatively applied a local differential privacy federated framework using blockchain, providing enhanced data security and higher accuracy. Wang et al. [46] proposed a privacy-preserving federated framework, PPeFL, based on local differential privacy, addressing the rapid increase in privacy budget and significantly reducing communication overhead. Ren et al. [47] introduced EFedDSA, a solution based on horizontal federated learning and differential privacy, aiming to enhance the scalability of machine learning architecture and address privacy concerns by improving local privacy protection strength. Wang et al. [48] presented a federated learning framework based on Edge-IoT to handle vertical and horizontal data, showcasing characteristics of low communication costs and high precision. Truex et al. proposed a federated learning system using local differential privacy (LDP-Fed) [42] that guarantees formal differential privacy for the repeated collection of model training parameters in federated training of large-scale neural networks. It also implements selection and filtering techniques for sharing selected parameter updates with the parameter server.

In conclusion, while these methods can achieve privacy protection for federated learning in CPS, the model's reliability is relatively low. Considering the potential shortage of computational and storage resources in CPS, the reliability of the model trained under such conditions could be further compromised. To address this issue, this paper reduces the training scale of the model through model parameter selection, while also reducing the communication cost of federated learning under the premise of ensuring model accuracy. This approach to privacy protection has a lesser impact on the reliability of the model. Therefore, the proposed method is suitable for resource-constrained CPS.

3. Distributed learning architecture under resource constrained CPS

In this section, we first give the formal statement of the problem encountered in this paper, then give the implementation details of the proposed Fed-MPS scheme.

3.1. Problem statement

The application of CPS-based federated learning to agriculture, and the development of an intelligent agricultural system driven by artificial intelligence and the Internet of Things, holds tremendous promise for addressing numerous challenges faced by the agriculture sector, such as unpredictable weather patterns, soil degradation, and limited access to water resources. Traditional farming methods are often labor-intensive and inefficient, failing to ensure optimal yield and quality. The envisioned intelligent agricultural system will be equipped with various sensors, including those for temperature, humidity, soil moisture, and other environmental factors, embedded in farmland.

Table 2
The list of main symbols.

Symbol	Meanings
D_i	Dataset of client i
B	Local mini-batch size
E	Number of local epochs
η	Learning rate
θ	Model parameters
ϵ	Privacy budget
$\mathbf{u}$	Update direction
$r(\mathbf{u}, \mathbf{u}_s)$	Relevance of local parameters and server update
δ	Failure probability
G	Bounded gradient
g	Gradient
σ	Gaussian distribution variance
$\phi(.)$	L2 sensitivity
L	L-smooth
μ	μ-strongly convex
e^{th}	Error threshold
α	Privacy budget of RDP
β	Failure probability of RDP
d, m	Dimensions of the model parameters
ξ	Dataset of small batch
Ψ	Average optimization rate

These sensors are connected to the internet, providing real-time data to cloud-based platforms. By leveraging efficient artificial intelligence algorithms [49], the system can analyze sensor data to offer farmers advice on optimizing agricultural outcomes, determining the best timing for planting or harvesting crops, managing irrigation amounts, and optimizing fertilizer use [50]. The system's learning capability is ensured through federated learning, continuously collecting farm data to enhance overall performance. Processed data is securely stored in the system's database, accessible to authorized users anytime and anywhere through a network-based dashboard. The recommendation system of the intelligent agricultural platform ensures the appropriate use of resources such as water, fertilizers, and pesticides, minimizing agricultural input waste. In conclusion, the integration of CPS-based federated learning with agriculture through the intelligent agricultural platform offers a sustainable, high-yield, efficient, and cost-effective future for agriculture.

This research has garnered widespread attention from academia and industry. For instance, Kumar et al. [51] proposed a federated learning framework called PEFL for CPS-based agriculture. This framework implements federated learning within the CPS architecture, enhancing the privacy protection capabilities in agriculture. Furthermore, Yu et al. [52] integrated federated learning into the agricultural CPS architecture, improving communication efficiency and achieving higher precision. However, the robustness of such an intelligent agricultural system heavily relies on the resources provided by the underlying CPS. If the resources of the underlying system are constrained, then any module built on that system may struggle to perform well. Addressing this challenge requires a formal statement of the problems encountered in this paper.

Suppose in resource constrained CPS, there are N clients and one server participating in federated learning. At each round, k ($k \leq n$) clients are randomly chosen to perform local model training. Let set $D_i = \{(x_{i,k}, y_{i,k})\}_{k=1}^{n_i}$ represent the private data of client i, where n_i is the corresponding sample number; θ represent the model parameters of client and server; b_i denote the Gaussian noise sampled from $N(0, \sigma^2 I_d)$ for client i. Our main concerns revolve around addressing the challenges of privacy budget explosion, limited computational and storage resources, and ensuring model training convergence. Therefore, the problem can be formally described as finding a solution that satisfies the following equation while minimizing the computation and storage

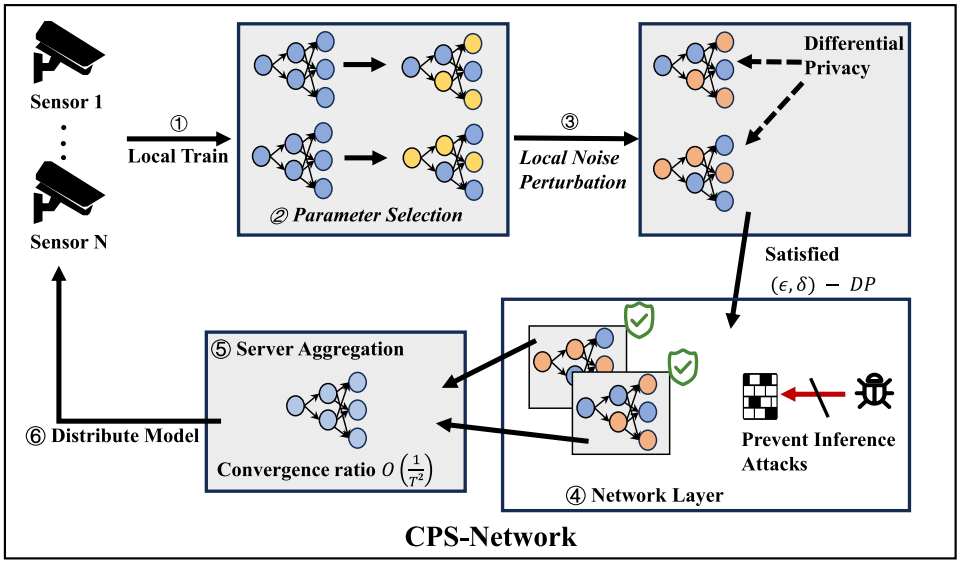


Fig. 2. The flowchart of the proposed Fed-MPS scheme. ① Each client train the local model using data collected by sensors. ② Parameter selection helps to reduce the scale of the trained model about to be uploaded, where the selected parameters are denoted by solid yellow circles. ③ The noise perturbation is implemented to provide (ϵ, δ)-DP protection for each client. ④ The shrinked and perturbed models are aggregated toward the server through the network layer. ⑤ The server aggregates the models with a convergence ratio of $O(\frac{1}{T^2})$ for T rounds of federated learning. ⑥ The aggregated model is distributed to all clients. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

resources.

$$\begin{aligned} &\arg\min_{\theta} \frac{1}{k} \sum_{j=1}^{k} f_j(\theta + b) \\ &st.\ b \sim N(0, \sigma^2 I_d), \\ &f_j(\theta) = \frac{1}{n_i} \sum_{i=1}^{n_i} L(x_i^k, y_i^k; \theta). \end{aligned} \tag{1}$$

Here, $L(.)$ is the loss function of the training error; $\arg\min_{\theta}$ ensures the added noise and the model size are minimized, which means the required privacy budget, computational resources, storage resources, and bandwidth are minimized. The list of main symbols used throughout this paper is summarized in Table 2.

3.2. The proposed scheme implementation

Federated learning in resource-constrained CPS faces two main challenges. Firstly, devices in CPS often lack sufficient resources to complete complex tasks through federated learning, posing a risk of user privacy disclosure during the process. Secondly, while the combination of local differential privacy and federated learning effectively protects users' privacy, it also introduces new challenges. The added noise is proportional to the model size, which is inversely proportional to the model accuracy. Modern deep learning neural networks, which may contain millions of model parameters, can be very large. Consequently, huge deep neural networks might suffer from the performance deterioration in model accuracy caused by "Privacy Budget Explosion".

In addressing the aforementioned problems, Wang et al. [25] found that a client's participation in federated training is somehow related to the alignment of the client and server model update directions. Here, the model update direction refers to the change in model parameters from the original to the updated model when performing an update. If this change is positive, it signifies a positive direction; otherwise, it is considered a negative direction. Additionally, they established a threshold, such that if a client's model update direction aligns with the server's update direction, then a counter is incremented. When the accumulated count surpasses the threshold, the client is permitted to engage in federated training; otherwise, participation is denied. Furthermore, experimental results from [25] demonstrated the effectiveness of determining a client's participation in federated training based on the alignment of the client and server model update directions.

Considering the potential drawback of directly discarding clients in the method proposed by Wang et al. [25], which may result in the loss of model parameters contributing to convergence, we take a different perspective on this issue and attempt to find a solution. While they determined a client's participation in federated training based solely on the model update direction, our solution enhances this approach by refining the retention or discarding of model parameters according to the model update direction. This refined approach offers a more sophisticated consideration, allowing us to maintain model parameters that contribute to convergence, expedite the convergence process of model training, and improve accuracy. Next, we introduce the proposed Fed-MPS scheme in details, which comprises three modules: model parameter selection, model aggregation, and local optimization and noise injection. The flowchart of Fed-MPS scheme is shown in Fig. 2 and Algorithm 1 gives its pseudocode. In Algorithm 1, the server is primarily responsible for initializing the model, collecting the model parameters from the selected k clients, and averaging the collected models. For the selected clients, the local model is trained by performing stochastic gradient descent as shown on Line 5 of Algorithm 1. Line 8 calculates the update direction between the model parameters and the server model. Line 9 calculates the correlation between the model parameters and the server updates. Line 11 indicates the noise added to the final model, which is then uploaded to the server.

Model Parameter Selection. From the above analysis, we know that we only need to compare the model update direction between the clients and server. Thereby, we need to define the update direction first before further analysis. We use the following equation to calculate the update direction

$$\mathbf{u_t} = \frac{\|Update_{t+1} - Update_t\|}{\|Update_t\|}. \tag{2}$$

Here, $Update_{t+1}$ and $Update_t$ denote the updates in the $(t+1)$th round and tth round respectively, and $\|.\|$ denotes the L2 norm throughout this paper. By Eq. (2), we can calculate the update directions of the local

Algorithm 1 The Pseudocode of the Fed-MPS Scheme

Require: the dataset of client i D_i, the number of local client n, the local mini-batch size B, the number of local epochs E, and the learning rate η.
Ensure: the global model θ
LocalUpdate:
receive θ^{t+1} from server;
for each local client $i \in k$ in parallel **do**
  for each local epoch $s = 0, 1...E$ **do**
    for each batch $b \in B$ **do**
      $\theta_i^{t,s+1} \leftarrow \theta_i^{t,s} - \eta g_i^{t,s}$;
    end for
  end for
  calculate the update direction of each model parameter according to Equation (2);
  compute r($\mathbf{u}$,$\mathbf{u_s}$) according to Equation (3);
  according to whether r($\mathbf{u}$, $\mathbf{u_s}$) are 1s, 0s, or 1s and 0s to decide whether the parameters are retained;
  upload θ'^t_i after adding noise $b_i^{t,s} \sim N(0, \sigma^2 I_d)$;
end for
ServerUpdate:
initialize model θ^0;
send θ^0 to all clients;
for each round $t = 1, 2, ...T$ **do**
  randomly select k ($k < n$) local clients;
  collect all models θ^t from selected clients;
  aggregate all models by $\theta^{t+1} \leftarrow \frac{1}{k}\sum_{i=0}^{k-1} \theta'^t_i$ with the '0' padding method;
  Send θ^{t+1} to all client;
end for

and server model parameters. Further, we need to determine whether the update directions of the local and server model parameters are consistent.

To this end, we define the following function $r(.)$ to assess the consistency of the update directions of the local and server model parameters.

Definition 1. Let $\mathbf{u}^i$ denote the update direction of the ith local model, and $\mathbf{u_s}$ denote the last update direction of the server. Thereby, the correlation of model update direction between the client and server can be measured using the following equation:

$$r(\mathbf{u}^i, \mathbf{u_s}) = \mathbb{I}\left(sgn(\mathbf{u}^i) = sgn(\mathbf{u_s})\right). \tag{3}$$

Here, $\mathbb{I}(.)$ represents the vector based indicator function that maps any input into a binary vector under a certain condition.

Then, we explain how to calculate $r(.)$ in details. To be more specific, let $\mathbf{u}^i$ take the form of $\mathbf{u}^i = \left(u_1^i, u_2^i, \ldots, u_m^i\right)$, where u_j^i represents the update of the ith local model in the jth dimension. Then, the sgn function, denoted by $sgn(.)$, transforms every element u_i^j of a m-dimension vector $\mathbf{u}^i$ into 1, 0 or -1 by

$$sgn(u_j^i) = \begin{cases} 1 & u_j^i > 0, \\ 0 & u_j^i = 0, \\ -1 & u_j^i < 0. \end{cases} \tag{4}$$

Following that, let $S_i = \sum_{j=1}^{m} sgn(u_j^i)$, and the process of converting the n-dimensional parameters to 1, 0, or -1 using the above sgn function with a pre-determined threshold $thres$ can expressed by

$$sgn(\mathbf{u}^i) = \begin{cases} 1 & S_i > thres, \\ 0 & S_i = thres, \\ -1 & S_i < thres. \end{cases} \tag{5}$$

By observing Eqs. (3), (4) and (5), we know that $r(\mathbf{u}, \mathbf{u_s})$ takes the form of a n-dimension vector that consists of 1s and 0s. Furthermore, let $\mathbf{u}^{i,t+1}$ and $\mathbf{u}_\mathbf{s}^t$ denote the update of the kth local model's parameters at the $(t+1)$th round and that of the server at the tth round, respectively. According to three possibilities of the update direction consistency, we summarize the value of $r(\mathbf{u}^i, \mathbf{u_s})$ by

$$r\left(\mathbf{u}^{i,t+1}, \mathbf{u}_\mathbf{s}^t\right) = \mathbb{I}\left(sgn(\mathbf{u}^{i,t+1}) = sgn(\mathbf{u}_\mathbf{s}^t)\right) = \begin{cases} \mathbf{1}^{1\times m} & Consistency, \\ \mathbf{0}^{1\times m} & Inconsistency, \\ \mathbf{r}^{1\times m},\ r \in \{0,1\} & Partially\ Consistency. \end{cases} \tag{6}$$

Here, $\mathbf{1}^{1\times m}$, $\mathbf{0}^{1\times m}$ and $\mathbf{r}^{1\times m}$ denote the vector that consists of n 1s, 0s and rs, respectively.

According to Eq. (6), we can make the appropriate decisions on model parameter selection as follows. For the consistency case, all parameters of the ith model should be uploaded; for the inconsistency case, the ith model should be discarded; for the partially consistency case, only those parameters of the ith model with the same model update direction as that of the server will be uploaded.

Overall, during the tth round of local training, the client first calculates the update direction of local and server model parameters using Eq. (2), to determine the value of $sgn(.)$ for model parameter i by Eqs. (3), (4) and (5). Next, it assesses the consistency between each local and server model parameter i using Eq. (6) to retain the parameters with consistent directions and discard those with inconsistent directions.

Model Aggregation. First, the server randomly initializes the model weights at the beginning with the total number of n local clients. Then, in the rth round of communication, the server randomly selects k ($k < n$) clients for local training. The individual selection of model parameters by each local client results in heterogeneous models of the clients.

As shown in Fig. 2, the model parameters marked by yellow solid circles in step 2 are the selected ones for training, maintaining distinct parameters at their respective model positions. For the server-side aggregation, we adopt the '0' padding method. To be specific, for positions in the model lacking parameters, we default the values at those model positions to zero and aggregate the values by averaging them with other model parameters. We aggregate the model in this manner and send the aggregated model to each client.

Local Optimization and Noise Injection. For each client, it uses its own private dataset for local model training. After the local training is completed, the update direction of the local model is calculated, so as the correlation between the update direction of the local model and that of the server. Only selected model parameters will be sent to the server for model aggregation. Before uploading the model, a certain amount of Gaussian noise b, $b \sim N(0, \sigma^2 I_d)$, will be added to the model to provide privacy protection. The rigorous theoretical analysis on the proposed scheme Fed-MPS is provided in Section 4, where Theorem 1 proves that Fed-MPS satisfies $(\epsilon, \delta) - DP$, while Theorem 2 guarantees that Fed-MPS converges to the global optimum.

4. Theoretical analysis

In this section, we conduct comprehensive privacy analysis and convergence analysis on the proposed Fed-MPS scheme. We prove that the proposed Fed-MPS scheme achieves $(\epsilon, \delta) - DP$, where $\epsilon = \frac{7q^2 I_i \tau \alpha O G^2}{B^2 \sigma^2} + \frac{log(1/\delta)}{\alpha - 1}$ denotes the privacy budget, while the convergence analysis indicates that this scheme can converge to the global optimum with a convergence ratio of $O\left(\frac{1}{T^2}\right)$ within T rounds of federated learning. We also analyze the communication overhead of the proposed scheme.

4.1. Privacy analysis

We first discuss the end-to-end privacy guarantees, which refers to the privacy guarantee from client to server and it should be satisfied by each client. The assumptions and lemmas required for the privacy analysis are given in the following.

Assumption 1 (*Bounded Gradient*). The loss function $l_{(i)}(x, z)$ has $G/\sqrt{d}$-bounded gradients, i.e., for any data sample z from D_i, we have $\left|\left[\nabla l_i(x, z)\right]_j\right| \leq G/\sqrt{d}$ for all $x \in R^d$, $j \in [d]$ and $i \in [n]$.

Lemma 1 (*RDP Composition [53]*). *If M_1 satisfies $(\alpha, \rho_1) - RDP$ and M_2 satisfies $(\alpha, \rho_2) - RDP$, then their composition $M_1 \circ M_2$ satisfies $\left(\alpha, \rho_1 + \rho_2\right) - RDP$.*

Lemma 2 (*Gaussian Mechanism [53]*). *Let $h : D \to R^d$ be a vector-valued function over datasets. The Gaussian mechanism $M = h(D) + b$ with $b \sim N(0, \sigma^2 I_d)$ satisfies $\left(\alpha, \alpha\phi^2(h)/2\sigma^2\right) - RDP$, where $\phi(h)$ is the L_2 sensitivity of h defined by $\phi(h) = sup_{D,D'}\left|\left|h(D) - h(D')\right|\right|$ with D, D' being two neighboring datasets in D.*

Lemma 3 (*RDP to DP Conversion [54]*). *If M satisfies $(\alpha, \rho) - RDP$, then it also satisfies $\left(\rho + \frac{log(1/\delta)}{\alpha - 1}, \delta\right) - DP$.*

For a Gaussian mechanism M and any m-datapoints dataset D, we define $M \circ SUBSAMPLE$ as applying M on the subsampled dataset as input, where B datapoints are subsampled without replacement from the dataset with $q = B/m$ as the sampling ratio. By this definition, we introduce Lemma 4 that ensures the RDP for subsampling.

Lemma 4 (*RDP for Subsampling Mechanism [54,55]*). *If M satisfies $(\alpha, \rho(\alpha)) - RDP$ with respect to the subsampled dataset for all integers $\alpha \geq 2$, then the new randomized mechanism $M \circ SUBSAMPLE$ satisfies $(\alpha, \rho'(\alpha)) - RDP$ with respect to D, where*

$$\rho'(\alpha) \leq \frac{1}{\alpha - 1} log\left(1 + q^2\binom{\alpha}{2} \min\left\{4(e^{\rho(2)} - 1)\right\} + \sum_{j=3}^{\alpha} q^2\binom{\alpha}{j} 2e^{(j-1)\rho(j)}\right). \tag{7}$$

If $\sigma'^2 = \sigma^2/\phi^2(h)$ and $\alpha \leq (2/3)\sigma^2 log\left(1/q\alpha(1 + \sigma'^2)\right) + 1$, then $M \circ SUBSAMPLE$ satisfies $\left(\alpha, 3.5q^2\phi^2(h)\alpha/\sigma^2\right) - RDP$.

Let $q = B/m$ denote the data sampling rate, τ denote the number of local iterations, I_i represent the number of rounds agent i participated, α denote the privacy budget of RDP, σ^2 denote the variance of Gaussian distribution, δ represent the failure probability in differential privacy, $o_{i,t}$ denote the optimization rate of agent i at tth round iteration, and O denote the average optimization rate over T federated learning rounds. We then introduce Theorem 1 that guarantees the privacy of the proposed Fed-MPS scheme.

Theorem 1 (*Privacy Guarantee*). *Assume that at each iteration small batches $\xi_i^{t,s}$ are sampled without replacement. Under Assumption 1, if $\sigma'^2 = \sigma^2 B^2/2OG^2 \geq 0.7$, then Fed-MPS achieves $(\epsilon, \delta) - DP$ for agent i, where*

$$\epsilon = \frac{7q^2 I_i \tau \alpha O G^2}{B^2\sigma^2} + \frac{log(1/\delta)}{\alpha - 1} \tag{8}$$

for any $\alpha \leq (2/3)\sigma^2 log\left(1/q\alpha(1 + \sigma'^2)\right) + 1$ and $\delta \in (0, 1)$.

Proof. In our differential privacy mechanism for Fed-MPS, the privacy guarantee provided by Gaussian noise is amplified by the model parameter selection. To analyze the end-to-end privacy, we need to analyze the sensitivity in Algorithm 1 at line 14.

Let $g_i^{t,s}$ and $b_i^{t,s}$ denote the model gradient and the added Gaussian noise of agent i after the tth round of compression, respectively. Then, we analyze the sensitivity of $g_i^{t,s}$ and then calculate the privacy guarantee after adding noise $b_i^{t,s}$ for agent i, given that any two neighboring data sets $\xi_i^{t,s}$ and $\xi'^{t,s}_i$ have the same size B but differ in one data sample (e.g., $z \in \xi_i^{t,s}$ and $z' \in \xi'^{t,s}_i$). Since in Fed-MPS, the model parameters are passed through a model parameter selection algorithm to obtain the final model parameters, the optimization rate of each agent is different at different iteration rounds. Thus, the $L2$ sensitivity can be expressed as follows under Assumption 1 with $2o_{i,t}G^2/B^2$ as a bound

$$\begin{aligned}\phi_{i,t}^2 &= \max\left\|g_i^{t,s} - \left[\nabla f_i(\theta_i^{t,s}, \epsilon_i^{t,s})\right]\right\|^2 \\ &= \max\left\|(1/B)\left[\nabla l(\theta_i^{t,s}, z) - \nabla l(\theta_i^{t,s}, z')\right]\right\|^2.\end{aligned} \tag{9}$$

We find that the sensitivity of $g_i^{t,s}$ is proportional to the optimization rate, which reduces the privacy loss according to Lemma 2. In each local iteration of Algorithm 1, a small batch $\xi_i^{t,s}$ of subsampling satisfies $(\alpha, \alpha) - RDP$, $\alpha = o_{i,t}G^2/B^2\sigma^2$. Furthermore, Lemma 4 guarantees that by subsampling $M \circ$ SUBSAMPLE satisfies $\left(\alpha, 3.5q^2\phi^2(h)\alpha/\sigma^2\right) - RDP$. Thus, we derive Theorem 1 according to Lemmas 1 to 4.

From Theorem 1, it can be deduced that given a fixed value of δ, ϵ is computed numerically by searching an optimal α that minimizes ϵ. We notice that the noise size σ is proportional to O. This implies that the size of Gaussian noise can be reduced when the compression ratio is less than 1, thus improving the model accuracy.

4.2. Convergence analysis

We then give the convergence analysis, and calculate the communication overhead as well. For the convenience of illustration, we assume that the size of the local data of all clients is the same, for any $i, j \in K$, $D_i = D_j$, and N clients are selected to train locally for E rounds each time. Similarly, the assumptions to support the proof of our convergence analysis are given as follows.

Assumption 2. L-smooth: $\forall x, y, F(y) \leq F(x) + (y - x)^T \nabla F(x) + \frac{L}{2}\left||y - x|\right|^2$.

Assumption 3. μ-strongly convex: $\forall x, y, F(y) \geq F(x) + (y - x)^T \nabla F(x) + \frac{\mu}{2}\left||y - x|\right|^2$.

Assumption 4. Bounded gradient and bounded variance of gradient: $E\left[\left||\nabla F\left(x^k[t], \xi^k[t]\right) - \nabla F\left(x^k[t]\right)|\right|^2\right] \leq \sigma^2$ and $E\left[\left||\nabla F\left(x^k[t], \xi^k[t]\right)|\right|\right] \leq G^2$.

Theorem 2 (*Convergence Analysis*). *When the above three assumptions hold, let $K = L/\mu$, $\gamma = max\{8K, E\}$, and the learning rate $\eta = \frac{2}{\mu(\gamma + t)}$, where L is the L-smooth, μ is μ-strongly convex, E is the number of local training rounds, and t is the number of global training rounds. If the error threshold in Fed-MPS satisfies the following inequality:*

$$e^{th}[t] \leq \eta_t^2 = \frac{4}{\mu^2(\gamma + t)^2} \sim O(\frac{1}{t^2}),\ E\left[e^i\right] = 0, \quad \forall i \in K. \tag{10}$$

Then, all local clients participating in training in Fed-MPS satisfy:

$$E\left[F\left(\overline{x}[T]\right) - F\left(x^*\right)\right] \leq \frac{K}{\gamma + T - 1}\left(\frac{2B}{\mu} + \frac{\mu\gamma}{2}\left[\left||x[1] - x^*|\right|^2\right]\right). \tag{11}$$

where $B = \sum_{i=1}^{N} \frac{\sigma_i^2}{N^2} + 6L\Gamma + 8(E - 1)G^2 + E\left[\left||e^{th}|\right|^2\right]$.

Proof. By the $L - smooth$ assumption, we can get the following inequality:

$$E\left[F\left(\overline{x}[t]\right)\right] - F(x^*) \leq \frac{L}{2}E\left[\left||\overline{x}[t] - x^*|\right|^2\right]. \tag{12}$$

In Fed-MPS, the uplink errors from different users are not independent. To solve this problem, we constrain the error term of the uplink as follows:

$$E\left[\left|\left|x^*[t]-\overline{x}[t]\right|\right|^2\right]=E\left[\left|\left|e[t]\ \right|\right|^2\right]=\frac{1}{N^2}E\left[\left|\left|\sum_{i=1}^{N}e^i[t]\right|\right|^2\right]\le e^{th}\left[t\right]. \quad (13)$$

We can also obtain the following inequality:

$$E\left[\ \left|\left|\overline{x}[t+1]-x^*\right|\right|^2\right]\le(1-\eta_t\mu)E\left[\left|\left|\overline{x}[t]-x^*\right|\right|^2\right]+e^{th}\left[t\right] + \eta_t^2\left[\sum_{i=1}^{N}\frac{\sigma_i^2}{N^2}+6L\Gamma+8(E-1)G^2\right]. \quad (14)$$

Let $\Delta_t = E\left[\left|\left|\overline{x}[t+1]-x^*\right|\right|^2\right]$, if the error threshold of Fed-MPS satisfies $e^{th}[t]\le\eta_t^2$, then we obtain

$$\Delta_{t+1}\le(1-\eta_t\mu)\Delta_t+\eta_t^2B. \quad (15)$$

where $B=\sum_{i=1}^{N}\frac{\sigma_i^2}{N^2}+6L\Gamma+8(E-1)G^2+E\left[\|e^{th}\|^2\right]$.

Given the learning rate $\eta_t=\frac{\beta}{t+\gamma}$, where $\beta\ge\frac{1}{\mu}$, $\gamma\ge0$, $\eta_1\le min\{1/\mu,1/4L\}=1/4L$, $\eta_t\le2\eta_{t+E}$, there is $\Delta_t\le\frac{\upsilon}{\gamma+t}$ such that the following inequality holds

$$\Delta_{t+1}\le\left(1-\eta_t\mu\right)\Delta_t+\eta_t^2B=\left(1-\frac{\beta\mu}{t+\gamma}\right)\frac{\upsilon}{t+\gamma}+\frac{\beta^2B}{(t+\gamma)^2} \le\frac{\upsilon}{t+\gamma+1}. \quad (16)$$

By substituting Δ_t into the above inequality and let $t=T$, we then prove Theorem 2.

Theorem 2 guarantees the convergence of the proposed Fed-MPS scheme. Following that, the communication overhead can be calculated as follows. Specifically, we give the total amount of communication data for the federated average algorithm by

$$C_{(FedAvg)}=T\left|\theta\right|. \quad (17)$$

Let ψ_i denote the optimization rate in round i. Thus, the total amount of communication data for the proposed Fed-MPS can be calculated by

$$C_{(Fed-MPS)}=\sum_{i=1}^{T}\left|\theta_i\right|=\sum_{i=1}^{T}\frac{1}{\psi_i}\left|\theta\right|. \quad (18)$$

Let Ψ denote the average optimization rate over the entire training process. Thus, we obtain $\sum_{i=1}^{T}\frac{1}{\psi_i}\left|\theta\right|=\frac{T}{\Psi}\left|\theta\right|$, then Theorem 3 is derived.

Theorem 3 (*Communication Overhead*). *Assume that the number of federated learning global training rounds is T. The compression rate of each round is different because the local client optimize the model by parameter selected algorithm individually. Let the optimization rate of the local model in round i as $\psi_i=\frac{|\theta|}{|\theta_i|}$, where θ is the model before optimization and θ_i is the model after compression. The communication overhead of Fed-MPS is $o(\frac{T}{\Psi}\left|\theta\right|)$ for any agent i.*

Since the communication cost of Fed-MPS is $o(\frac{T}{\Psi}\left|\theta\right|)$ for any agent i, we can deduce that its computational complexity is inversely proportional to Ψ. The larger the value of Ψ, the smaller its computational complexity. The communication overhead for the baseline solution is $o(\frac{T}{|\theta|})$, which means that Fed-MPS's communication overhead is only $\frac{1}{\Psi}$ of other methods, i.e., pFEDLA [26] and G-FML [24].

5. Performance evaluation

We assess the performance of our Fed-MPS scheme in comparison to the baseline algorithms CMFL [25], LDP-Fed [42], pFEDLA [26], and G-FML [24] using benchmark datasets Mnist, Cifar10, and FashionMNIST. The experiments were conducted on a computer running Windows 10, equipped with a 12th generation Core i7 processor capable of reaching speeds up to 4.70 GHz, and an RTX3060 GPU. Our Fed-MPS scheme was implemented in Python, and the experimental setup is detailed as follows.

5.1. Experiment setup

Baselines. CMFL [25] provides clients with feedback information regarding the global trend of model updating. Each client checks whether its update aligns with this global trend and is relevant enough for model improvement. By avoiding uploading irrelevant updates to the server, CMFL can substantially reduce communication overhead while still guaranteeing learning convergence. LDP-Fed [42] provides a formal guarantee of differential privacy for the repeated collection of model training parameters in the federated training of large-scale neural networks over multiple individual participants' private datasets. Additionally, LDP-Fed implements a suite of selection and filtering techniques for perturbing and sharing selected parameter updates with the parameter server. pFEDLA [26] uses a dedicated hypernetwork for each client on the server side, which is trained to identify the mutual contribution factors at the layer level. At the same time, a parameterized mechanism is introduced to update the layer-wise aggregation weights, gradually exploiting the inter-user similarity and achieving accurate model personalization. G-FML [24] employs a simple yet effective grouping mechanism to adaptively partition the clients into multiple groups. This mechanism ensures that each group is formed by the clients with similar data distribution, enabling the group-wise meta-model to achieve personalization. Thereby, it can be generalized to a highly heterogeneous environment.

Datasets. The experiments were conducted using the FashionMNIST, Mnist, and Cifar10 benchmark datasets. The FashionMNIST dataset includes images from 10 categories, with a total of 60,000 samples in the training dataset and 10,000 samples in the test dataset. The Cifar10 dataset comprises images from 10 categories, totaling 50,000 images and corresponding labels. The Mnist dataset consists of handwritten digital images, with 60,000 images and labels in the training set and 10,000 images and labels in the test set.

Local Models. For the FashionMNIST dataset, we utilized a CNN model comprising two $5*5$ convolutional layers. The first convolutional layer consists of 32 filters, followed by a second convolutional layer with 32 filters. Each convolutional layer is succeeded by a $2*2$ pooling layer and a Relu activation function. Additionally, a dropout layer was incorporated to prevent overfitting. The model also includes a hidden layer with a 1024-dimensional input and a 512-dimensional output, as well as an output layer with a 512-dimensional input and a 10-dimensional output. For the Mnist dataset, we employed a CNN model with two $5*5$ convolutional layers. The first convolutional layer is equipped with 10 filters, followed by a second convolutional layer with 10 filters. Similar to the FashionMNIST model, each convolutional layer is followed by a $2*2$ pooling layer and a Relu activation function. A dropout layer was also included to prevent overfitting. Two fully-connected layers were incorporated, with the first having a 320-dimensional input and a 50-dimensional output, and the second having a 50-dimensional input and a 10-dimensional output. For the Cifar10 dataset, we also utilized a CNN model with two $5*5$ convolutional layers. The first convolutional layer contains 6 filters, followed by a second convolutional layer with 16 filters. Each of these convolutional layers is succeeded by a $2*2$ pooling layer and a Relu activation function. The model includes three fully connected layers, with the first having $256*2*2$ dimensional inputs and 128 dimensional output, the second having 128-dimensional input and 256-dimensional output, and the last having 256-dimensional input and 10-dimensional output.

Training Parameters Configuration. In the experiment, there are 100 agent clients, with 10 out of 100 agents randomly selected in each training round to conduct local training on the FashionMNIST,

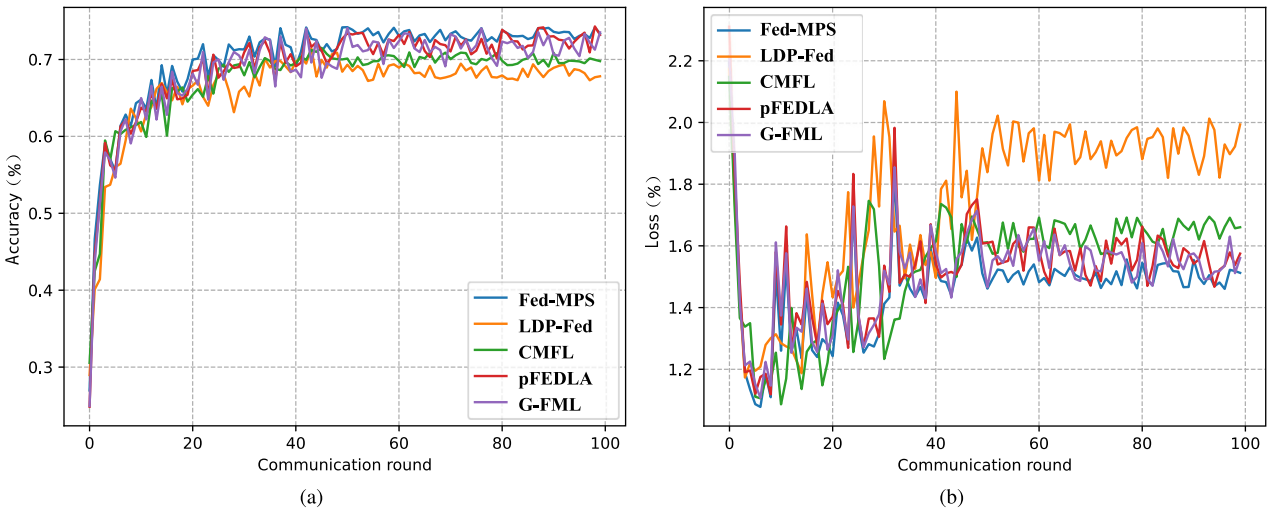


Fig. 3. Training (a) accuracy and (b) loss of Fed-MPS and baselines on FashionMNIST in 100 rounds.

Mnist, and Cifar10 datasets. For the FashionMNIST dataset, each client agent is allocated a training sample consisting of 600 samples and a test sample containing 100 samples. Regarding the Mnist dataset, each client agent is assigned a training set of 600 images and labels, along with a test set of 100 images and labels. As for the Cifar10 dataset, each client agent is provided with a training set containing 500 images and labels, as well as a test set containing 100 images and labels. All approaches ensure local differential privacy by incorporating Gaussian noise, with the parameters ϵ and δ initially set to 0.6 and 1e−3, respectively, for the model accuracy test. Subsequently, different privacy budgets will be applied for the subsequent model accuracy tests. It is important to note that the experiment is repeated 10 times on all datasets, and the experimental results are presented based on the median values. In each experiment, the local training is executed in 50 rounds, while the global training is conducted over 100 rounds. The performance metrics mentioned above are taken into consideration for the effectiveness of experimental comparisons.

5.2. Experiment result

Model Accuracy under a Fixed Privacy Budget. We first conduct experiments to evaluate the performance of the proposed scheme under a fixed privacy budget ϵ. We let $\epsilon = 0.6$, and the experimental results are shown in Figs. 3, 4, and 5.

Fig. 3 illustrates the accuracy and loss of the Fed-MPS, CMFL, LDP-Fed, pFEDLA, and G-FML schemes during 100 rounds of training on the FashionMNIST benchmark dataset. In terms of accuracy, our scheme achieves convergence after approximately 50 training rounds, maintaining relatively stable accuracy thereafter. We observe that our Fed-MPS scheme consistently outperforms CMFL and LDP-FED after the 30th round of training, with our scheme being 4% and 6% more accurate than CMFL and LDP-FED, respectively, after 50 rounds of training. While our scheme exhibits slightly lower accuracy than the pFEDLA scheme and higher accuracy than the G-FML scheme at the 30th round, after 50 rounds, our scheme's accuracy is approximately 4% higher than pFEDLA and 3% higher than G-FML. Clearly, our scheme demonstrates superior accuracy performance on the Fashion-MNIST dataset compared to other baselines. Regarding training loss, we observe that all methods generally converge after approximately 50 training rounds, with the training loss remaining within a certain range thereafter. However, the training loss of the Fed-MPS scheme is consistently lower than the other two compared schemes throughout the 100 rounds of federated training, and the loss of the CMFL scheme is also smaller than the training loss of the LDP-Fed scheme. Furthermore, our scheme exhibits better performance in training loss on the FashionMNIST dataset compared to the pFEDLA and G-FML schemes, as expected. This is attributed to the parameter selection module, which retains only model parameters with the same update directions as the global model.

Fig. 4 showcases the accuracy and training loss of the Fed-MPS, CMFL, LDP-Fed, pFEDLA, and G-FML schemes over 100 rounds of training on the CIFAR10 benchmark dataset. In terms of accuracy, it is evident that the Fed-MPS scheme consistently outperforms the LDP-Fed scheme in accuracy throughout the training process. Additionally, our scheme displays fluctuations compared to CMFL and initially exhibits slightly lower accuracy than CMFL until the 30th round. However, after the 30th round, our scheme surpasses CMFL in accuracy, outperforming CMFL and LDP-Fed by 5% and 7%, respectively. After 50 rounds of training, our scheme achieves an improvement of approximately 1% in accuracy over pFEDLA and 2% over G-FML, indicating superior accuracy performance on the CIFAR10 dataset compared to the baseline schemes. Although our scheme initially exhibits slightly lower accuracy compared to other methods in the first 20 rounds, it surpasses them in accuracy after 20 rounds. Notably, the maximum accuracy of 42.6% is achieved in the 48th round of training. Regarding training loss, the training loss of the Fed-MPS scheme is marginally lower than that of CMFL after 18 rounds, while the training loss of the LDP-Fed scheme is slightly higher than that of CMFL and Fed-MPS until 25 rounds, and slightly lower than that of CMFL and Fed-MPS after 25 rounds. Furthermore, the performance of Fed-MPS in training loss also surpasses the state-of-the-art federated algorithms pFEDLA and G-FML. While the training loss of G-FML fluctuates the most, our scheme maintains relatively stable training loss throughout the entire training process, consistently outperforming other baseline methods. This underscores the effectiveness of our scheme in training on the CIFAR10 dataset, stemming from the model selection that reduces the model size and ultimately improves accuracy, demonstrating superior training outcomes compared to baseline approaches.

Fig. 5 demonstrates the accuracy and loss of the Fed-MPS, CMFL, LDP-Fed, pFEDLA, and G-FML schemes over 100 rounds of training on the MNIST benchmark dataset. In terms of accuracy, our scheme achieves convergence after approximately 50 training rounds, with the accuracy remaining relatively stable thereafter. Additionally, the maximum accuracy of 89.8% is attained in the 49th round of training. Our proposed Fed-MPS scheme exhibits higher accuracy compared to the LDP-Fed scheme after the 10th round of training. While there are fluctuations between our scheme and CMFL before the 35th round, our scheme's accuracy surpasses CMFL's after the 35th round. After 50 rounds of training, our scheme's accuracy outperforms CMFL and LDP-Fed by 5% and 8%, respectively. Furthermore, our scheme achieves

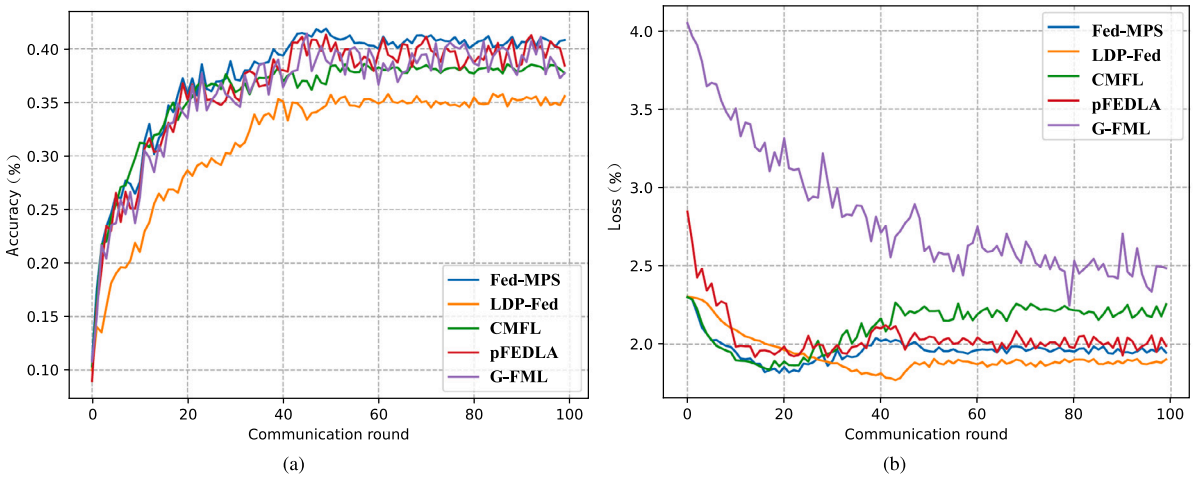


Fig. 4. Training (a) accuracy and (b) loss of Fed-MPS and baselines on CIFAR10 in 100 rounds.

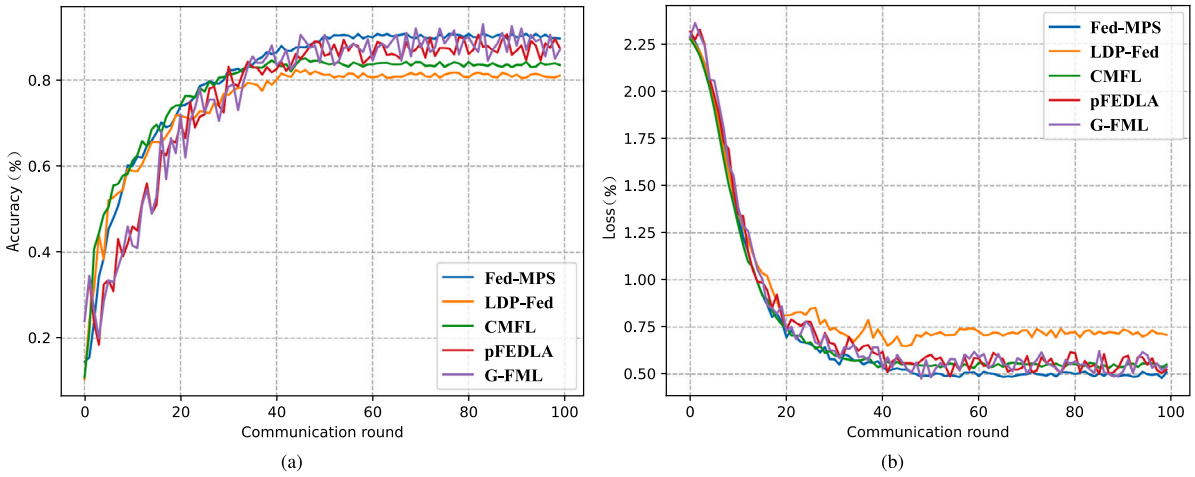


Fig. 5. Training (a) accuracy and (b) loss of Fed-MPS and baselines on MNIST in 100 rounds.

higher accuracy on the MNIST dataset compared to the latest federated algorithms pFEDLA and G-FML. In the 50th round of training, our scheme achieves a accuracy that is 3% higher than G-FML and 2% higher than pFEDLA. Although our scheme initially exhibits lower accuracy than CMFL and LDP-Fed in the first 20 rounds, it consistently outperforms other baseline methods in accuracy after the initial 20 rounds. The model parameter selection in our scheme undoubtedly facilitates the training process and improves accuracy. Regarding training loss, we observe that all approaches generally converge after around 50 training rounds, with the training loss remaining within a certain range thereafter. The training loss of the Fed-MPS scheme is similar to that of CMFL throughout the 50 rounds of federated training and is smaller than that of the LDP-Fed scheme. Our scheme exhibits lower training losses on the MNIST dataset compared to both pFEDLA and G-FML. Furthermore, throughout the entire training process, our scheme demonstrates greater stability with less training loss compared to G-FML and pFEDLA. As a result, our Fed-MPS scheme showcases superior performance in loss and accuracy on the MNIST dataset compared to other baseline methods, attributed to the model selection module of our scheme.

Table 3 compares the accuracy of the Fed-MPS, CMFL, LDP-Fed, pFEDLA, and G-FML schemes on the FashionMNIST, MNIST, and CIFAR10 datasets over 30, 50, and 100 rounds of training, while Table 4 presents the training loss comparisons under the same settings. Evidently, the accuracy of the Fed-MPS scheme surpasses that of the baselines on all three datasets, while the training loss of Fed-MPS is marginally lower. Figs. 6(a) and 6(b) depict histograms with variance of accuracy and loss of the Fed-MPS, CMFL, LDP-Fed, pFEDLA, and G-FML schemes on the FashionMNIST, MNIST, and CIFAR10 datasets after 100 rounds of training. As anticipated, the proposed Fed-MPS outperforms all baseline schemes in both accuracy and loss.

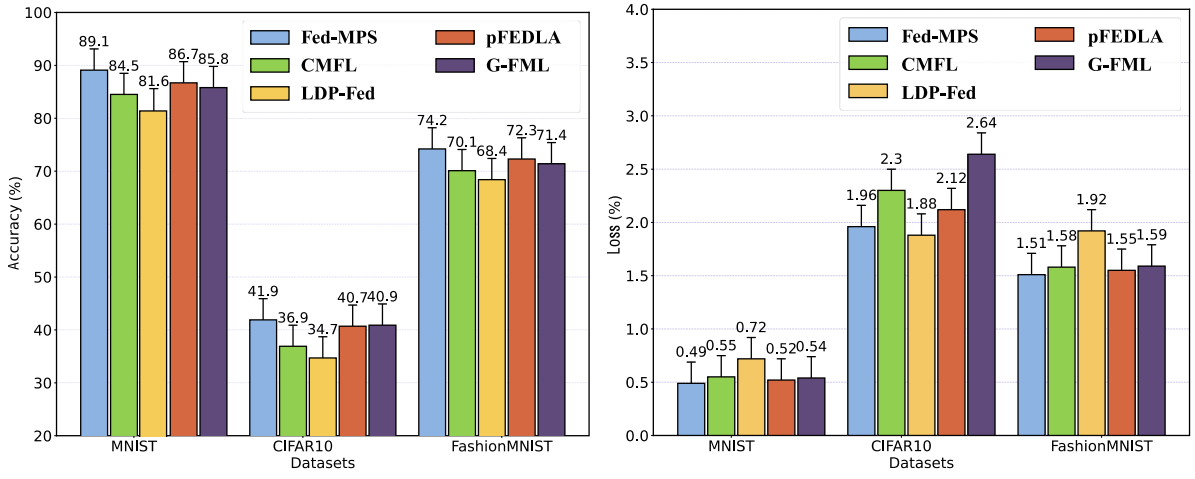
Model Accuracy under Different Privacy Budgets. We conducted experiments with varying privacy budget values from $[0.2, 1.0]$ to compare the performance of Fed-MPS, LDP-Fed, and pFEDLA on the MNIST and CIFAR10 datasets. The experimental results are presented in Figs. 7(a) and 7(b). It is evident that our Fed-MPS scheme outperforms LDP-Fed and pFEDLA under different privacy budgets on both datasets. The performance of pFEDLA is superior to LDP-Fed, as LDP-Fed relies on local differential privacy to protect privacy without considering the impact of differential privacy noise on model accuracy, resulting in lower accuracy due to the addition of excessive noise. In contrast, our Fed-MPS scheme, through the model selection algorithm, selects parameters conducive to accuracy, ultimately reducing the model size. Consequently, less noise is added under the same privacy strength, thereby improving the accuracy. This explains why our scheme outperforms LDP-Fed and pFEDLA in accuracy under different privacy budgets. It is noticeable from the figures that as the privacy budget increases, the accuracy of the schemes improves. This is because a smaller privacy budget (more added noise) leads to lower accuracy. The experimental results depicted in Figs. 7(a) and 7(b) indicate that

Table 3
Comparison of accuracy (%) between Fed-MPS and baselines in the 30th, 50th, and 100th rounds.

Dataset	Round	Fed-MPS	CMFL	LDP-Fed	pFEDLA	G-FML
FashionMNIST	30th	71.3 ± 1.23	69.3 ± 1.34	65.8 ± 1.18	**71.6** ± 0.84	69.7 ± 1.19
	50th	**74.2** ± 0.34	70.0 ± 0.47	68.4 ± 0.29	72.3 ± 0.74	71.2 ± 0.17
	100th	**74.5** ± 0.23	70.2 ± 0.24	68.1 ± 0.53	72.2 ± 0.54	71.4 ± 0.47
MNIST	30th	79.0 ± 0.86	**80.7** ± 1.35	76.9 ± 0.91	76.5 ± 1.21	75.3 ± 1.37
	50th	**89.1** ± 0.54	84.5 ± 0.64	81.6 ± 0.48	86.8 ± 0.29	85.7 ± 0.78
	100th	**89.7** ± 0.41	84.2 ± 0.67	81.3 ± 0.31	86.6 ± 0.62	85.8 ± 0.26
CIFAR10	30th	36.8 ± 0.71	36.8 ± 0.86	30.2 ± 1.11	**36.9** ± 0.96	35.7 ± 0.92
	50th	**41.9** ± 0.41	36.9 ± 0.48	34.7 ± 0.25	40.7 ± 0.78	40.3 ± 0.36
	100th	**42.2** ± 0.39	37.2 ± 0.27	35.0 ± 0.30	41.1 ± 0.47	40.9 ± 0.35

Table 4
Comparison of training loss (%) between Fed-MPS and baselines in the 30th, 50th, and 100th rounds.

Dataset	Round	Fed-MPS	CMFL	LDP-Fed	pFEDLA	G-FML
FashionMNIST	30th	**1.31** ± 0.095	1.53 ± 0.153	1.73 ± 0.147	1.60 ± 0.161	1.57 ± 0.171
	50th	**1.51** ± 0.024	1.58 ± 0.031	1.92 ± 0.022	1.55 ± 0.026	1.59 ± 0.024
	100th	**1.53** ± 0.013	1.63 ± 0.011	1.95 ± 0.027	1.57 ± 0.032	1.56 ± 0.013
MNIST	30th	0.62 ± 0.053	0.62 ± 0.067	0.74 ± 0.073	0.63 ± 0.062	**0.61** ± 0.059
	50th	**0.49** ± 0.019	0.55 ± 0.023	0.72 ± 0.018	0.52 ± 0.017	0.54 ± 0.021
	100th	**0.50** ± 0.013	0.57 ± 0.016	0.70 ± 0.014	0.54 ± 0.023	0.55 ± 0.032
CIFAR10	30th	**1.86** ± 0.134	1.94 ± 0.121	1.89 ± 0.135	1.98 ± 0.119	2.81 ± 0.145
	50th	1.96 ± 0.024	2.30 ± 0.023	**1.88** ± 0.022	2.12 ± 0.015	2.64 ± 0.056
	100th	1.95 ± 0.021	2.27 ± 0.019	**1.85** ± 0.017	2.10 ± 0.026	2.58 ± 0.032



(a) (b)

Fig. 6. Histogram with variance of accuracy and loss of Fed-MPS and baselines.

our Fed-MPS scheme achieves a smaller privacy budget compared to LDP-Fed and pFEDLA, demonstrating the effectiveness of Fed-MPS in reducing the privacy budget.

Results Analysis. The experimental results above demonstrate that the proposed Fed-MPS scheme, whether with a fixed or non-fixed privacy budget, outperforms the baseline strategies CMFL, LDP-Fed, pFEDLA, and G-FML in terms of accuracy and loss across various datasets. This is attributed to the model parameter selection algorithm based on update direction consistency that we have adopted, along with its corresponding model aggregation algorithm. This also fully illustrates that reducing the model scale to the parameter level can ensure model accuracy while reducing the training and uploading of model parameters, thereby addressing the issue of model training under resource-constrained conditions in CPS.

6. Conclusions

In this paper, we propose a Fed-MPS scheme that utilizes local differential privacy to address potential constraints in computational and storage resources in CPS. This scheme, with its distributed learning and reduced communication overhead, effectively addresses the limitations in computational and storage resources in CPS, while ensuring data privacy and security. Specifically, Fed-MPS employs a model parameter selection algorithm to select the optimal parameters to participate in subsequent training. This model parameter selection algorithm can choose the model parameters whose update directions are consistent with that of the server in the previous update. Then, Gaussian noise is added to the optimized model before uploading it for privacy enhancement. Since the model parameters are compressed through parameter selection, adding noise on this basis can protect client users' privacy while reducing the privacy budget. Furthermore, uploading such a model can reduce the communication overhead. Through rigorous privacy analysis and convergence analysis, we prove that the proposed scheme satisfies $(\epsilon, \delta) - DP$ and converges to the global optimum with a convergence ratio of $O(\frac{1}{T^2})$ within T rounds of federated learning. Extensive experiments are conducted on prominent benchmark datasets FashionMNIST, Mnist, and Cifar10. The experimental results demonstrate that compared with baselines, the proposed Fed-MPS scheme can provide higher accuracy for CPS under resource constraints. Although our Fed-MPS scheme can effectively reduce the parameters of model

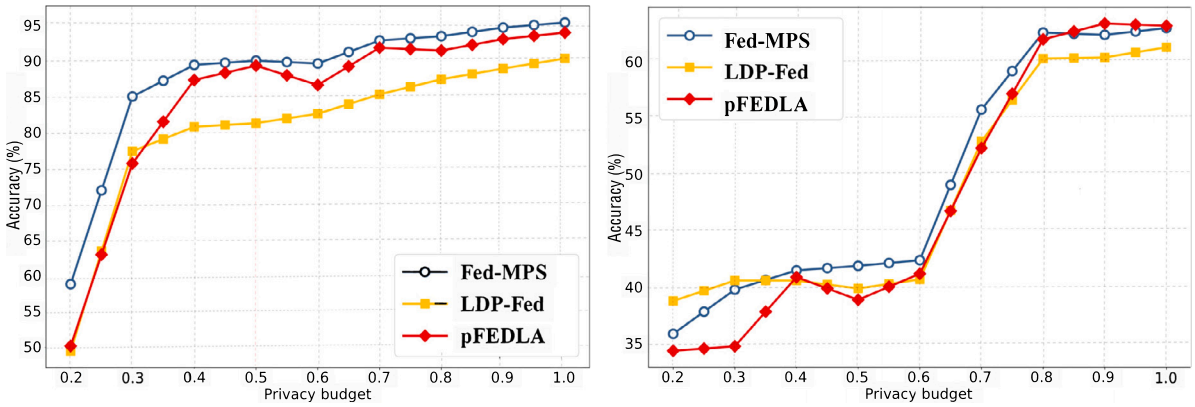


Fig. 7. The accuracy comparison under different privacy budgets on MNIST and CIFAR10.

training, the heterogeneity of the data itself can cause model bias. Privacy protection based on this may lower the reliability of the model. This requires the study of personalized federated learning schemes for data heterogeneity, and the design of corresponding privacy protection, which will be our future research direction.

CRediT authorship contribution statement

Shui Jiang: Writing – original draft. **Xiaoding Wang:** Writing – review & editing, Methodology, Funding acquisition, Conceptualization. **Youxiong Que:** Writing – review & editing, Writing – original draft, Resources, Investigation, Data curation. **Hui Lin:** Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by National Key Research and Development Program of China under Grant No. 2022YFD2301100, Agriculture Research System of China under Grant No. CARS-17, National Natural Science Foundation of China under Grant No. 61702103 and U1905211, and Natural Science Foundation of Fujian Province, China under Grant No. 2020J01167 and 2020J01169.

References

[1] U. Odyurt, A.D. Pimentel, I. Gonzalez Alonso, Improving the robustness of industrial cyber-physical systems through machine learning-based performance anomaly identification, J. Syst. Archit. 131 (2022) 102716.

[2] J. Pang, Z. Han, R. Zhou, H. Tan, Y. Cao, Online scheduling algorithms for unbiased distributed learning over wireless edge networks, J. Syst. Archit. 131 (2022) 102673.

[3] Z. Mo, Z. Gao, C. Zhao, Y. Lin, FedDQ: A communication-efficient federated learning approach for internet of vehicles, J. Syst. Archit. 131 (2022) 102690.

[4] T.-D. Cao, T. Truong-Huu, H. Tran, K. Tran, A federated deep learning framework for privacy preservation and communication efficiency, J. Syst. Archit. 124 (2022) 102413.

[5] X. You, X. Liu, N. Jiang, J. Cai, Z. Ying, Reschedule gradients: Temporal non-IID resilient federated learning, IEEE Internet Things J. 10 (1) (2022) 747–762.

[6] F. Hu, W. Zhou, K. Liao, H. Li, Contribution-and participation-based federated learning on non-IID data, IEEE Intell. Syst. 37 (4) (2022) 35–43.

[7] I. Behnke, C. Blumschein, R. Danicki, P. Wiesner, L. Thamsen, O. Kao, Towards a real-time IoT: Approaches for incoming packet processing in cyber-physical systems, J. Syst. Archit. 140 (2023) 102891.

[8] X. Wang, S. Garg, H. Lin, J. Hu, G. Kaddoum, M. Jalil Piran, M.S. Hossain, Toward accurate anomaly detection in industrial internet of things using hierarchical federated learning, IEEE Internet of Things Journal 9 (10) (2022) 7110–7119, http://dx.doi.org/10.1109/JIOT.2021.3074382.

[9] X. Zhou, X. Zheng, X. Cui, J. Shi, W. Liang, Z. Yan, L.T. Yang, S. Shimizu, K.I.-K. Wang, Digital twin enhanced federated reinforcement learning with lightweight knowledge distillation in mobile networks, IEEE J. Sel. Areas Commun. 41 (10) (2023) 3191–3211, http://dx.doi.org/10.1109/JSAC.2023.3310046.

[10] X. Zhou, X. Zheng, T. Shu, W. Liang, K.I.-K. Wang, L. Qi, S. Shimizu, Q. Jin, Information theoretic learning-enhanced dual-generative adversarial networks with causal representation for robust OOD generalization, IEEE Trans. Neural Netw. Learn. Syst. (2023) 1–14, http://dx.doi.org/10.1109/TNNLS.2023.3330864.

[11] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: 2017 IEEE Symposium on Security and Privacy, SP, IEEE, 2017, pp. 3–18.

[12] C. Dwork, Differential privacy, in: Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33, Springer, 2006, pp. 1–12.

[13] M. Yang, H. Cheng, F. Chen, X. Liu, M. Wang, X. Li, Model poisoning attack in differential privacy-based federated learning, Inform. Sci. 630 (2023) 158–172, http://dx.doi.org/10.1016/j.ins.2023.02.025, URL https://www.sciencedirect.com/science/article/pii/S0020025523002141.

[14] X. Zhou, Q. Yang, X. Zheng, W. Liang, K.I.-K. Wang, J. Ma, Y. Pan, Q. Jin, Personalized federation learning with model-contrastive learning for multi-modal user modeling in human-centric metaverse, IEEE J. Sel. Areas Commun. (2024) http://dx.doi.org/10.1109/JSAC.2023.3345431, 1–1.

[15] X. Zhou, W. Liang, I. Kevin, K. Wang, Z. Yan, L.T. Yang, W. Wei, J. Ma, Q. Jin, Decentralized P2P federated learning for privacy-preserving and resilient mobile robotic systems, IEEE Wirel. Commun. 30 (2) (2023) 82–89.

[16] X. Zhou, X. Ye, K.I.-K. Wang, W. Liang, N.K.C. Nair, S. Shimizu, Z. Yan, Q. Jin, Hierarchical federated learning with social context clustering-based participant selection for internet of medical things applications, IEEE Trans. Comput. Soc. Syst. (2023) 1–10.

[17] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, H. Ludwig, Hybridalpha: An efficient approach for privacy-preserving federated learning, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019, pp. 13–23.

[18] F. Sattler, S. Wiedemann, K.-R. Müller, W. Samek, Robust and communication-efficient federated learning from non-iid data, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2019) 3400–3413.

[19] Z. Yang, M. Chen, W. Saad, C.S. Hong, M. Shikh-Bahaei, Energy efficient federated learning over wireless communication networks, IEEE Trans. Wireless Commun. 20 (3) (2020) 1935–1949.

[20] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, IEEE Trans. Ind. Inform. 16 (10) (2019) 6532–6542.

[21] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, R. Pedarsani, Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2020, pp. 2021–2031.

[22] X. Zhou, W. Liang, K.I.-K. Wang, L.T. Yang, Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations, IEEE Trans. Comput. Soc. Syst. 8 (1) (2021) 171–178.

[23] A. Ghosh, J. Chung, D. Yin, K. Ramchandran, An efficient framework for clustered federated learning, Adv. Neural Inf. Process. Syst. 33 (2020) 19586–19597.

[24] L. Yang, J. Huang, W. Lin, J. Cao, Personalized federated learning on non-IID data via group-based meta-learning, ACM Trans. Knowl. Discov. Data 17 (4) (2023) 1–20.

[25] L. Wang, W. Wang, B. Li, CMFL: Mitigating communication overhead for federated learning, in: 2019 IEEE 39th International Conference on Distributed Computing Systems, ICDCS, IEEE, 2019, pp. 954–964.
[26] X. Ma, J. Zhang, S. Guo, W. Xu, Layer-wised model aggregation for personalized federated learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10092–10101.
[27] C. Wang, X. Wu, G. Liu, T. Deng, K. Peng, S. Wan, Safeguarding cross-silo federated learning with local differential privacy, Digit. Commun. Netw. 8 (4) (2022) 446–454.
[28] Y. Miao, R. Xie, X. Li, X. Liu, Z. Ma, R.H. Deng, Compressed federated learning based on adaptive local differential privacy, in: Proceedings of the 38th Annual Computer Security Applications Conference, 2022, pp. 159–170.
[29] H. Zong, Q. Wang, X. Liu, Y. Li, Y. Shao, Communication reducing quantization for federated learning with local differential privacy mechanism, in: 2021 IEEE/CIC International Conference on Communications in China, ICCC, IEEE, 2021, pp. 75–80.
[30] M. Kim, O. Günlü, R.F. Schaefer, Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2021, pp. 2650–2654.
[31] Z. Chuanxin, S. Yi, W. Degang, Federated learning with Gaussian differential privacy, in: Proceedings of the 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence, 2020, pp. 296–301.
[32] R. Hu, Y. Gong, Y. Guo, Federated learning with sparsification-amplified privacy and adaptive optimization, 2020, arXiv preprint arXiv:2008.01558.
[33] L. Ruixuan, C. Yang, C. Hong, G. Ruoyang, Y. Masatoshi, FLAME: differentially private federated learning in the shuffle model, 2020, CoRR, abs/2009.08063.
[34] L. Sun, J. Qian, X. Chen, Ldp-fl: Practical private aggregation in federated learning with local differential privacy, 2020, arXiv preprint arXiv:2007.15789.
[35] R. Hu, Y. Gong, Y. Guo, Federated learning with sparsified model perturbation: Improving accuracy under client-level differential privacy, 2022, arXiv preprint arXiv:2202.07178.
[36] Y. Wang, Y. Tong, D. Shi, Federated latent dirichlet allocation: A local differential privacy based framework, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 6283–6290.
[37] L. Sun, L. Lyu, Federated model distillation with noise-free differential privacy, 2020, arXiv preprint arXiv:2009.05537.
[38] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, N.N. Xiong, An adaptive federated learning scheme with differential privacy preserving, Future Gener. Comput. Syst. 127 (2022) 362–372.
[39] S.A. Alvi, Y. Hong, S. Durrani, Utility fairness for the differentially private federated-learning-based wireless IoT networks, IEEE Internet Things J. 9 (19) (2022) 19398–19413.
[40] X. Jiang, X. Zhou, J. Grossklags, SignDS-FL: Local differentially private federated learning with sign-based dimension selection, ACM Trans. Intell. Syst. Technol. 13 (5) (2022) 1–22.
[41] L. Javed, A. Anjum, B.M. Yakubu, M. Iqbal, S.A. Moqurrab, G. Srivastava, ShareChain: Blockchain-enabled model for sharing patient data using federated learning and differential privacy, Expert Syst. 40 (5) (2023) e13131.
[42] S. Truex, L. Liu, K.-H. Chow, M.E. Gursoy, W. Wei, LDP-Fed: Federated Learning with Local Differential Privacy, Association for Computing Machinery, New York, NY, USA, 2020, pp. 61–66.
[43] M. Seif, R. Tandon, M. Li, Wireless federated learning with local differential privacy, in: 2020 IEEE International Symposium on Information Theory, ISIT, 2020, pp. 2604–2609.
[44] A. Girgis, D. Data, S. Diggavi, P. Kairouz, A. Theertha Suresh, Shuffled model of differential privacy in federated learning, in: A. Banerjee, K. Fukumizu (Eds.), Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 130, PMLR, 2021, pp. 2521–2529.
[45] B. Jiang, J. Li, H. Wang, H. Song, Privacy-preserving federated learning for industrial edge computing via hybrid differential privacy and adaptive compression, IEEE Trans. Ind. Inform. 19 (2) (2021) 1136–1144.
[46] B. Wang, Y. Chen, H. Jiang, Z. Zhao, PPeFL: Privacy-preserving edge federated learning with local differential privacy, IEEE Internet Things J. (2023).
[47] C. Ren, T. Wang, H. Yu, Y. Xu, Z.Y. Dong, EFedDSA: An efficient differential privacy-based horizontal federated learning approach for smart grid dynamic security assessment, IEEE J. Emerg. Sel. Top. Circuits Syst. (2023).
[48] Y. Wang, X. Zhang, J. Ma, Q. Jin, LDP-fed+: a robust and privacy-preserving federated learning based classification framework enabled by local differential privacy, Concurr. Comput.: Pract. Exper. 35 (19) (2023) e7429.
[49] X. Wang, S. Garg, H. Lin, G. Kaddoum, J. Hu, M.S. Hossain, A secure data aggregation strategy in edge computing and blockchain-empowered internet of things, IEEE Internet of Things Journal 9 (16) (2022) 14237–14246, http://dx.doi.org/10.1109/JIOT.2020.3023588.
[50] X. Wang, H. Zeng, L. Lin, Y. Huang, H. Lin, Y. Que, Deep learning-empowered crop breeding: intelligent, efficient and promising, Frontiers in Plant Science (ISSN: 1664-462X) 14 (2023) http://dx.doi.org/10.3389/fpls.2023.1260089.
[51] P. Kumar, G.P. Gupta, R. Tripathi, PEFL: Deep privacy-encoding-based federated learning framework for smart agriculture, IEEE Micro 42 (1) (2021) 33–40.
[52] C. Yu, S. Shen, K. Zhang, H. Zhao, Y. Shi, Energy-aware device scheduling for joint federated learning in edge-assisted internet of agriculture things, in: 2022 IEEE Wireless Communications and Networking Conference, WCNC, IEEE, 2022, pp. 1140–1145.
[53] I. Mironov, Rényi differential privacy, in: 2017 IEEE 30th Computer Security Foundations Symposium, CSF, IEEE, 2017, pp. 263–275.
[54] Y.-X. Wang, B. Balle, S.P. Kasiviswanathan, Subsampled Rényi differential privacy and analytical moments accountant, in: The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1226–1235.
[55] L. Wang, B. Jayaraman, D. Evans, Q. Gu, Efficient privacy-preserving stochastic nonconvex optimization, 2019, arXiv preprint arXiv:1910.13659.