Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060380

ARTICLE

Differential Privacy Federated Learning Based on Adaptive Adjustment

Yanjin Cheng[1,2], Wenmin Li[1,2,*], Sujuan Qin[1,2] and Tengfei Tu[1,2]

[1]The State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[2]The School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876, China

*Corresponding Author: Wenmin Li. Email: liwenmin@bupt.edu.cn

Received: 30 October 2024; Accepted: 19 December 2024; Published: 06 March 2025

ABSTRACT: Federated learning effectively alleviates privacy and security issues raised by the development of artificial intelligence through a distributed training architecture. Existing research has shown that attackers can compromise user privacy and security by stealing model parameters. Therefore, differential privacy is applied in federated learning to further address malicious issues. However, the addition of noise and the update clipping mechanism in differential privacy jointly limit the further development of federated learning in privacy protection and performance optimization. Therefore, we propose an adaptive adjusted differential privacy federated learning method. First, a dynamic adaptive privacy budget allocation strategy is proposed, which flexibly adjusts the privacy budget within a given range based on the client's data volume and training requirements, thereby alleviating the loss of privacy budget and the magnitude of model noise. Second, a longitudinal clipping differential privacy strategy is proposed, which based on the differences in factors that affect parameter updates, uses sparse methods to trim local updates, thereby reducing the impact of privacy pruning steps on model accuracy. The two strategies work together to ensure user privacy while the effect of differential privacy on model accuracy is reduced. To evaluate the effectiveness of our method, we conducted extensive experiments on benchmark datasets, and the results showed that our proposed method performed well in terms of performance and privacy protection.

KEYWORDS: Federated learning; privacy protection; differential privacy; deep learning

1 Introduction

The acceleration of digitization has resulted in individuals, companies, and countries generating large amounts of data daily. To improve the user experience, machine learning technology can be applied to deeply mine data and gain valuable information. However, it should be noted that in practice, a single data source contains a relatively limited amount of data. The communication and integration of data are hindered by many factors, including regulatory requirements, privacy restrictions, and other considerations. This has led to the creation of siloed data repositories, limiting the potential for data utilization and hindering the further enhancement of model functionality [1,2]. To break through data silos and protect data security [3,4], federated learning (FL) has been applied, which essentially reduces the centralized storage and transmission of data, and thus helps to reduce the risk of data breaches. However, federated learning frameworks still face privacy and security challenges during the application process. For example, attackers can infer user data by analyzing model updates or gradient information during the training process, leading to issues such as data breaches [5,6]. In the increasingly stringent data protection environment, ensuring that federated learning can withstand various potential threats while protecting privacy is a necessary condition for promoting the widespread application of federated learning.

To better ensure user privacy and security, technologies such as statistics, cryptography, and differential privacy have been applied by researchers to federated learning, with differential privacy (DP) [7] being a key concern due to its strict privacy guarantee and other advantages [8]. In existing research, the implementation of differential privacy in the context of federated learning is distinguished by two principal categories: sample-level DP and user-level DP. This distinction is based on the differing definitions of adjacent datasets [9]. User-level differential privacy can prevent an attacker from inferring the data contribution of a single user through model updates, even if the attacker has access to all data except for that user. As a result of this feature aligning more closely with the requirements of federated learning, it has received more attention.

The procedure for implementing user-level DP in federated learning can be roughly outlined as follows: (1) In the initialization phase, the server performs the initialization of the global model parameters. Each client is assigned a privacy budget; (2) Each client receives the global model and the corresponding privacy budget from the server; (3) In each training round, the client loads their own local dataset to train the global model and then performs clipping and noise addition operations on the updated model, which is obtained at the end of the process; (4) The model updates protected by user-level differential privacy are transmitted from the client to the server; (5) The server receives the model updates provided by all clients in the current training round and updates the global model in accordance with the received data. In order to mitigate the impact of differential privacy on the effectiveness of federated models, References [10–13] have proposed that an adaptive noise mechanism be incorporated into the training process. This would serve to conceal the actual parameters transmitted by the client and diminish the quantity of noise. References [14,15] use DP variants for privacy protection in deep neural networks to decrease the effect of privacy noise on the model. Reference [16] improvements to the gradient clipping process, such as using different clip thresholds for gradients uploaded by different users, to decrease the negative effect of model update clipping. However, the above methods only explore measures to reduce the impact of differential privacy on the model from the perspective of single factors such as noise intensity or gradient clipping. In fact, the performance impact of differential privacy on models is influenced by several factors, it is therefore necessary to consider the issues from a comprehensive perspective in order to achieve the best balance between privacy and model performance. Firstly, the heterogeneity of clients and the fixed and identical privacy budgets during the training process, which often leads to an overall imbalance in privacy protection, increases the computational burden of the model and even violates the original update direction of the model. Secondly, although the commonly used update clipping mechanism in differential privacy effectively suppresses privacy leakage, It can also be a source of loss of important information, with a consequent impact on the accuracy and generalizability of the model. Therefore, we propose federated learning based on adaptive adjustment differential privacy protection (AADP-FL) to address the above issues.

Firstly, considering the differences in data volume between each client in the real world and the decreasing trend of model updates in the later stages of training, we dynamically adjust the privacy budget during the training process based on the client's data volume and model convergence speed. A dynamic adaptive privacy budget allocation strategy is proposed to satisfy the differential privacy requirements of the client during training. Secondly, considering the limitation of all factors in the model update, discarding factors with less impact on the model may reduce its impact. Therefore, we propose a longitudinal clipping differential privacy strategy. In addition, we have introduced adaptive optimization techniques to promote the convergence of the federated model to address the situation where the sparsity of model updates leads to a decrease in the convergence speed of the algorithm and an increase in the total number of communication rounds. The main contributions of this paper are summarized as follows:

- We propose a dynamic adaptive privacy budget allocation strategy, this strategy appropriately allocates the privacy budget for each client in each round, and appropriately adjusts the privacy budget in the later stages of training to reduce the increase of noise in the later stages, thus improving the accuracy of the model;
- We propose a longitudinal clipping differential privacy strategy that transforms model updates from horizontal clipping to longitudinal clipping. When determining the clipping threshold, sparse methods are used to reduce the relatively small model update ratio, thereby reducing the impact of clipping on model performance and convergence;
- Extensive experimentations evaluate our scheme on the MNIST, the FMNIST, and the Cifar-10 datasets and provide a privacy analysis of our method. The experimental results show that under the same conditions, our method significantly improves the accuracy of the federated model.

2 Related Work

Federated Learning. A machine learning framework called federated learning was first proposed by Google in 2016 [1], which can effectively support multiple individuals or organizations in data usage and machine learning models while meeting privacy, data security, and government regulatory requirements. McMahan et al. [17] proposed a classical algorithm based on FedAvg for federated averaging, which is widely used in various federated learning scenarios. Because of the characteristics of privacy protection in federated learning, its application in various fields has gained a significant amount of attention, especially in application scenarios such as smart devices, healthcare, and finance [6]. With the widespread promotion of federated learning, privacy breaches have gradually emerged, such as attackers using reverse attacks to obtain user data [4], researchers have applied various security models and privacy protection techniques to federated learning frameworks to ensure data privacy and security, such as secure multiparty computation (MPC) [18,19], homomorphic encryption [20], and differential privacy [21–23]. Among these, differential privacy has received more focus from researchers for its strong privacy guarantees and flexibility.

Differential Privacy. In 2006, Dwork et al. [24] first proposed differential privacy methods. It is a method of protecting data privacy by introducing noise to protect individual privacy and has been rigorously proven mathematically. Due to the inevitable impact of the introduced noise on model performance, researchers have begun to explore optimization strategies for machine learning frameworks that protect differential privacy mechanisms [25,26]. In order to reduce the variance of noise, Yu et al. [27] have developed an algorithm for perturbed low-dimensional gradient embedding and small-norm residual gradient. Their objective was to address the significant increase in differential privacy caused by the large size of the model. Phan et al. [28] put forth a mechanism that deliberately introduces additional noise into features with a lower correlation to the model output. Liu et al. [29] applied differential privacy to multi-agent systems with limited tolerance for faulty agents. However, federated learning typically depends on a single server to aggregate model updates uploaded by different client devices. Clients usually cannot communicate directly with each other, so the above methods may not be directly applicable to federated learning.

Federated Learning with Differential Privacy. The use of user-level differential privacy in federated learning began with McMahan et al. [30], who implemented differential privacy using Gaussian mechanisms and ensured privacy through moment accountants. While differential privacy technology offers a robust security solution in the context of federated learning, it also has the unintended consequence of reducing the accuracy of federated learning models. In order to mitigate the impact of differential privacy technology on federated models, researchers have conducted a substantial body of studies on the subject. The comparative analysis of the existing method is shown in Table 1.

Table 1: Summary of existing research on federated learning methods with differential privacy protection. (✓ represents considering this issue, ✗ represents not considering this issue)

Existing methods	Data heterogeneity	Privacy budget	Parameter update clipping	Adaptive noise adjustment	Accelerate convergence
GFDPFL [14]	✓	✓	✗	✗	✗
DP-SCAFFOLD [31]	✓	✗	✗	✗	✗
PPeFL [32]	✓	✓	✗	✗	✗
Fed-SPA [21]	✓	✗	✓	✗	✓
PPFL [33]	✓	✗	✓	✗	✓
PLDP-FL [34]	✓	✓	✗	✗	✗
CLFLDP [22]	✓	✓	✓	✗	✗
DPFL-AGN [23]	✓	✗	✗	✓	✓
Robust-HDP [35]	✓	✓	✗	✓	✗
Ours	✓	✓	✓	✓	✓

Guo et al. [14] adopted privacy loss distribution (PLD) and privacy curve instead of directly analyzing privacy budget epsilon through various methods, which reduced the errors caused by PLD truncation and discretization, and expanded the discretization interval to reduce computational workload. Wang et al. [32] proposed three local DP mechanisms to address privacy issues in the federated learning process, among which the Exponential Mechanism Filtering and Screening (FS-EM) was proposed based on the contribution of weight parameters to the neural network, filtering out better global aggregation parameters. This method not only solves the problem of rapidly increasing privacy budget when perturbation mechanisms are applied locally but furthermore, it significantly cuts down on communication costs. Noble et al. [31] used the DP-SCAFFOLD framework to address the issue of data heterogeneity under DP constraints, applying user-level differential privacy to each client. Hu et al. [21] proposed a method that integrates the sparsification of randomness and gradient disturbance for each agent to enhance privacy protection in response to the problem of the DP mechanism introducing random noise proportional to the model size. In addition, acceleration technology was introduced to reduce privacy costs. Weng et al. [33] strengthened the privacy protection of participants by applying DP locally and centrally, and to improve the accuracy and performance of the model, sparsification gradients were implemented on both the server and client sides. Shen et al. [34] put forward a perturbation algorithm that addresses the issue of insufficient or excessive privacy protection for certain participants due to the application of the exact same privacy budget settings for all clients. In doing so, the algorithm takes into account the differences in privacy requirements among different clients. The algorithm permits customers to modify privacy parameters in accordance with the sensitivity of their data, thereby enabling the system to provide personalized privacy protection. Chen et al. [22] allocated privacy budgets based on client similarity and employed a layer-pruning method based on gradient correlation to reduce communication overhead. This approach resulted in a reduction in both the loss of privacy budgets and the size of model noise. Jiao et al. [23] put forth the concept of an adaptive Gaussian noise. This scheme protects the data privacy and security of the federated learning training process by adding adaptive Gaussian noise during the training process, which hides the real parameters uploaded by the client. Malekmohammadi et al. [35] proposed a noise-aware robust algorithm for heterogeneous DP-FL by analyzing the privacy requirements of the client and the heterogeneity of batch or dataset sizes, which improved the utility and convergence speed of the model.

Most of the above methods explore ways to reduce the impact of differential privacy on the model from the perspective of privacy budget allocation, adaptive noise adjustment, and threshold clipping. The effect of differential privacy on the performance of the model is influenced by multiple factors, so it is necessary to consider these issues from a comprehensive perspective in order to achieve privacy protection while maximizing the practicality of the model.

3 Preliminary

3.1 Federated Learning

Federated Learning [1] is a distributed machine learning method where data is maintained by each participant locally and model training is also performed locally for distributed computing. Because the data used in federated learning comes from a variety of end-user devices, these data are usually not independent of each other or evenly distributed. In this setting, FL normally considers the following optimization problem:

$$\begin{aligned} &\min\left[F(x)=\frac{1}{N}\sum_{i=1}^{N}f_i(x)\right] \\ &\text{where } f_i(x)=\mathbb{E}_{z\sim p_i}\left[f(x;z)\right] \end{aligned} \tag{1}$$

where N is the number of participating clients; $f_i(x)$ is the loss function of the i-th client; $f(x;z)$ is the loss of a model x at an example z; each of N clients has a local data distribution p_i.

The widely used federated learning algorithm currently available is FedAvg [36]. The general process of the algorithm is shown in Fig. 1, and the training process of the algorithm includes:

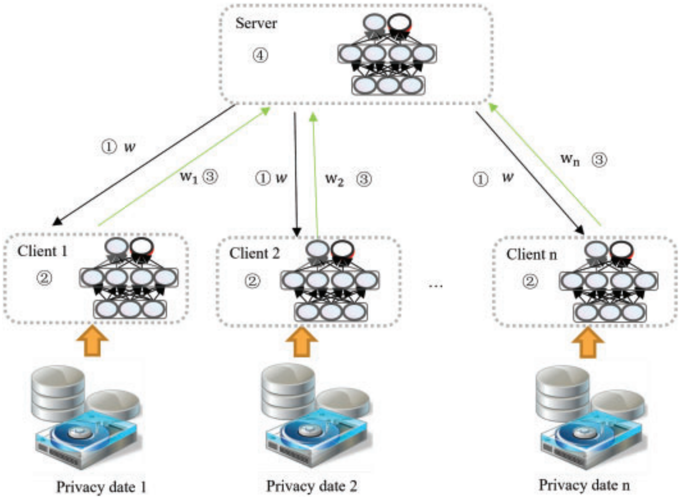


Figure 1: Federated learning framework

Firstly, the server randomly selects the client and distributes the federated model parameters (or initial model parameters) w to the client;

Secondly, the client uses local data for a determined amount of training rounds and updates the local model parameters; Then, the client transfers the locally trained model parameters to the server;

Finally, the server receives the model parameter gradient loaded by the local client and aggregates them to form a new federated model.

Repeat this process until the federated model has converged, or until it has completed the specified number of training rounds.

3.2 Threat Model

In the context of federated learning, as described in this paper, the server does not directly access data, is only responsible for model aggregation and coordination, and is considered "honest but curious," i.e., it follows the algorithm requirements to correctly perform model updates, aggregation, and other operations without tampering with data or algorithms. Each client independently owns and stores its private data and does not directly share information with other clients, making it impossible for external attackers to steal private information directly from the client. All clients can upload local models in a timely and normal manner, without allowing malicious parameters to be uploaded to the server, and without client failures or other situations. Therefore, it can be concluded that the following situations may lead to privacy leakage:

(1) The server has a strong interest in the privacy data or information it can obtain due to its "curiosity". It may attempt to infer raw data or obtain other potentially sensitive items of information from the model parameters received from the client.

(2) External attackers can obtain model parameter updates loaded by the client by intercepting communication between the clients and the server. Then, using attack methods such as gradient flipping, they can infer privacy information from the model updates.

3.3 Differential Privacy for Federated Learning

Due to the requirement that servers cannot identify a client's participation by observing the output of local updates, user-level DP has received much attention in research on the integration of federated learning and differential privacy. The difference between user-adjacent datasets and adjacent datasets is that the units faced change from a single sample to all samples of a certain user.

The Gaussian mechanism is often used to ensure user-level DP in FL.

Definition 1. (The Gaussian mechanism of $(\varepsilon, \delta) - DP$). To satisfy $(\varepsilon, \delta) - DP$, the Gaussian noise mechanism $\mathscr{M}: D \in \mathbb{N}^{|\mathscr{X}|}$ is defined as $\mathscr{G}_{\sigma}\mathscr{M}(D) = \mathscr{M}(D) + N(0, \sigma^2 I)$, where $\sigma > \frac{\Delta\mathscr{M}\sqrt{2\log\frac{1.25}{\sigma}}}{\varepsilon}$, I is an identity matrix.

4 Our Approach

The application of traditional user-level DP in federated learning frameworks can result in high information loss and excessive noise addition, leading to low model accuracy and slow convergence speed. To address such issues, we propose AADP-FL, which is a federated learning method with user-level DP, as shown in Algorithm 1.

First, each client has a different amount of data, so it is necessary to allocate privacy budgets of different sizes to each client, thereby ensuring that their information is more effectively safeguarded. In subsequent training stages, as the model converges to a certain degree, the privacy budget is changed to mitigate the impact of noise on the model.

Secondly, the information discarded during the differential privacy clipping process has a substantial effect on the efficiency and convergence speed of the federated model. Therefore, we propose a longitudinal clipping differential privacy strategy. This strategy uses sparsity methods to minimize the less influential parts in model updates to reduce the $L2$ norm of model updates.

Algorithm 1: AADP-FL

Input: number of rounds T, number of clients n, number of local iterations τ, client sampling probability $p \in (0,1]$, batch size B, clipping threshold S, privacy budget ε, privacy budget threshold ε_{max}, momentum parameters u, v, learning rates $\eta_c, \eta s$, noise reduction rate $\rho\,(0 < \rho < 1)$, convergence threshold l.

Output: Trained model w^T.

```
Sever:
Initial global model $w_0$
all clients $i$, $d_i$ in parallel do
    $\varepsilon_i = \left(\frac{d_i}{\frac{\sum d_i}{n}}\right)\varepsilon$
if $t = 1$ do
    Selecting client set $W$ with a probability of $p$;
    for $i \in W$ in parallel do
        $\hat{\theta}_i^t = Users(w^{t-1}, i, \varepsilon_i^t)$;
    end for
else if
    for $t = 2$ to $T$ do
        Selecting client set $W$ with a probability of $p$;
        if $|m_t - m_{t-1}| < l$ and $|n_t - n_{t-1}| < l$ do
            if $\varepsilon_i^{t-1} < \varepsilon_{max}$ do
                $\varepsilon_i^t = \varepsilon_i^{t-1}\frac{1}{\rho}$
        for $i \in W$ in parallel do
            $\hat{\theta}_i^t = Users(w^{t-1}, i, \varepsilon_i^t)$;
        end for
        $g = \frac{\sum_{i=1}^{|W|}\hat{\theta}_i^t}{|W|}, \hat{g} = \frac{g}{1-u^t}$;
        $m_t = um_{t-1} + (1-u)g, \hat{m}_t = \frac{m_t}{1-u^t}\bar{m}_t = (1-u)\hat{g} + u\hat{m}_t$;
        $n_t = vn_{t-1} + (1-v)g^2, \hat{n}_t = \frac{n_t}{1-v_t}$;
        $w^t = w^{t-1} + \eta_s\frac{\bar{m}_t}{\sqrt{\hat{n}_t + \gamma}}$;
    end for
return $w^T$
Users:
$w_i^{t,0} = w^{t-1}$;
for $j = 1$ to $\tau$ do
    $w_i^{t,j} = w_i^{t,j-1} - \eta_c\frac{\sum\nabla f_i}{|B|}$
end for
$\theta_i^t = w_i^{t,\tau} - w_i^{t,0}$;
$\tilde{\theta}_i^t = longitudinal - clip(\theta_i^t, S)$;
return $\tilde{\theta}_i^t + N\left(0, \frac{S^4 I}{|w|\varepsilon^2}\right)$
```

4.1 Framework of AADP-FL

Similarly, AADP-FL is a server-client architecture, as illustrated in Fig. 2. The framework in this paper is divided into the following steps:

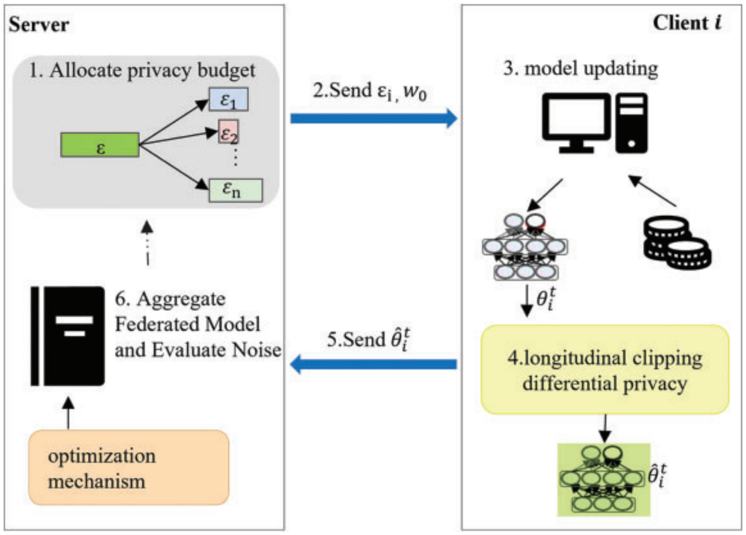

Server
1. Allocate privacy budget
ε
ε_1
ε_2
ε_n
2.Send ε_i, w_0
Client i
3. model updating
θ_i^t
4.longitudinal clipping differential privacy
$\hat{\theta}_i^t$
5.Send $\hat{\theta}_i^t$
6. Aggregate Federated Model and Evaluate Noise
optimization mechanism

Figure 2: Overall framework of AADP-FL

Initialization: The server initializes the federated model w and the clipping threshold C.

Step 1: On the server side, calculate the privacy budget ε_i that client i should allocate based on the amount of data it owns.

Step 2: The server sends the federated model w and the privacy budget ε_i of the selected client.

Step 3: After receiving the federated model, the client updates the model with local data to obtain the updated θ of the model.

Step 4: The client performs a longitudinal clipping differential privacy strategy on the updated θ_i^t of the model, and adds an appropriate proportion of noise.

Step 5: The client sends the resulting $\hat{\theta}_i^t$ to the server.

Step 6: The server runs an aggregation operation on the received $\hat{\theta}_i^t$, and then uses the optimization mechanism to update the federated model and obtain a new federated model w.

Step 7: The server uses the changes in the first-order and second-order moment estimates in the optimization mechanism to determine the convergence of the federated model. If it is deemed to be close to convergence, check whether the privacy budget is higher than expected. If not, decrease the volume of the noise to be added, otherwise the noise level remains unchanged.

Step 8: Repeat steps 2–7 until the FL aggregation number reaches the specified value.

4.2 Dynamically Adaptive Privacy Budget Allocation Strategy

The dynamically adaptive privacy budget allocation strategy primarily modifies the privacy budget on two nodes. First, as the amount of data contained by the clients varies, the server allocates different sizes of privacy budgets to each client based on the amount of data before training begins. Ensure that clients with different amounts of data obtain reasonable trade-offs in the allocation of privacy budgets. The privacy budget allocated to the i-th client is:

$$\varepsilon_i = \left(\frac{n * d_i}{\sum d_i}\right)\varepsilon, \tag{2}$$

Among them, ε is the overall privacy budget, and d_i is the amount of data contained in the i-th client, n is the total number of clients. This allocation method distributes the privacy budget more equitably and can better balance privacy and model performance.

Second, in the early stages of the training phase of federated learning, when the model updates significantly, the added noise has a relatively small impact on the overall model. As the number of training epochs increases, the model updates become smaller, and the same amount of noise as in the early training stage can have a significant impact on the overall model. Even in the later stages of training, excessive noise can prevent the model from converging to the optimal state. Therefore, we choose to reduce the noise appropriately after the model has converged to a certain degree to ensure model performance. To accelerate model convergence, we used the Nadam optimizer.

The Nadam optimizer includes mechanisms for traditional momentum and prediction of future gradient updates, including first and second-order moment estimates. The first-order moment estimate is the mean of the gradient and is used to guide the main direction of parameter updates; the second-order moment estimate is the mean of the squared gradient and is used to adaptively adjust the learning rate. When both the first and second-order moment estimates become stable, it usually indicates that the direction and variance of the gradient are no longer changing significantly. At this point, the model is approaching convergence, thus reducing the amount of noise. The reduction of noise is determined by the following formula:

$$\varepsilon_i^t := \begin{cases} \varepsilon_i^{t-1}\frac{1}{\rho}, & \text{if } |m_t - m_{t-1}| < l \; and \; |n_t - n_{t-1}| < l \; and \; \varepsilon_i^{t-1} < \varepsilon_{max} \\ \varepsilon_i^{t-1} & \text{otherwise.} \end{cases} \tag{3}$$

Among them, l is the decay threshold and ρ is the decay parameter. If $|m_t - m_{t-1}| < l$ and $|n_t - n_{t-1}| < l$, check whether the privacy budget exceeds the expected threshold. If it does not, the privacy budget of all clients in round t increases uniformly, Otherwise, keep the privacy budget from the previous round and continue to train.

4.3 Longitudinal-Clipping Differential Privacy Strategy

The longitudinal-clipping differential privacy strategy refers to performing clipping operations on a single parameter dimension. Specifically, this method uses the sparsity method to clip user parameters, as shown in Algorithm 2. In previous DP-FL studies, the main role of sparsity methods was to expand the effectiveness of privacy protection. This method can also play a role in reducing the impact of clipping updates on the model; this method takes the clipping threshold as the boundary and utilizes the method of sparsification of specific parameters to update and reset updates, to reduce the effect of clipping operations on the global model and better preserve user information in local updates. The definition of sparser is as follows:

Definition. (Sparser) For parameter $k \in [d]$, the operator is defined for a vector $x \in \mathbb{R}^d$ as

$$\text{Sparse}(x, k) = \begin{cases} 0 & \text{if id} = k \\ x_{\text{id}} & \text{otherwise} \end{cases}$$

The mechanism of the sparse clipping method works as follows:

Before threshold clipping begins, the client marks the position ID of each element x_{id} of θ_i^t.

(1) The client sorts the elements in θ_i^t from smallest to largest based on their absolute values and records their position ID.

(2) The client calculates the second normal form of the current θ_i^t and compares it with the clipping threshold S.

(3) If $\|\theta_i^t\|_2$ is larger than S, the client sets the first non-zero element in the absolute value sequence to zero and repeats steps (2)–(3).

Algorithm 2: Longitudinal-Clip

Input: parameters updates θ_i^t, clipping threshold S.

Output: clipped parameters updates $\tilde{\theta}_i^t$.

```
label the position $id$ of each element $w_{id}$ of $\theta_i^t$
while $\left\|\theta_i^t\right\|_2 > S$ do
    $k = id$ where $min(|w_{id}|)$
    Clip based on sparsifier $\theta_i^t = Sparse_i^t(\theta_i^t, k)$
return $\tilde{\theta}_i^t$
```

5 Privacy Analysis

In this method, each client incorporates a specific quantity of Gaussian noise into the local training model, without rounds, in order to mitigate the attacks outlined in the threat model. Therefore, we only evaluate privacy loss from the perspective of the individual customer. Next, we will analyze the privacy loss of the k-th client.

Theorem 1. *(Privacy Loss of Algorithm 1) After T rounds, the i-th client DP Privacy budget of Algorithm 1 satisfies $(\varepsilon_i, \delta) - DP$ if σ satisfies:*

$$\varepsilon_i \leq \sum_{t=0}^{T} \frac{1}{\alpha - 1} \sum_{j=0}^{\alpha} \binom{\alpha}{j} q^j (1-q)^{\alpha - j} exp\left(\frac{j\,(j-1)}{2\sigma^2}\right) + \frac{\log \frac{1}{\delta}}{\alpha - 1}$$

where $(\cdot)$ denotes the binomial coefficient, $q = \frac{batchsize}{d_i}$ and any integer $\alpha \geq 2$.

We use RDP to calculate the loss of privacy. Firstly, the privacy cost of each round is calculated using the subsampling Gaussian theorem of RDP. Subsequently, an advanced combination of RDP is employed, accompanied by the associated cost of multiple rounds. Ultimately, the RDP privacy is transformed into DP.

Definition 1. (Subsampled Gaussian Mechanism [37]) Let f be a function mapping subsets of C to $\mathbb{R}^d$. We define the Subsampled Gaussian Mechanism parameterized with the sampling rate $0 < q \leq 1$ and the $\sigma > 0$ as:

$$SG_{q,\sigma} \triangleq f\left(\{x : x \in C \text{ } is \text{ } subsampled \text{ } with \text{ } probability \text{ } q\}\right) + \mathcal{N}\left(0, \sigma^2 \mathbb{I}^d\right)$$

f is the clipped gradient evaluation in subsampled data points $f\left(\{x_i\}_{i \in B}\right) = \sum_{i \in B} \bar{g}_t\left(x_i\right)$. In the event that $\bar{g}_t$ is obtained by clipping g_t with a gradient norm bound S, it can be demonstrated that the sensitivity of f equals S.

Definition 2. (RDP privacy budget of Subsampled Gaussian Mechanism [37]) A Gaussian mechanism function f l_2-sensitivity is 1 and satisfies $(\alpha, \varepsilon) - RDP$ if:

$$\varepsilon \leq \frac{1}{\alpha - 1} \log \max \left(A_\alpha \left(q, \sigma \right), B_\alpha \left(q, \sigma \right) \right)$$

Due to the explanation in Reference [37]:

$$\begin{cases} A_\alpha \left(q, \sigma \right) \triangleq \mathbb{E}_{z \sim \mu_0} \left[\left(\mu \left(z \right) / \mu_0 \left(z \right) \right)^\alpha \right] \\ B_\alpha \left(q, \sigma \right) \triangleq \mathbb{E}_{z \sim \mu} \left[\left(\mu_0 \left(z \right) / \mu \left(z \right) \right)^\alpha \right] \end{cases}$$

With $\mu_0 \triangleq \mathcal{N} \left(0, \sigma^2 \right)$, $\mu_1 \triangleq \mathcal{N} \left(1, \sigma^2 \right)$ and $\mu \triangleq \left(1 - q \right) \mu_0 + q \mu_1$.

The two Gaussian distributions μ_0 and μ_1 are used in the RDP, and they satisfy

$$A_\alpha (q, \sigma) \geq B_\alpha \left(q, \sigma \right).$$

From this, f satisfies $\left(\alpha, \frac{1}{\alpha - 1} \log A_\alpha \left(q, \sigma \right) \right)$-RDP.

After that, We obtained the compute method of $A_\alpha (q, \sigma)$ on integer α from Reference [38].

$$A_\alpha = \sum_{j=0}^{\alpha} \binom{\alpha}{j} q^j (1 - q)^{\alpha - j} exp \left(\frac{j \left(j - 1 \right)}{2\sigma^2} \right)$$

Based on Definition 2 and the calculation method of $A_\alpha (q, \sigma)$, we can conclude that the prerequisite for f to satisfy $\left(\alpha, \frac{1}{\alpha - 1} \log A_\alpha (q, \sigma) \right)$-RDP is

$$\varepsilon \leq \frac{1}{\alpha - 1} \sum_{j=0}^{\alpha} \binom{\alpha}{j} q^j (1 - q)^{\alpha - j} exp \left(\frac{j \left(j - 1 \right)}{2\sigma^2} \right)$$

Definition 3. (Composition of RDP [38]). For two randomized mechanisms f, g such that f is (α, ε_1)-RDP and g is (α, ε_2)-RDP the composition of f and g which is defined as (X, Y) (a sequence of results), where $x \sim f$ and $Y \sim g$, satisfies $(\alpha, \varepsilon_1 + \varepsilon_2)$-RDP.

We assume that the privacy budget undergoes adaptive changes when the model approaches convergence in this method, so the changes can be ignored. From the above, Lemma 1 can be derived.

Lemma 1. Given the sampling rate $q = batchsize/d_i$ for each round of the local dataset and σ as the noise factor for round t, the total RDP privacy loss of the i-th client for round T loss for any integer $\alpha \geq 2$ is:

$$\varepsilon_i \leq \sum_{t=0}^{T} \frac{1}{\alpha - 1} \sum_{j=0}^{\alpha} \binom{\alpha}{j} q^j (1 - q)^{\alpha - j} exp \left(\frac{j \left(j - 1 \right)}{2\sigma^2} \right)$$

Definition 4. (Translation from RDP to DP [39]) If a randomized mechanism f: $D \longrightarrow R$ satisfies (α, ε)-RDP, then it satisfies $\left(\varepsilon + \frac{\log 1/\delta}{\alpha - 1}, \delta \right)$-DP where $0 < \delta < 1$.

With Lemma 1 and Definition 4, Theorem 1 is proved. We use the result of Theorem 1 to calculate the privacy cost.

6 Experiment Settings

In this section, we conducted comparative experiments between the MNIST dataset [40], the Fashion-MNIST [41], and the CIFAR-10 dataset [42].

Datasets: The MNIST dataset comprises an image dataset of 10 types of handwritten digits, consisting of $70k$ monochrome images of the digits, including $60k$ training samples and $10k$ test samples. Each image is 28 by 28 pixels and contains a number between 0 and 9. The Fashion-MNIST (FMNIST) is a dataset containing 10 categories of fashion clothing images, with 6000 training images and 1000 test images for each category, making a total of 70,000 images. Each image is a 28 × 28 pixel monochrome image with pixel values ranging from 0 to 255. The CIFAR-10 dataset is a collection of images used for identifying common items. It contains $50k$ training samples and $10k$ test samples, including 10 types of RGB color images.

In the experiment, we segment the dataset in two distinct ways: (1) IID Data setting, whereby samples are distributed evenly across each client; (2) Non-IID Data settings, whereby the client exhibited imbalanced samples [43].

Models and Environment: We conducted experiments on models with different structures, using three different models for the MNIST, FMNIST, and CIFAR-10 datasets, including the MLP model with two fully connected layers, the CNN used for the MNIST and FMNIST datasets with two convolutional layers and two fully connected layers (each filter size is 5 × 5), and the CNN used for the CIFAR-10 dataset with three fully connected layers. The AlexNet model consists of 5 convolutional layers and 2 fully connected layers, with a ReLU activation function and a pooling layer after each convolutional layer. All methods were implemented using PyTorch and all experiments were conducted with an NVIDIA GeForce RTX 4090 GPU.

Configuration: In this experiment, the number of clients is 20. For MNIST, FMNIST, and CIFAR-10 respectively, we set the number of rounds T to 50, 200, and 300, the batch size to 32, 32, and 50, and the percentage of non-IID is 0.5. In all experiments, the failure probability δ of differential privacy is set to $1e-5$, the momentum parameters u, v are set to $0.9, 0.999$, the noise reduction rate ρ is set to 0.1, privacy budget threshold ε_{max} is set to 20, the local and global learning rates are 0.01.

7 Experimental Results

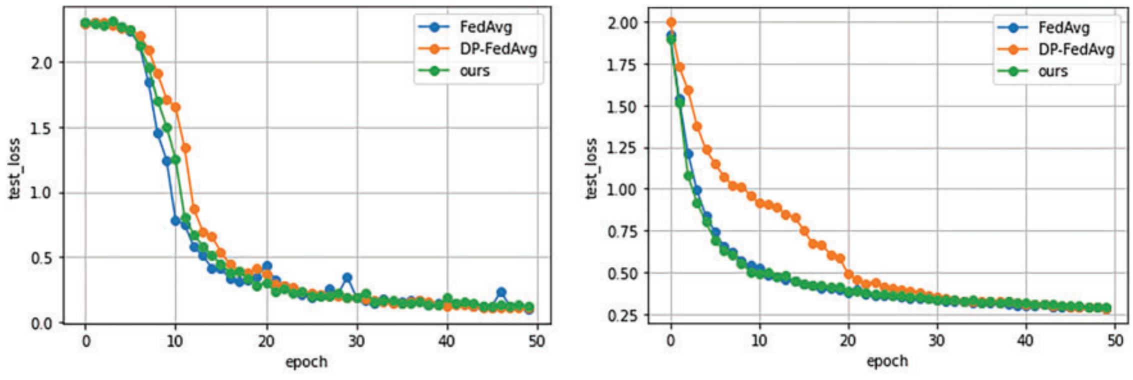
We have examined the behavior of AADP-FL on different datasets and benchmarked it against FedAvg, DP-FedAvg, DDGauss [44], and DP-SCAFFOLD [31] methods in the case of non-iid. Specifically, we apply deep learning models such as MLP, CNN, and AlexNet to both the MNIST and CIFAR-10 datasets for training in the framework. Considering the randomness of the disturbance process, this paper conducted 5 experimental experiments and used the midpoint as the experimental results.

Performance Evaluation. Table 2 presents the exact accuracy data for each algorithm, obtained under identical experimental conditions. Training on the MNIST dataset is completed after 50 rounds of communication, the training on the FMNIST dataset is completed after 200 rounds of communication, while training on the CIFAR−10 dataset is concluded after 300 rounds of communication. Among the algorithms under consideration, FedAvg represents a baseline method that does not include any form of privacy protection or communication compression. In contrast, alternative algorithms employ differential privacy protection mechanisms. The parameters utilized in this experiment are consistent with the benchmark settings delineated in this article, with a privacy budget of 4. The experimental results demonstrate that in comparison with FedAvg, the deployment of a differential privacy method entails a certain degree of compromise in model performance, with the objective of achieving varying degrees of privacy protection across different models. The performance of the DDGauss and DP-SCAFFOLD methods is comparable, with the latter exhibiting excellent performance. In particular, with regard to the MNIST dataset, the discrepancy in precision terms between our method and FedAvg is less than one percentage point. For the FMNIST dataset, the accuracy difference between our method and FedAvg is less than 2 percentage points. In the case of the CIFAR-10 dataset, the accuracy difference between our method and FedAvg is less than 2 percentage points. The discrepancy in the outcomes of the multiple experiments is not substantial, suggesting that the reliability of each method is largely comparable. Our method is demonstrably more accurate.

Convergence. Fig. 3 displays the evolution of the test set loss for various models in the MNIST data set as a function of the number of training epochs. Fig. 3a depicts the experimental outcomes of the MLP model. It is evident from the figure that our methodology involved a comparison of loss values under identical conditions prior to the model reaching a state of convergence. Fig. 3b illustrates the experimental outcomes of the AlexNet model. Furthermore, it is evident that during the testing phase, the loss values of this method reach a state of stability with greater rapidity. A reduction in test loss value and an increase in stability are indicative of a model with good convergence and generalization abilities during the training process. In light of these findings, it can be concluded that irrespective of the model employed, this method exhibits superior convergence and generalization capabilities. This is mainly due to two aspects: first, the dynamic adaptive privacy budget proposed in this paper automatically adjusts the privacy budget during training, reducing noise. Second, the longitudinal clipping differential privacy strategy reduces the effects of clipping on model behavior and minimizes the influence of differential privacy on convergence speed.

Table 2: Performance comparison under different models on different datasets (%)

Dataset	Model	FedAvg	DP-FedAvg	DDGauss	DP-SCAFFOLD	Ours
MNIST	MLP	92.13 ± 0.34	89.88 ± 0.24	90.37 ± 0.22	90.21 ± 0.23	91.60 ± 0.22
	CNN	93.29 ± 0.21	90.02 ± 0.26	91.25 ± 0.15	91.02 ± 0.17	93.28 ± 0.16
	AlexNet	95.98 ± 0.31	93.04 ± 0.19	94.08 ± 0.21	93.05 ± 0.20	94.99 ± 0.20
FMNIST	MLP	78.84 ± 0.41	73.91 ± 0.34	75.17 ± 0.28	75.21 ± 0.24	75.60 ± 0.32
	CNN	82.79 ± 0.35	78.02 ± 0.25	80.35 ± 0.30	80.62 ± 0.27	81.28 ± 0.22
	AlexNet	84.37 ± 0.34	79.44 ± 0.29	82.58 ± 0.33	82.15 ± 0.27	83.09 ± 0.22
CIFAR-10	MLP	48.55 ± 0.24	41.25 ± 0.21	45.25 ± 0.17	45.66 ± 0.20	47.68 ± 0.18
	CNN	54.35 ± 0.19	48.33 ± 0.14	53.63 ± 0.15	50.28 ± 0.16	54.87 ± 0.15
	AlexNet	70.66 ± 0.22	63.25 ± 0.17	66.27 ± 0.11	65.72 ± 0.13	68.52 ± 0.16

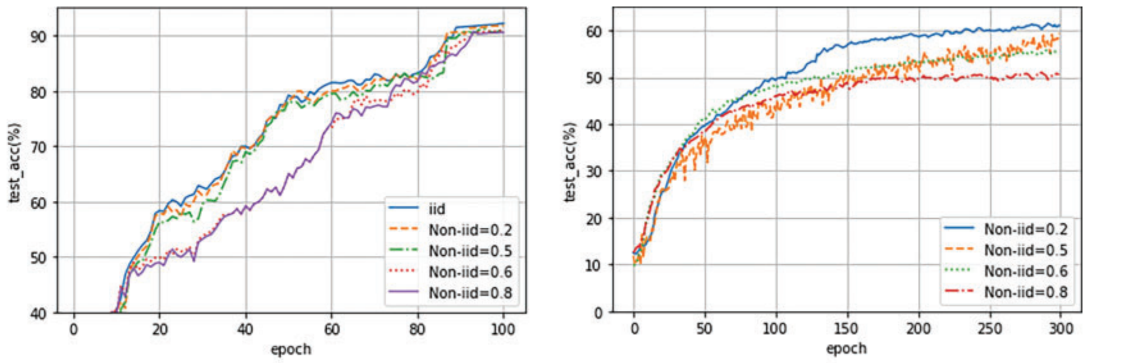


(a) Losses tested using the MLP model.

(b) Losses tested using the AlexNet model.

Figure 3: Comparison of test losses on the MNIST dataset

The impact of different data distributions. The non-iid setting simulates the situation where data distributed across different clients in real scenarios is not fully consistent. Fig. 4a shows the experimental results of using the MLP model on the MNIST dataset under different iid settings. Due to the relatively simple nature of the MNIST dataset, the final results presented under different iid settings differ slightly. However, it can still be observed that the greater the non-iid, the slower the model achieves its optimum results. Fig. 4b shows the experimental results of using a CNN model on the Cifar-10 dataset under different iid settings. The Cifar-10 dataset is relatively complex, so it can be directly seen that the larger the non-iid, the greater the difference in data distribution and the lower the accuracy of the model. This indicates that our method is still sensitive to non-IID data and does not have the ability to resist non-IID data.

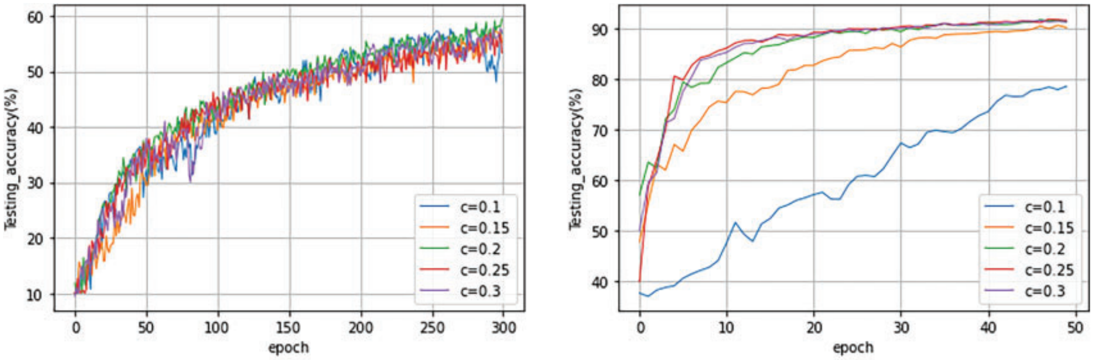


(a) Accuracy in the CIFAR-10 dataset and CNN model. (b) Accuracy in the MNIST dataset and MLP model.

Figure 4: Test the accuracy of different data distributions under different settings

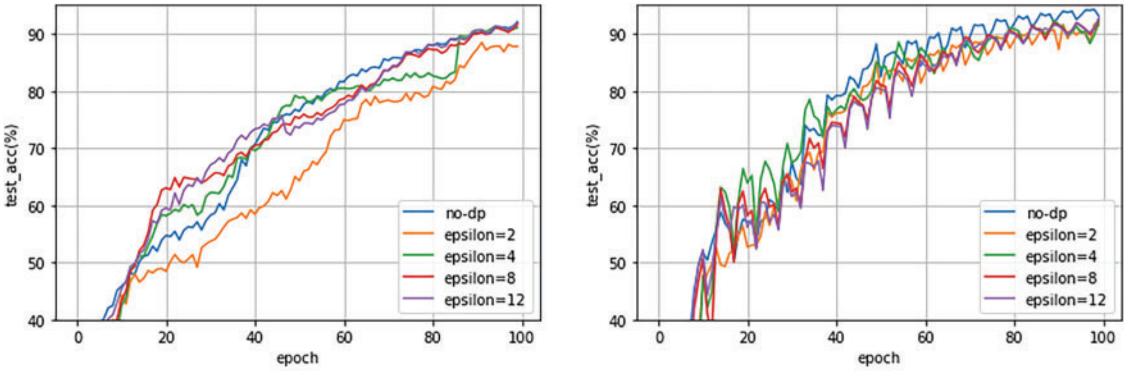
The impact of clipping threshold. Given the fixed $L2$ norm of the model update, it is imperative to exercise caution when establishing the clipping threshold. If the threshold is set too high, it will result in non-clipping and the addition of excessive differential noise, which will have a detrimental effect on the precision of the model. If the threshold is set too low, a significant amount of user information will be discarded, potentially leading to a decline in model accuracy. To reduce the influence of other factors, this experiment only performs threshold clipping and does not add additional noise. Fig. 5a indicates the impact of varying shear thresholds on model accuracy when utilizing the CIFAR-10 dataset and CNN model in experimental settings. It is evident that there is a notable discrepancy in the outcomes between the clipping thresholds of 0.15 and 0.3. However, it is clear that the accuracy is considerably enhanced when the threshold is set to 0.2 in comparison to the other thresholds. Fig. 5b indicates the impact of varying shear thresholds on model accuracy when utilizing the MNIST dataset and MLP model in experimental settings. It is evident that there are considerable discrepancies in the experimental outcomes at varying thresholds. Notably, the model accuracy at a shear threshold of 0.25 exhibits a marked improvement in comparison to the other thresholds.

The impact of privacy budget. Fig. 6 shows the impact of different privacy budgets on the accuracy of the tests using MLP and CNN models on the MNIST dataset. Fig. 6a shows the experimental results of the MLP model on the MNIST dataset, and Fig. 6b depicts the experimental results of the CNN model. Under different privacy budgets, our method has a smaller difference in accuracy compared to frameworks that do not use differential privacy. We also observed that the performance of more complex models (CNN) is less affected when privacy budgets fluctuate compared to simpler models (MLP), which is a favorable advantage since we tend to use more complex models to obtain better outcomes. Although the complex model is less affected when the privacy budget fluctuates, the accuracy of the model is still significantly affected when the privacy budget is set below the general level. Therefore, in practical applications, highly sensitive scenarios such as medical data and financial data may have a smaller privacy budget range for security settings (e.g., set the range to $(0.5-1)$); low sensitivity scenarios (such as personal actions, recommendation systems, etc.) may have a larger privacy budget range (e.g., set the range to $(10-100)$).



(a) Accuracy in the CIFAR-10 dataset, CNN model. (b) Accuracy in the MNIST dataset and MLP model.

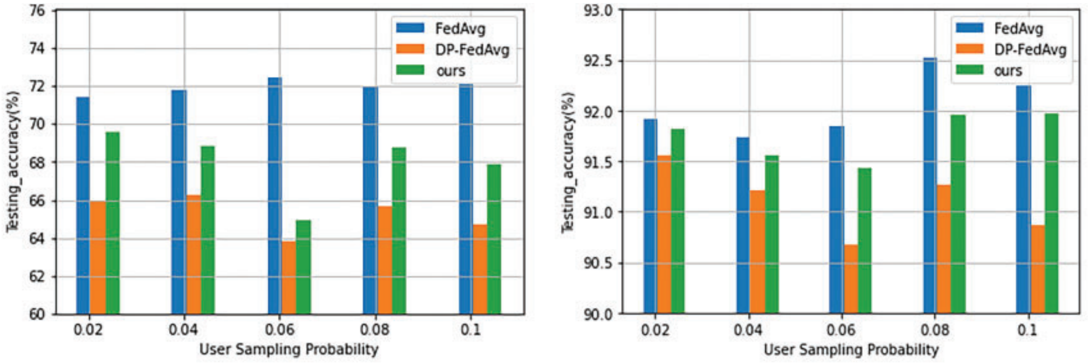
Figure 5: Test the accuracy of different clipping thresholds at different settings



(a) Accuracy in the MLP model. (b) Accuracy in the CNN model.

Figure 6: Testing the accuracy of different privacy budgets in the MNIST dataset

The impact of user sampling probability. A comparative experimental analysis was conducted to evaluate the impact of varying user sampling probabilities on different datasets. The shear boundary is set to 0.3 for both DP FedAvg and our method. Fig. 7a illustrates the model accuracy of the CIFAR-10 dataset and AlexNet model at varying user sampling probabilities, whereas Fig. 7b depicts the model accuracy of the MNIST dataset and MLP model at distinct user sampling probabilities. A higher user sampling probability signifies an increased number of agents engaged in each communication round. It is evident that disparate user sampling rates yield disparate model outcomes. At an identical sampling rate, this method yields superior outcomes. When the user sampling rate is high, the performance comparison between this method and the comparison method is more pronounced, indicating that this method demonstrates robust stability in the context of multi-user participation.



(a) Accuracy in the CIFAR-10 dataset, AlexNet model. (b) Accuracy in the MNIST dataset and MLP model.

Figure 7: Testing the accuracy of different user sampling probabilities under different settings

8 Conclusion

This paper proposes AADP-FL, which includes a dynamic adaptive privacy budget allocation strategy and a longitudinal differential privacy clipping strategy. This method mitigates the impact of differential privacy on model accuracy while maintaining user privacy. We conducted a theoretical analysis of the privacy of the method and verified it through experiments. The method can be applied to smart applications on mobile devices, health monitoring on smart wearable devices, and other scenarios. However, due to the deeper impact of differential privacy on the performance of larger models, further research is needed on the privacy security of complex models. In addition, there are still uncertainties in the practical application of federated learning in terms of data complexity, user identity security (possibly malicious users), and communication. Future research could consider combining differential privacy with meta-learning, knowledge distillation, and encryption algorithms to facilitate the practical application of federated learning.

Acknowledgement: The authors wish to express their appreciation to the reviewers and editors for their helpful suggestions which greatly improved the presentation of this paper.

Funding Statement: This research was funded by the Science and Technology Project of State Grid Corporation of China (Research on the theory and method of multiparty encrypted computation in the edge fusion environment of power IoT, No. 5700-202358592A-3-2-ZN), and the National Natural Science Foundation of China (Grant Nos. 62272056, 62372048, 62371069).

Author Contributions: Study conception and design: Yanjin Cheng; data collection: Tengfei Tu; analysis and interpretation of results: Yanjin Cheng, Wenmin Li, Sujuan Qin; draft manuscript preparation: Yanjin Cheng, Wenmin Li. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used to support the findings of this study are publicly available on the Internet as follows: MNIST: http://yann.lecun.com/exdb/mnist/ (accessed on 20 August 2024); FMNIST: https://research.zalando.com/welcome/mission/research-projects/fashion-mnist/ (accessed on 20 August 2024); CIFAR-10: https://www.cs.toronto.edu/kriz/cifar.html (accessed on 20 August 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Huang W, Wang D, Ouyang X, Wan J, Liu J, Li T. Multimodal federated learning: concept, methods, applications and future directions. Inf Fusion. 2024;112:102576. doi:10.1016/j.inffus.2024.102576.
2. Song Y, Wu Y, Wu S, Li D, Wen Q, Qin S, et al. A quantum federated learning framework for classical clients. Sci China: Phys Mech Astron. 2024 May;67(5). doi:10.48550/arXiv.2312.11672.
3. Wang S, Tuor T, Salonidis T, Leung KK, Makaya C, He T, et al. When edge meets learning: adaptive control for resource-constrained distributed machine learning. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications; 2018; Honolulu, HI, USA. p. 63–71. doi:10.1109/INFOCOM.2018.8486403.
4. Huang R-Y, Samaraweera D, Chang JM. Exploring threats, defenses, and privacy-preserving techniques in federated learning: a survey. Computer. 2024 Apr;57(4):46–56. doi:10.1109/MC.2023.3324975.
5. Lim WYB, Luong NC, Hoang DT, Jiao Y, Liang Y-C, Yang Q, et al. Federated learning in mobile edge networks: a comprehensive survey. IEEE Commun Surv Tutorials. 2020;22(3):2031–63.
6. Gupta M, Kumar M, Dhir R. Unleashing the prospective of blockchain-federated learning fusion for IoT security: a comprehensive review. Comput Sci Rev. 2024;54:100685.
7. Dwork C, Roth A. The algorithmic foundations of differential privacy. Foundat Trends® Theoret Comput Sci. 2014;9(3–4):211–407. doi:10.1561/0400000042.
8. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security; 2016. p. 308–18. doi:10.1145/2976749.2978318.
9. Zhang X, Chen X, Hong M, Wu ZS, Yi J. Understanding clipping for federated learning: convergence and client-level differential privacy. In: International Conference on Machine Learning, ICML 2022; Baltimore, MD, USA; 2022.
10. Wei W, Liu L, Zhou J, Chow K-H, Wu Y. Securing distributed SGD against gradient leakage threats. IEEE Trans Parall Distrib Syst. 2023 Jul;34(7):2040–54. doi:10.1109/TPDS.2023.3273490.
11. Xue R, Xue K, Zhu B, Luo X, Zhang T, Sun Q, et al. Differentially private federated learning with an adaptive noise mechanism. IEEE Trans Inf Forensics Secur. 2024;19:74–87. doi:10.1109/TIFS.2023.3318944.
12. Fu J, Chen Z, Han X. Adap DP-FL: differentially private federated learning with adaptive noise. In: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom); 2022 Dec; Los Alamitos, CA, USA: IEEE Computer Society. p. 656–63.
13. Wang J, Zhang Z, Tian J, Li H. Local differential privacy federated learning based on heterogeneous data multi-privacy mechanism. Comput Netw. 2024;254(3):110822. doi:10.1016/j.comnet.2024.110822.
14. Guo S, Yang J, Long S, Wang X, Liu G. Federated learning with differential privacy via fast fourier transform for tighter-efficient combining. Sci Rep. 2024;14(1):26770. doi:10.1038/s41598-024-77428-0.
15. Ling J, Zheng J, Chen J. Efficient federated learning privacy preservation method with heterogeneous differential privacy. Comput Secur. 2024;139:103715. doi:10.1016/j.cose.2024.103715.
16. Chen L, Ding X, Bao Z, Zhou P, Jin H. Differentially private federated learning on non-iid data: convergence analysis and adaptive optimization. IEEE Trans Knowl Data Eng. 2024 Sep;36(9):4567–81.
17. McMahan HB, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of Machine Learning Research; 2016; Fort Lauderdale, FL, USA.
18. Bonawitz K, Ivanov V, Kreuter B, Marcedone A, Mcmahan HB, Patel S, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security; 2017; Dallas, TX, USA.
19. Ma Y, Woods J, Angel S, Polychroniadou A, Rabin T. Flamingo: multi-round single-server secure aggregation with applications to private federated learning. In: 2023 IEEE Symposium on Security and Privacy (SP); 2023 May; Los Alamitos, CA, USA: IEEE Computer Society. p. 477–96.
20. Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. Privacy-preserving deep learning: revisited and enhanced. In: Batten L, Kim D, Zhang X, Li G, editors. Applications and techniques in information security. Singapore: Springer. Vol. 719, p. 100–10. doi:10.1007/978-981-10-5421-1_9.

21. Hu R, Gong Y, Guo Y. Federated learning with sparsification-amplified privacy and adaptive optimization. arXiv preprint arXiv:2008.01558. 2020.
22. Chen S, Yang J, Wang G, Wang Z, Yin H, Feng Y. CLFLDP: communication-efficient layer clipping federated learning with local differential privacy. J Syst Archit. 2024;148:103067. doi:10.1016/j.sysarc.2024.103067.
23. Jiao S, Cai L, Wang X, Cheng K, Gao X. A differential privacy federated learning scheme based on adaptive gaussian noise. Comput Model Eng Sci. 2023;138(2):1679–94. doi:10.32604/cmes.2023.030512.
24. Dwork C. Differential privacy: a survey of results. In: International Conference on Theory and Applications of Models of Computation; 2008; Berlin: Springer. p. 1–19.
25. Lu Z, Asghar HJ, Kaafar MA, Webb D, Dickinson P. A differentially private framework for deep learning with convexified loss functions. IEEE Trans Inf Forens Secur. 2022;17:2151–65. doi:10.1109/TIFS.2022.3169911.
26. Phan NH, Vu M, Liu Y, Jin R, Thai MT. Heterogeneous gaussian mechanism: preserving differential privacy in deep learning with provable robustness. In: The 28th International Joint Conference on Artificial Intelligence (IJCAI-19); 2019; Macao, China.
27. Yu D, Zhang H, Chen W, Liu T-Y. Do not let privacy overbill utility: gradient embedding perturbation for private learning. In: International Conference on Learning Representations; 2021; Austria.
28. Phan N, Wu X, Hu H, Dou D. Adaptive laplace mechanism: differential privacy preservation in deep learning. In: 2017 IEEE International Conference on Data Mining (ICDM); 2017 Nov; Los Alamitos, CA, USA: IEEE Computer Society. p. 385–94.
29. Liu B, Zhao C. Trade-off between privacy and accuracy in resilient vector consensus. In: 2024 American Control Conference (ACC); 2024; Toronto, ON, Canada. p. 1807–12.
30. McMahan HB, Ramage D, Talwar K, Zhang L. Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963. 2017.
31. Noble M, Bellet A, Dieuleveut A. Differentially private federated learning on heterogeneous data. In: International Conference on Artificial Intelligence and Statistics. PMLR; 2022; Spain. p. 10110–45.
32. Wang B, Chen Y, Jiang H, Zhao Z. PPeFL: privacy-preserving edge federated learning with local differential privacy. IEEE Internet Things J. 2023. doi:10.1109/JIOT.2023.3264259.
33. Weng S, Zhang L, Feng D, Feng C, Wang R, Klaine PV, et al. Privacy-preserving federated learning based on differential privacy and momentum gradient descent. In: 2022 International Joint Conference on Neural Networks (IJCNN); 2023; Padua, Italy. p. 1–6. doi:10.1109/JIOT.2023.3264259.
34. Shen X, Jiang H, Chen Y, Wang B, Gao L. PLDP-FL: federated learning with personalized local differential privacy. Entropy. 2023;25(3). doi:10.3390/e25030485.
35. Malekmohammadi S, Yu Y, Cao Y. Noise-aware algorithm for heterogeneous differentially private federated learning. In: International Conference on Machine Learning, ICML 2024; 2024; Vienna, Austria. Vol. 235, p. 34461–98.
36. McMahan HB, Moore E, Ramage D, y Arcas BA. Federated learning of deep networks using model averaging. arXiv preprint arXiv:1602.05629v1. 2016.
37. Mironov I, Talwar K, Zhang L. Rényi differential privacy of the sampled gaussian mechanism. arXiv preprint arXiv:1908.10530. 2019.
38. Mironov I. Rényi differential privacy. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF); 2017 Aug; Los Alamitos, CA, USA: IEEE Computer Society. p. 263–75.
39. Zhou Y, Liu X, Fu Y, Wu D, Li C, Yu S. Optimizing the numbers of queries and replies in federated learning with differential privacy. arXiv preprint arXiv:2107.01895. 2021.
40. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.
41. Xiao H, Rasul K, Vollgraf R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. 2017. doi:10.48550/arXiv.1708.07747.
42. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. In: Handbook of systemic autoimmune diseases; 2009.

43. Caldas S, Duddu SMK, Wu P, Li T, Konečnỳ J, McMahan HB, et al. LEAF: a benchmark for federated settings. arXiv preprint arXiv:1812.01097. 2018.
44. Kairouz P, Liu Z, Steinke T. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In: International Conference on Machine Learning. PMLR; 2021. p. 5201–12.