Efficient and Privacy-Enhanced Federated Learning Based on Parameter Degradation

Wenling Li, Ping Yu, Yanan Cheng, Jianen Yan, and Zhaoxin Zhang

***Abstract*—Federated Learning ensures that clients can collaboratively train a global model by uploading local gradients, keeping data locally, and preserving the security of sensitive data. However, studies have shown that attackers can infer local data from gradients, raising the urgent need for gradient protection. The differential privacy technique protects local gradients by adding noise. This paper proposes a federated privacy-enhancing algorithm that combines local differential privacy, parameter sparsification, and weighted aggregation for cross-silo setting. Firstly, our method introduces Rényi differential privacy by adding noise before uploading local parameters, achieving local differential privacy. Moreover, we dynamically adjust the privacy budget to control the amount of noise added, balancing privacy and accuracy. Secondly, considering the diversity of clients' communication abilities, we propose a novel Top-K method with dynamically adjusted parameter upload rates to effectively reduce and properly allocate communication costs. Finally, based on the data volume, trustworthiness, and upload rates of participants, we employ a weighted aggregation method, which enhance the robustness of the privacy framework. Through experiments, we validate the effective trade-off among privacy, accuracy, communication costs and robustness achieved by the proposed method.**

***Index Terms*—Federated learning, differential privacy, communication costs, credibility, aggregation.**

I. Introduction

A. Background

With the development of big data, artificial intelligence has reached new heights. However, achieving high-precision learning models requires large-scale and high-quality data support [1]. Unfortunately, different organizations are hesitant to contribute their own data due to privacy concerns. This has resulted in data silos, which hinder effective data integration. Federated learning (FL) is an emerging distributed machine learning approach that addresses this issue by enabling data privacy and ownership protection. In Federated Learning, participants train their models locally and upload the model parameters to a central server, which aggregates these parameters, updates the global model, and sends the updated model back to the participants. This process is iterated until the global model reaches a predefined accuracy or convergence condition [2]. In the realm of FL, there exist two predominant configurations: cross-device and cross-silo. In cross-device FL, participants typically comprise edge devices such as smart gadgets and laptops, which may number in the thousands or even millions. These participants are generally considered unreliable and possess limited computational capabilities. In contrast, in the cross-silo FL paradigm, the stakeholders are organizations; the number of participants is relatively limited, usually ranging between 2 and 100 [3] [4]. Given the nature of the participants, the process is generally deemed reliable, and each entity possesses significant computational resources. Cross-silo scenarios are exceedingly common in real-world applications, such as credit card fraud detection [5], clinical disease prediction [6], 6G network [7] and so on. The focus of this study primarily revolves around the cross-silo FL setting.

In many cross-silo scenarios [5] [7], the federated learning architecture is applied as a privacy protection scheme, i.e. the interaction between the server and the client remains the original model parameters. Although federated learning does not require sharing local data, it is possible to infer relevant privacy information about local nodes from parameter updates alone. Related works have shown that the exposure of gradients can reveal information about the class representation of other honest nodes [8] [9], sensitive attributes in samples that are not relevant to the main task [10], and even local training samples [11] [12] [13]. Therefore, enhancing the privacy protection of federated learning models is necessary. This is also the problem that this paper is dedicated to study—privacy enhancement method under cross-silo federated learning.

Differential privacy (DP) achieves the privacy goal through data perturbation and introduces minimal additional computational burden, making it widely applicable in various scenarios. This paper also embraces differential privacy as a means to enhance privacy in federated learning. Differential privacy achieves privacy preservation by injecting noise into the parameters. The greater the amount of noise, the stronger the privacy, albeit at the expense of accuracy. Privacy refers to the extent to which details of the client's local data are protected from disclosure. Accuracy refers to the degree to which the trained final model predicts accurately in the face of new samples. Consequently, balancing privacy with accuracy is a focal point in the design of differential privacy approaches.

Irrespective of the privacy protection method employed, communication cost remains a pivotal challenge to address. Communication refers to the number of parameters interacting between the server and the participants throughout the training process. During each epoch, participants need to transmit their local model parameters to a central server or other participants. However, model parameters can typically number in the tens of thousands or even millions, and transmitting such

Wenling Li is with the Faculty of Computing of Harbin Institute of Technology, Harbin, Heilongjiang 150001, China. E-mail: 21b903074@stu.hit. edu.cn.

Ping Yu, Yanan Cheng, Jianen Yan and Zhaoxin Zhang are with the Harbin Institute of Technology, Harbin, Heilongjiang 150001, China. E-mail: yuping0428@hit.edu.cn, mrcheng0910@gmail.com, yanjianen@hit.edu.cn and zhangzhaoxin@hit.edu.cn

Corresponding authors: Jianen Yan and Zhaoxin Zhang

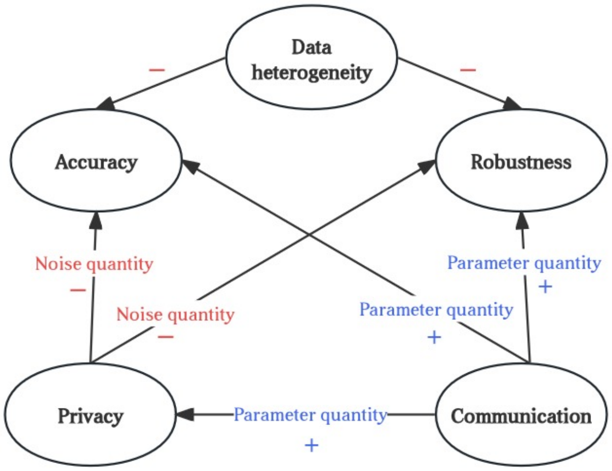


Fig. 1. Federated privacy enhances architectural performance relationships.

large amounts of data over limited network bandwidth may lead to communication delays or failures, ultimately reducing the overall efficiency of federated learning [2]. Additionally, for differential privacy, the amount of noise added is directly proportional to the number of parameters; more noise leads to a greater impact on model accuracy. In addition, there is heterogeneity in equipment, computing capacity and communication capacity of each participant, and reasonable adaptive adjustment should be made according to the actual situation. Therefore, adaptively achieving a balance between model accuracy, privacy, and communication cost remains an important topic.

Finally, the robustness of global model aggregation are key issues in privacy-enhanced federated learning. Robustness refers to the ability of the model training process to proceed normally in the face of foreseeable and unforeseeable disturbances. In this paper, the predictable disturbance is a carefully designed noise added to achieve differential privacy. Unforeseeable disturbances are mainly attacks, such as model poisoning attacks launched by malicious attackers. Against the backdrop of data distribution heterogeneity, there are variations in the quality and quantity of local data among participants. The process of aggregating noised parameters from multiple local nodes to obtain a comprehensive model representation is critical to the success of this approach. The effectiveness of the local node training process can significantly impact the quality of global model aggregation. Several factors can influence the effectiveness of local node training, such as the quality of training data, the amount of noise, the number of parameters uploaded, and malicious attackers, etc. Therefore, the credibility of local parameters must be thoroughly assessed to ensure the quality of aggregation.

The heterogeneity of client data distribution makes the local data quality of clients uneven, which will directly affect the accuracy and robustness of aggregation. Privacy is achieved by adding noise, and the amount of noise also affects accuracy and robustness. The amount of noise can be controlled by the number of interaction parameters, but the accuracy and robustness of the model will be adversely affected if the number of communication parameters is too low. To sum up, building an efficient and robust federated privacy enhancement architecture must consider privacy, accuracy, communication, and robustness. The relationship between these properties is shown in Fig. 1, where the '+' indicates positive correlation and the '−' indicates negative correlation.

B. Contributions

To address these challenges, namely how to balance privacy, accuracy, communication cost, and aggregation robustness, we propose a federated privacy-enhanced architecture. Our algorithm adaptively adjusts the privacy budget and parameter upload rate and employs importance-weighted aggregation to achieve robust learning in scenarios involving malicious participants.

We summarize the main contributions of this paper as follows:

- We introduce a simple yet effective dynamic privacy budget adjustment mechanism for Rényi differential privacy. This adjustment, based on changes in global model accuracy within a given time window, directly mitigates the accuracy decline caused by added noise.
- Addressing the issue of communication cost, we propose an adaptive parameter upload rate adjustment method based on communication latency. This method first assesses the capabilities of participating nodes and then dynamically adjusts the parameter upload rate based on the heterogeneity of node capabilities.
- We propose an importance-weighted aggregation method. By evaluating the contribution of local node parameters to the global model through multiple factors and considering the credibility of parameters by integrating both local-global and intra-local node relationships, we effectively enhance the robustness and efficiency of global model aggregation.

The rest of this paper is organized as follows. Section II introduces some related works in the past. Section III shows the primitives used in the paper. Section IV describes the design of method in detail. Section V presents the performance of the method. Finally, Section VI concludes the paper and declare the future work. The main mathematical notations in this article are summarized in Table I.

II. Related works

A. Privacy

In recent years, various schemes have been proposed to address privacy threats in federated learning. Secure multi-party computation (SMC) [14], homomorphic encryption [15], and differential privacy are among the most researched directions. SMC and homomorphic encryption, both based on encryption [16] [17], demand high computational power and come with other limitations, restricting their practical application. Differential privacy, a privacy-preserving technique based on perturbation, achieves varying degrees of privacy protection

This article has been accepted for publication in IEEE Transactions on Services Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSC.2024.3399659

TABLE I
DESCRIPTION OF MAIN NOTATIONS.

Notation	Description
K	Number of clients
$x_1, ..., x_{n_k}$	Examples on client k
$\mathcal{B}$	The local minibatch Examples
B	Local batchsize
E	The number of local epochs
η_l^k	The learning rate of client k
C	The clipping threshold
z^k	The noise added to client k, and $z^k \sim \mathcal{N}(0, \sigma^2 C^2)$
ϵ	The total privacy budget
p_t^k	The parameter upload rate of client k in round t
T	The total number of training rounds
acc	The model test accuracy in server
Δacc	Accuracy variation value
$\mathcal{L}(w)$	Loss function, $\mathcal{L}(w) = \frac{1}{n_k}\sum_i \mathcal{L}(w, x_i)$

by adding carefully designed noise to samples [18], local parameters [19] [20], etc. Compared to record-level DP, client-level DP is more effective in preventing information leakage in FL settings. Our focus is primarily on client-level differential privacy to protect the local model updates of participating training clients.

Various methods have been proposed in literature for client-level privacy in federated learning. One approach is to add noise to the global model during server aggregation, thereby concealing the individual contributions of participants. Yuan et al. [21] and Zhao et al. [22] achieve differential privacy by adding Gaussian and Laplace noise, respectively. Another approach is to require participants to add noise locally when uploading model updates, preventing the server from inferring privacy information. Wu et al. [23] add Gaussian noise multiple times to the participant's local training process to satisfy differential privacy. The primary challenge in implementing differential privacy in federated learning is balancing privacy with accuracy. Previous works mostly relied on complex external techniques like shuffling [24], combining encryption with noise interference [25], or adjusting noise based on attribute contributions [26] and model dimensions [27] [28]. These methods are often complex, limited to specific scenarios, and incur additional overhead. For instance, [25] combined differential privacy with SMC, reducing noise injection but significantly increasing computational overhead.

In our research, we introduce a simple yet effective dynamic privacy budget adjustment mechanism for Rényi differential privacy (RDP). Based on changes in global model accuracy within a given historical time window, we dynamically adjust the privacy budget for the next round, directly mitigating the accuracy decline caused by added noise.

B. Communication

Besides privacy and accuracy, communication volume is a direct factor affecting both. In Federation learning, communication cost refers to the number of parameters in an interaction. The greater the traffic, the greater the number of parameters to interact with. Differential privacy at the client-level is to add noise to each parameter before interaction, and the larger the number of parameters, the more noise is needed to achieve the same degree of privacy. But at the same time, the amount of communication is less, that is, fewer parameters are uploaded, and the incomplete parameters of the model will inevitably introduce the inaccuracy of the model. Therefore, the trade-off among privacy, accuracy, and communication cost needs simultaneous consideration. [29] focuses on methods that balance communication efficiency and privacy preservation, without delving into the trade-off between data privacy and model accuracy. In [30], the algorithm for dynamically adjusting the privacy budget requires constant monitoring and updating of the privacy situation of each client, introducing additional computation and communication overhead that may increase the complexity and latency of the system.

Parameter sparsification is a widely used technique to reduce communication costs in federated learning. In [31] and [1], it is proposed that during training, clients can upload gradients with absolute values greater than a certain threshold. Wang et al. [32] further consider gradient selection based on sparsity and variance of the gradient, while Alistarh et al. [33] provide convergence analysis of the Top-K gradient selection algorithm. In the literature [24] [34], participants are allowed to select gradient parameters with larger absolute values in proportion to the global model, considering the positive correlation between the absolute values of the parameters and their effects on the global model. Meanwhile, the other parameters are set to zero.

However, the determination of the parameter selection rate has not been thoroughly studied in specific works. Typically, the parameter selection rate is treated as a hyperparameter to demonstrate the stability of the model. In the experimental sections of both [8] and [11], comparative results of method accuracy under different parameter selection rates are provided.

These works typically fix the threshold K, limiting the lower bound of local information compression rate and consequently increasing communication costs to some extent. In addition, there is heterogeneity in equipment, computing capacity and communication capacity of each participant, and reasonable adaptive adjustment should be made according to the actual situation. In our study, we propose an adaptive parameter upload rate adjustment method based on communication latency. This method first assesses the capabilities of participating nodes and then dynamically adjusts the parameter upload rate based on the heterogeneity of node capabilities, thereby making the lower bound of local information compression rate more compact and effectively reducing communication costs.

C. Global Model Aggregation

In terms of global model aggregation, the process is critical in the context of heterogeneous data distribution, where variations in quality and quantity of local data exist among participants. Aggregating noised parameters from multiple local nodes to form a comprehensive model is crucial for the success of differential privacy methods. Noise presence, coupled with potential malicious external attackers, can hinder correct convergence of aggregation algorithms. Relatively few studies have focused on the robustness of privacy enhancement methods based on differential privacy. Shen et al. [35]

proposed a regularization method to enhance the robustness of training models against noise injection. Li et al. [36] proposes a federal proximal algorithm (FedProx), which enhances the aggregation effect by adding regularization to the local loss function. Lu et al. [37] proposed a method that calculates the L2 distance between the local model of participants and the global model of the previous round, assigning different weights to participants based on the probability density function of the Gaussian distribution. The closer the L2 distance is to the center, the higher the weight, and the aggregation result is the weighted average of the uploaded gradients. Wu et al. [38] utilized the variance reduction technique SAGA to reduce the noise of random gradients, making malicious gradients easier to distinguish. They further improved the security of the model by combining it with the median aggregation algorithm.

However, it should be noted that the assumptions of these aggregation algorithms do not necessarily hold in all scenarios and some methods only applicable to data sets with independent and identical distribution (IID), which may not be suitable for federated learning. For example, the median-based aggregation algorithm designed by Yang et al. [39] relies on the assumption that the Lipschitz feature of the normal gradient is very close to the true gradient. Although these methods have shown effectiveness under certain conditions, they have limitations when dealing with non-IID data sets and may not be robust enough in the context of federated learning.

In this study, we propose an importance-weighted aggregation method. This method assesses the contributions of local node parameters to the global model using multiple factors, and it considers the credibility of parameters by integrating both local-global and intra-local node relationships. We effectively improve the robustness and efficiency of global model aggregation by weighting the aggregated noised parameters based on their utility.

A comparison analysis of existing surveys related to FL is summarized in Table II. Where the privacy model contains Secure Multi-party Computation (SMC), Homomorphic Encryption (HE), Difference Privacy (ϵ-DP), Approximate Difference Privacy ((ϵ, δ)-DP) and Rényi differential privacy ((α, ϵ)-RDP.

III. Preliminary

In scenarios where data is centrally stored in a single organization for data processing, such as data publishing, centralized differential privacy can effectively protect the privacy of the entire dataset. However, in federated learning, where data is stored in a decentralized manner across multiple participants, local differential privacy is often utilized to defend against inference attacks on shared parameters. Prominent companies like Google [40], Apple, and Microsoft [41] have integrated local differential privacy into their products. In a federated learning system that leverages local differential privacy, users upload perturbed parameter values instead of the original ones, ensuring that the perturbed parameters are safeguarded against privacy inference attacks. The privacy level of differential privacy is determined by the privacy budget ϵ. A lower privacy budget provides a higher degree of privacy protection, but it can also lead to lower model accuracy. To achieve more stringent control over the privacy budget and enhance its applicability in different environments, several variants have been developed. In the following section, we delve into differential privacy and its various adaptations.

Definition 1. ((ϵ, δ)-differential privacy, (ϵ, δ)-DP) [42]: A randomization mechanism M satisfies (ϵ, δ)-differential privacy ($\epsilon > 0$, $\delta > 0$) when and only when for any adjacent input datasets D and D' and any possible set of output values R_M, there is

$$Pr[M(D) \in R_M] \leq e^{\epsilon} \cdot Pr[M(D') \in R_M] + \delta \tag{1}$$

The relaxed differential privacy definition (ϵ, δ)-DP can be understood as the mechanism satisfies ϵ-DP with a minimum $1 - \delta$ probability.

Sequential Composition: If $F_1(x)$ satisfies (ϵ_1, δ_1)-DP, while $F_2(x)$ satisfies (ϵ_2, δ_2)-DP, then the mechanism $G(x) = (F_1(x), F_2(x))$ satisfies $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP.

Parallel Composition: If the dataset D is divided into k disjointed subdata blocks $x_1 \cup ... \cup x_k = D$, $F(x_1), ..., F(x_k)$ satisfy (ϵ_1, δ)-DP,...,(ϵ_k, δ)-DP respectively, then the mechanism for publishing all results $F(x_1), ..., F(x_k)$ is satisfied $(max(\epsilon_1, ..., \epsilon_k), \delta)$-DP.

Definition 2. (ℓ_2-Sensitivity) [43]: For the real-valued function f acting on the dataset D and D', the ℓ_2 sensitivity of s is expressed as

$$s = \underset{D,D'}{max} \parallel f(D) - f(D') \parallel_2 \tag{2}$$

Sensitivity is the degree to which a change in a single piece of data has the greatest impact on the overall database query result.

Lemma 1. DP for Gaussian mechanism: One way to make the mechanism satisfy differential privacy is to add noise to the results. Gaussian mechanisms help mechanisms achieve differential privacy by adding noise that satisfies a Gaussian distribution. But the Gaussian mechanism cannot satisfy ϵ-differential privacy, it can satisfy (ϵ, δ)-differential privacy. For a random function F(x), the Gaussian mechanism can be used to obtain a random function satisfying (ϵ, δ)-differential privacy $F'(x)$:

$$F'(x) = F(x) + \mathcal{N}(\sigma^2) \tag{3}$$

where $\sigma^2 = \frac{2s^2 ln(1.25/\delta)}{\epsilon^2}$, and s is the sensitivity of F to quantify the level of data privacy exposure, $\mathcal{N}(\sigma^2)$ denotes the sampling result of Gaussian (normal) distribution with mean is 0 and variance is ϵ^2. One of the advantages of the Gaussian mechanism is that the Gaussian noise added to achieve privacy protection has the same type as the other noise sources; in addition, the sum of two Gaussian distributions is still a Gaussian distribution, so the effect of the privacy mechanism on the statistical analysis may be easier to understand and correct [43].

TABLE II
COMPARISONS WITH EXISTING REVIEW WORKS RELATED TO FL.

References	Research objectives	The privacy model employed	Local data distribution	Limitations
[14]	Privacy	SMC	IID	High computing power and communication capability are required.
[15]	Privacy	HE	IID	High computing power and communication capability are required.
[21]	Privacy	(ϵ, δ)-DP	IID	The server must be fully trusted.
[22]	Privacy	ϵ-DP	IID	The server must be fully trusted.
[24]	Privacy	(ϵ, δ)-DP	IID & Non-IID	Additional trusted shuffles are required for assistance
[25]	Privacy	(ϵ, δ)-DP & SMC	IID	It reduces noise injection but significantly increases computational overhead.
[26]	Privacy	ϵ-DP	IID	It adjusts noise based on attribute contributions, limited to specific scenarios.
[27]	Privacy	ϵ-DP	IID	The adjustment method based on parameter dimension is complicated.
[28]	Privacy	(ϵ, δ)-DP	IID	The adjustment method based on parameter dimension is complicated.
[29]	Privacy	(α, ϵ)-RDP	IID	Without delving into the trade-off between data privacy and model accuracy.
[30]	Privacy	(α, ϵ)-RDP	IID & Non-IID	It introduces additional computation and communication overhead that may increase the complexity and latency of the system.
[34]	Privacy & Communication	SMC	IID & Non-IID	It does not consider the dynamic adjustment of K value.
[35]	Privacy & Robustness	(ϵ, δ)-DP	IID & Non-IID	Only suitable for a small range of noise deviation Settings.
[37]	Robustness	—	IID & Non-IID	The influence of local model deviation on global model is not considered.
[39]	Robustness	—	IID	It only applicable to datasets with IID distribution.

Definition 3. (Rényi differential privacy, RDP) [44]: If for all the Neighboring dataset D and D', the random mechanism F satisfies

$$\frac{1}{\alpha - 1} \ln \left(\frac{F(x)}{F(x')}\right)^{\alpha} \leq \epsilon \tag{4}$$

Then this mechanism $F(x)$ satisfies (α, ϵ)-RDP. The idea of Rényi differential privacy is mainly to use Rényi Divergence to measure the relationship between the distributions of two datasets.

Sequential Composition: If $F_1(x)$ satisfies (α, ϵ_1)-RDP, while $F_2(x)$ satisfies (α, ϵ_2)-RDP, then the Composition mechanism of $F_1(x), F_2(x)$ satisfies $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.

Lemma 2. RDP for Gaussian mechanism: Gaussian mechanism is the basic mechanism to achieve Rényi differential privacy. For a function $f : \mathcal{D} \rightarrow \mathbb{R}^k$ with sensitivity s, a mechanism F follows (α, ϵ)-RDP can be constructed by

$$F(x) = f(x) + \mathcal{N}(\sigma^2) \quad where \quad \sigma^2 = \frac{s^2 \alpha}{2\epsilon} \tag{5}$$

Lemma 3. From (α, ϵ)-RDP to (ϵ, δ)-DP: If $F(x)$ satisfies (α, ϵ)-RDP, then for any given $\delta > 0$, F satisfies (ϵ', δ)-differential privacy, where $\epsilon' = \epsilon + \frac{ln(1/\delta)}{\alpha - 1}$. The value of δ is generally taken as $\delta \leq \frac{1}{n^2}$.

RDP is more flexible and combines the parameters α and sensitivity Δf. RDP can select the appropriate parameter α according to different application scenarios. When $\alpha = 1$, RDP is equal to DP. When $\alpha > 1$ can provide stronger privacy protection. When the value of σ^2 is given, the sequential composition of the Rényi differential privacy is used to limit the privacy consumption of repeated applications of the Gaussian mechanism, and then convert the Rényi differential privacy to (ϵ, δ)-differential privacy. The total privacy consumption obtained by this procedure is usually much lower than that obtained by directly applying the sequential composition in (ϵ, δ)-differential privacy. Based on this property, this paper also uses Rényi differential privacy to implement parameters perturbation in local datasets.

IV. METHOD DESIGN

This section focuses on a new federated privacy enhancement architecture to achieve client-level privacy protection, low communication overhead, and high robust aggregation. We begin with an overview of our privacy protection FL model. Then the proposed scheme is described and analyzed in detail.

A. System model

Our aim is to devise a privacy-centric and robust federated learning framework tailored for cross-silo settings. Our specific objectives are outlined as follows:

- Communication: We have designed an adaptive parameter upload rate adjustment method based on Top-K, which tightens the lower bound of local information compression rate, effectively reducing communication costs.
- Robustness: We propose an importance-weighted aggregation method, significantly enhancing the robustness of global model aggregation under differential privacy with noised parameters.
- Privacy: We strive to achieve client-level differential privacy by employing a dynamic privacy budget adjustment mechanism. This approach facilitates a judicious balance between privacy and accuracy.

The proposed architecture executes an iterative process, comprising the following steps: 1) The server broadcasts the initialized model, the current round's upload rate, and the privacy budget to all local clients. Each client then performs local stochastic gradient descent using their local data to obtain updated local weight differences. 2) To reduce communication costs, Top-K parameter sparsification is performed based on the given upload rate. 3) To protect client privacy, Gaussian noise is introduced to perturb the sparsified model parameters, based on the given privacy budget. 4) The noised weight difference parameters are uploaded to the server. 5) The server performs weighted aggregation of the uploaded model parameters, considering factors such as each client's data volume, upload rate, and parameter reliability, to obtain a new global model. Additionally, the server dynamically adjusts the privacy budget for the next round based on the performance of the global model and assesses the communication capabilities of local devices by analyzing the time delay in node parameter uploads, thereby dynamically adjusting the parameter upload rate for each client in the next round. The entire process is illustrated in Fig. 2. Our overall cross-silo federated learning training scheme's pseudo-code is given in Algorithm 1.

B. Dynamic privacy budget adjustment

The primary objective of dynamic privacy budget allocation is to achieve a delicate balance between model accuracy and data privacy. We use differential privacy to protect privacy by adding carefully designed noise to parameters before they are uploaded to the server. The lower the privacy budget, the greater the noise amount and the higher the privacy degree, but the higher the parameter distortion degree, thus reducing the accuracy. The privacy budget of the conventional differential privacy algorithm is fixed in each round. For example, the privacy budget is set too small. According to equation 3, the smaller ϵ is, the larger σ^2 is. Large noise will make useful information drowned out, resulting in a serious impact on accuracy. In order to achieve a balance between accuracy and privacy, the privacy budget needs to be dynamically adjusted for each round based on the actual performance of the model. Therefore, the server adapts the privacy budget for the next round based on the accuracy performance of the global model in the current round. The adjustment principle involves increasing the privacy budget of the next round if the model accuracy in the current round falls below the expected effect.

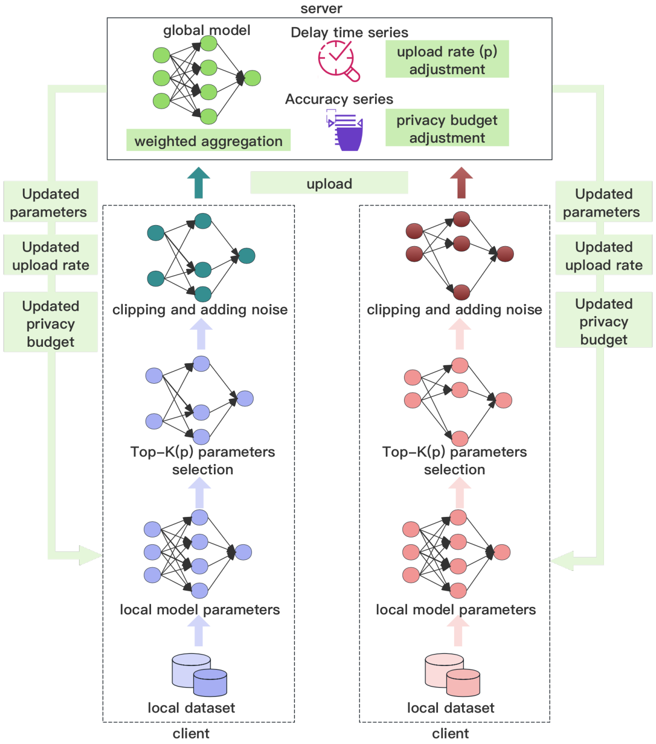


Fig. 2. The Overall Process of Our Work.

Conversely, if the model accuracy of the current round exceeds the expected effect, the privacy budget for the next round is reduced. Importantly, the expected model accuracy is defined as a range of values rather than a single value, providing flexibility in the adjustment process to accommodate various scenarios. The specific adjustment scheme is as follows:

Step_1: The change in accuracy value of the global model in time window of round t and round $t-1$ is $\Delta acc_{t-1} = acc_t - acc_{t-1}$. If $\Delta acc_{t-1} < 0$, it means that the accuracy of the model at the end of the t-th training round has decreased instead of increased. In this case, the next round should add less noise, and ϵ_{t+1} should be larger. Assuming that the amount of noise is desired to be reduced by at least c, we can use the following formula (6):

$$\epsilon_{t+1} \leq \epsilon_t - c \tag{6}$$

Step_2: If $\Delta acc_{t-2} - \Delta acc_{t-1} \geq d, (\Delta acc_{t-2} > 0, \Delta acc_{t-1} > 0)$, then it means that the model accuracy at the end of the t-th training round has increased but the effect is very little. Then the next round should add less noise, and ϵ_{t+1} should be larger. In this way, the same result as (6).

Step_3: If $\Delta acc_{t-1} - \Delta acc_{t-2} \geq l, (\Delta acc_{t-1} > 0)$, then it means that the accuracy of the model is improved more at the t-th training round, and stronger protection of privacy can also be implemented while ensuring the training is carried out properly. Then the noise should be increased in the next round, and ϵ_{t+1} should be smaller. Similarly, assuming that the amount of noise is desired to increase by at least c, we

Algorithm 1 Algorithm of the proposed model

```
Input: $x_1, ..., x_{n_k}$
Server exectues:
random initialize $w_0$, $p_0$ and $\epsilon_0 \leftarrow \frac{\epsilon}{T}$
for each round $t = 0, 1, 2, ..., T-1$ do
    for each client $k \in K$ do
        $w_t^k \leftarrow ClientUpdate(k, w_t, p_t^k, \epsilon_t, \alpha)$
    end for
    $w_{t+1} \leftarrow w_t + \sum_{k=1}^{K} Imp_k \cdot w_t^k$ // Aggregation
    $acc_t$ ← (model accuracy on auxiliary dataset with $w_{t+1}$)
    if $t >= 1$ then // Upload rate adjustment
        compute $\{\overline{d_1}, ..., \overline{d_k}\}$
        $p_{t+1}$ ← (adjust based on $\{\overline{d_1}, ..., \overline{d_k}\}$)
    end if
    if $t >= 2$ then // Privacy budget adjustment
        $\Delta acc_{t-1} = acc_t - acc_{t-1}$
        $\Delta acc_{t-2} = acc_{t-1} - acc_{t-2}$
        $\epsilon_{t+1}$ ← (adjust based on $\Delta acc_{t-1}$ and $\Delta acc_{t-2}$)
    end if
end for

ClientUpdate $(k, w, p^k, \epsilon, \alpha)$: // Run on client $k$
for each local epoch $e$ from 0 to $E-1$ do
    $\mathcal{B}$ ← (randomly selected $B$ samples from $n_k$)
    compute stochastic gradient $\nabla\mathcal{L}(w_e, \mathcal{B}_e)$ with $\mathbb{E}[\nabla\mathcal{L}(w_e, \mathcal{B}_e)] = \nabla\mathcal{L}(w_e)$ // Compute gradient
    $w_{e+1} \leftarrow w_e - \eta_l^k \nabla\mathcal{L}(w_e, \mathcal{B}_e)$
end for
$\Delta w_E = w_E - w$
$\Delta w'_E \leftarrow TopK(\Delta w_E, p^k)$ // Parameters selection
$\Delta \bar{w}_E \leftarrow \Delta w'_E \cdot \min(1, \frac{C}{\|\Delta w'_E\|})$ // Clip gradient
$\Delta \tilde{w}_E \leftarrow \Delta \bar{w}_E + z^k$ // Add noise
return $\Delta \tilde{w}_E$ to server
```

have

$$\epsilon_{t+1} \geq \epsilon_t - c \tag{7}$$

Step_4: In the remaining cases, ϵ_{t+1} is the average of the remaining privacy budget, i.e. $\epsilon_{t+1} = \frac{\epsilon - \sum_i^t \epsilon_i}{T-t}$.

Where, threshold c is used to control the magnitude of the adaptive privacy budget adjustment. A higher c value means a greater change in the privacy budget, which can lead to great instability in the overall training process. Therefore, it is recommended to select $\frac{1}{\vartheta}$ of the mean privacy budget, where ϑ is a positive integer. The thresholds d and l determine when the privacy budget is adjusted. When the value of d and l is set larger, it means that the tolerance of instability in the training process is greater, and the number of adaptive adjustments in the training process is less. The specific values of c, d, l in the above cases need to be adjusted according to the actual situation and a relatively suitable value is found according to the experimental situation. At the end of T rounds of training, The overall is satisfied by (α, ϵ)-RDP, and satisfied by $(\epsilon + \frac{ln(1/\delta)}{\alpha - 1}, \delta)$-DP.

C. Adaptive upload rate adjustment Top-K

The more the communication parameters, the more noise differential privacy needs to add for the same level of privacy, but reduced communication can lead to model fuzziness. TOP-K is a widely used parameter selection mechanism to reduce communication cost. K is a predefined threshold which balances the training accuracy and communication cost. Typically, the value of K is fixed throughout the training process. In each training round, the local client trains local model using its local data and selects the maximum K parameters to upload to the sever, instead of uploading all the parameters. Its superiority has been theoretically proven [33], making it a favorable choice to reduce communication costs in this paper. In cross-silo federated learning, there exist variations in device computing power and communication bandwidth among participants. When a participant's device has limited computing power and communication bandwidth, it may experience prolonged local computation and communication times, or even face challenges in uploading parameters. In such cases, uploading a reduced number of parameters can effectively reduce communication costs. Conversely, uploading more parameters can help the global model better adapt to local data and improve the accuracy of federated learning. However, in the previous work, the threshold K is usually fixed which on the one hand does not give full play to the role of clients with good communication conditions, and on the other hand limits the lower bound of local information compression rate, thereby increasing the communication cost to a certain extent. Specifically, when the K value is too small, the client with good communication conditions can upload more parameters without affecting the training progress to make the local model more accurate, but it can only waste part of the bandwidth due to the fixed K value. When the value of K is too large, the client with poor communication conditions will need longer communication time to complete the interaction and slow down the training progress. Therefore, it becomes imperative to dynamically adjust the parameter uploading ratio based on the actual situation of participating devices, ensuring the efficiency and accuracy of RDP federated learning while optimizing communication costs.

The device capabilities can be evaluated based on the time latency of local nodes, and the parameter upload rate for the next round can be determined accordingly. Local nodes with low time latency are generally considered to have better communication and computation capabilities, while those with high time latency are considered to have lower capabilities. The specific process is shown below:

Step_1: Set the initial parameter upload rate p_0.

Step_2: The server records the parameter upload time for each participant k in the past r rounds separately, $\{d_k^{t-r}, ..., d_k^t\}$. Then the average upload duration is $\overline{d_k} = \frac{\sum_j^r d_k^j}{r}$.

Step_3:

$$\begin{cases} p_k^{t+1} = p_k^t - \Delta p & \text{if } \overline{d_k} \in max_{\lceil \varrho \times K \rceil} \overline{d_k} \\ p_k^{t+1} = p_k^t + \Delta p & \text{if } \overline{d_k} \in min_{\lceil \varrho \times K \rceil} \overline{d_k} \end{cases} \tag{8}$$

Where ϱ is the percentage of the number of clients that need to adjust. It is necessary to ensure $0 < p_k^{t+1} < 1$, otherwise $p_k^{t+1} = p_k^t$.

Step_4: The $p_k^{t+1} \times n$ largest parameters need to be selected for upload, where n is the total number of model parameters.

D. Weighted Aggregation based on Importance

The conventional weighted aggregation method determines weights based on the data quantity of local nodes. However, we propose that besides data quantity, the reliability of the uploaded parameters by participants should also be considered. The addition of differential privacy noise and external malicious attackers will reduce the accuracy of the aggregation results. The server evaluates the reliability of the received parameters before aggregation, and assigns lower weights to the less reliable parameters to reduce the impact of loud noise and malicious parameters on the model accuracy.

Parameter credibility is assessed based on the similarity between two consecutive rounds of parameters from a node and the similarity with the global parameters. Due to the large number of parameters and fine-tuning based on previous training, parameters from adjacent rounds usually exhibit similar orientations and magnitudes. Therefore, the local parameter similarity between two consecutive parameter uploads by a node can serve as a measure of upload confidence. Moreover, nodes significantly contributing to global parameters generally have similar directions and magnitudes compared to the global parameters. Consequently, the similarity between local parameters uploaded by a node in the current round and the global parameters from the previous round can be used to assess the trustworthiness of parameter uploads.

In addition, considering the dynamic adjustment of parameter upload rate, usually a client with a high parameter upload rate means a good hardware facility, so it should increase the contribution of its local model to the global model.

Global model weighted aggregation is determined by considering three key factors: the amount of node data, the parameter upload rate, and parameter credibility.

Step_1: Calculate the parameter credibility $Cied_k$ of node k. According to (10) we can calculate $cos(\Delta w_k^{t-1}, \Delta w_k^t)$ and $cos(\Delta w_k^t, \Delta w^{t-1})$ respectively, then we have

$$cied_k = \beta\, cos(\Delta w_k^{t-1}, \Delta w_k^t) + (1-\beta) cos(\Delta w_k^t, \Delta w^{t-1}) \quad (9)$$

Where $0 < \beta < 1$. Nodes with a larger amount of data typically have a greater impact on the global model, and as a result, their local weights uploaded in each round are likely to be more similar to the global weights from the previous round. Therefore, we suggest setting a higher weight value $(1-\beta)$ for these nodes.

The similarity of vectors A, B is calculated by the cosine similarity. That is

$$cos(A,B) = \frac{A \cdot B}{\parallel A \parallel \parallel B \parallel} \quad (10)$$

where A and B denote the parameter vectors, $\cdot$ denotes the dot product operation of the vectors, and $\parallel A \parallel$ and $\parallel B \parallel$ denote the L2 Norm of A and B. The cosine similarity ranges from -1 to 1, where a value closer to 1 indicates a higher similarity in direction between the two vectors. Conversely, a value closer to -1 indicates a higher dissimilarity in direction, while a value closer to 0 indicates a greater difference between the directions of the two vectors. As we aim to measure the similarity in direction between the two parameter vectors, we consider the case of low similarity as the opposite direction.

$$\begin{cases} cos(A,B) = 0 & if\ cos(A,B) \le 0 \\ cos(A,B) = cos\_sim(A,B) & if\ cos(A,B) > 0 \end{cases} \quad (11)$$

Step_2: Calculate the importance score Imp_i^t based on the amount of data, the parameter upload rate and parameter credibility of node i:

$$Imp_k^t = \gamma_1 \frac{n_k}{\sum_k^K n_k} + \gamma_2 \frac{p_k^t}{\sum_k^K p_k^t} + \gamma_3 \frac{cied_k}{\sum_k^K cied_k} \quad (12)$$

Where n_k is the amount of data for each client. $0 < \gamma_1, \gamma_2, \gamma_3 < 1$, and $\gamma_1 + \gamma_2 + \gamma_3 = 1$. These three parameters respectively determine the proportion of local data volume, parameter upload rate and parameter credibility in aggregation. The amount of data plays a crucial role in the training of neural network models, which determines the upper limit of the accuracy of local model training to some extent. Therefore, in the case of data heterogeneity, the amount of local data is closely related to the contribution of the local model to the global model, and $\gamma_1$is usually the largest of the three parameters.

Step_3: Global parameter weighted aggregation

$$w^{t+1} = w^t + \sum_{k=1}^{m} \frac{Imp_k^t}{\sum_k^K Imp_k^t} \cdot \Delta w_k^t \quad (13)$$

E. Privacy Analysis

Assuming a total privacy budget of ϵ, with T total training rounds, and a privacy budget per round of ϵ_t. For a given privacy budget, the Rényi Differential Privacy (RDP) can select an appropriate parameter α such that the conversion of RDP to Differential Privacy (DP) minimizes the privacy budget. Therefore, in each round, different clients k satisfy (α_t^k, ϵ_t)-RDP. According to Lemma 3, it can be converted to DP as (ϵ_t', δ)-DP. Given the parallel composition property of DP, each round satisfies $(\max(\epsilon_t'), \delta)$-DP. Following the sequential composition property of DP, after T rounds, it satisfies $\left(\sum_{t=1}^T \max(\epsilon_t'), T\delta\right)$-DP, where $\epsilon_t' = \epsilon_t + \frac{\ln(1/\delta)}{\alpha_t^k - 1}$.

Since $\alpha_t^k \in [2, 100]$, when $\alpha_t^k = 2$, ϵ_t' attains its maximum value, which is $\epsilon_t + \ln(1/\delta)$, leading to

$$\sum_{t=1}^T \max(\epsilon_t') \le \sum_{t=1}^T [\epsilon_t + \ln(1/\delta)] = T\ln(1/\delta) + \epsilon \quad (14)$$

When $\alpha_t^k = 100$, ϵ_t' attains its minimum value, which is $\epsilon_t + \frac{\ln(1/\delta)}{99}$, leading to

$$\sum_{t=1}^T \max(\epsilon_t') \ge \sum_{t=1}^T [\epsilon_t + \frac{\ln(1/\delta)}{99}] = \frac{T}{99}\ln(1/\delta) + \epsilon \quad (15)$$

Let $T\ln(1/\delta) = \mu$, then there is

$$\frac{\mu}{99} + \epsilon \le \epsilon' \le \mu + \epsilon \quad (16)$$

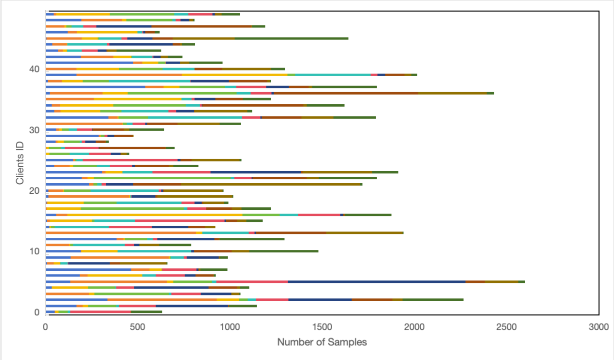


Fig. 3. Non-IID setting of 50 clients by $q \sim \text{Dirichlet}(0.5)$. Where different colors represent different class labels.

V. Experimental results

In this section, we present a series of experiments designed to evaluate the effectiveness of our proposed approach. We aim to demonstrate the impact of the adaptive upload rate adjustment method. Additionally, we investigate the effectiveness of the importance-based global model parameter aggregation method on model convergence. Lastly, we analyze the trade-off between model accuracy and data privacy in the context of the noisy Rényi differential privacy (RDP) based approach.

A. Experimental setup

We evaluated our proposed method on the widely used MNIST and CIFAR-10 datasets, which is a standard benchmark in the field of deep learning. The MNIST dataset consists of hand-written digits represented as 28x28 images. To preprocess the dataset, we performed normalization to center the digits within the images. The dataset contains 60,000 training examples and 10,000 test examples. The CIFAR-10 dataset contains 50,000 training samples and 10,000 test samples, all of which are 32 × 32 color images. For a cross-silo federated learning setting, the number of clients typically ranges from 2 to 100. To demonstrate the generalizability of our scheme, we set the number of clients for the MNIST dataset to 5, 50, and 100, and for the CIFAR-10 dataset to 5 and 50. We assume that the total number of samples in the dataset is fixed and there is no overlap in the samples distributed among clients. Therefore, the more clients there are, the less data each client receives. Our experiments include two distribution configurations: IID (Independent and Identically Distributed) and Non-IID (Non-Independent and Identically Distributed). In the IID setting, the local data held by clients are independent of each other but follow the same distribution. For the Non-IID setting, the label distribution $q \sim \text{Dirichlet}(\tau)$ where $\tau = 0.5$. The example diagram is shown in the Fig. 3.

For our experiments, we employed CNN model architectures, but the model architecture is different for the two datasets. The details are shown in Table III and IV. The experiments were conducted on a computer with a 3.70GHz CPU and a GeForce GTX 1080 Ti GPU, operating on the Linux system. We implemented the experiments using the pytorch package in the Python programming language. With this setup, we conducted comprehensive evaluations to assess the performance of our proposed method on the MNIST and CIFAR-10 datasets within the federated learning framework.

TABLE III
CNN model architecture for MNIST dataset.

Layer	Parameters
Convolution	16 filters of 8x8, strides 2
Max-Pooling	2x2
Convolution	32 filters of 4x4, strides 2
Max-Pooling	2x2
Fully connected	32 units
Softmax	10 units

TABLE IV
CNN model architecture for CIFAR-10 dataset.

Layer	Parameters
Convolution	32 filters of 3x3
Max-Pooling	2x2, stride 2
Convolution	32 filters of 3x3
Max-Pooling	2x2, stride 2
Convolution	32 filters of 3x3
Max-Pooling	2x2, stride 2
Fully connected	128 units
Fully connected	10 units

In this section, we evaluate the performance of Adaptive Top-K (Atop-K), Importance Weighted Aggregation (ImpWA), and Adaptive RDP (ARDP) methods by comparing them with the following baselines:

- FedAvg [2]: This serves as the classical baseline in Federated Learning (FL) algorithms, which does not take privacy issues into account.
- FedProx [36]: An advanced approach to address data heterogeneity in federated learning, enhancing the aggregation effect by incorporating regularization into the local loss function. We set the regularization value to 0.01.
- FedAvg-topk: A communication-efficient variant of FedAvg, where each client's local model updates are sparsified using top-k before uploading, and the value of K remains fixed throughout the training process.
- FedProx-topk: A communication-efficient variant of FedProx, similar to FedAvg-topk, where each client's local model updates are sparsified using top-k before uploading, with a constant K value during training.
- DP-FedAvg [45]: The state-of-the-art differential privacy variant of FedAvg, capable of implementing client-level DP.
- PEDPFL [35]: A cutting-edge differential privacy scheme, introducing a classifier-disturbance regularizer incorporated into the objective function. We set the regularization value to 0.5.

Without special instructions, the model hyperparameters are set as follows: $T = 50$, $E = 3$, $\eta_l = 0.01$, $\delta = 1e-5$ and $\beta = 0.5, \gamma_1 = 0.3, \gamma_2 = 0.2$. The local batchsize for MNIST model is 64, and CIFAR-10 is 32. Because there is oscillation in the model training process, we take the average of the last 5 rounds as the result of one experiment. In addition, in order to eliminate the influence of randomness on the results, we conducted 5 experiments for each configuration and took the average value as the final result.

B. Atop-K for Communication

In this section, we focused on analyzing the performance of adaptive upload rate adjustment of Top-K. Specifically, we mainly verify the effect of Atop-K method on the communication cost, and compares the effectiveness of Atop-K method compared with top-K method when the initial K (p_0) value is 1.0, 0.5 and 0.1, respectively.

TABLE V
COMMUNICATION AND TEST ACCURACY OF 5 CLIENTS IN MNIST DATASET.

Aggregation	top-p_0	IID		Non-IID	
		ACC(%)	up98	ACC(%)	up98
FedAvg	topk-1.0	98.45	87	98.49	76
	topk-0.5	98.49	49.50	98.53	40.50
	topk-0.1	98.02	20.7	98.08	17.40
	Atopk-1.0	98.50	75.09	98.52	71.18
	Atopk-0.5	98.52	44.5	98.52	38
	Atopk-0.1	98.15	19	98.21	15.98
FedProx	topk-1.0	98.47	82	98.51	78
	topk-0.5	98.51	47.50	98.53	41.50
	topk-0.1	98.07	19.50	98.12	18.10
	Atopk-1.0	98.50	77.31	98.49	66.57
	Atopk-0.5	98.48	45	98.49	39.50
	Atopk-0.1	98.16	18.43	98.17	17.62
ImpWA	topk-1.0	98.51	86	98.58	76
	topk-0.5	98.53	47.50	98.43	41.50
	topk-0.1	98.05	19.60	98.10	17.50
	Atopk-1.0	98.53	80.87	98.54	68.42
	Atopk-0.5	98.59	40	98.57	36.50
	Atopk-0.1	98.17	17.78	98.27	14.21

TABLE VI
COMMUNICATION AND TEST ACCURACY OF 50 CLIENTS IN MNIST DATASET.

Aggregation	top-p_0	IID		Non-IID	
		ACC(%)	up95	ACC(%)	up95
FedAvg	topk-1.0	95.90	1800	96.33	1470
	topk-0.5	95.92	885	96.14	820
	topk-0.1	94.5	249	94.69	248
	Atopk-1.0	96.15	1472.86	96.39	1335.95
	Atopk-0.5	96.05	810	96.15	790
	Atopk-0.1	94.57	264.75	95.29	236.93
FedProx	topk-1.0	95.74	1860	96.34	1480
	topk-0.5	95.90	880	96.23	755
	topk-0.1	94.71	245	94.76	246
	Atopk-1.0	96.06	1564.73	96.37	1362.91
	Atopk-0.5	96.02	825	96.33	750
	Atopk-0.1	94.58	260.29	95.10	251.16
ImpWA	topk-1.0	95.96	1690	96.35	1530
	topk-0.5	95.80	920	96.06	815
	topk-0.1	94.26	247	94.56	246
	Atopk-1.0	96.21	1575.22	96.39	1400.5
	Atopk-0.5	96.08	825	96.25	755
	Atopk-0.1	94.87	255.92	95.1	237.244

TABLE VII
COMMUNICATION AND TEST ACCURACY OF 100 CLIENTS IN MNIST DATASET.

Aggregation	top-p_0	IID		Non-IID	
		ACC(%)	up93	ACC(%)	up93
FedAvg	topk-1.0	93.73	4140	94.13	3860
	topk-0.5	93.44	2240	94.05	1900
	Atopk-1.0	93.64	4119.82	94.57	3330.57
	Atopk-0.5	93.35	2220	94.48	1700
FedProx	topk-1.0	93.59	4300	94.06	3900
	topk-0.5	93.36	2200	94.44	1720
	Atopk-1.0	93.84	3901.49	94.34	3588.81
	Atopk-0.5	93.58	2120	94.63	1640
ImpWA	topk-1.0	93.68	4060	93.97	3820
	topk-0.5	93.28	2210	94.20	1860
	Atopk-1.0	93.92	3731.90	94.41	3469
	Atopk-0.5	93.64	2079.96	94.54	1699.99

TABLE VIII
COMMUNICATION AND TEST ACCURACY OF 5 CLIENTS IN CIFAR-10 DATASET.

Aggregation	top-p_0	IID		Non-IID	
		ACC(%)	up70	ACC(%)	up70
FedAvg	topk-1.0	72.28	107	71.28	94
	topk-0.5	72.03	57.40	71.37	47.50
	topk-0.1	70.64	19.80	70.36	15.20
	Atopk-1.0	72.33	100.09	71.40	91.53
	Atopk-0.5	72.41	53	71.63	43.50
	Atopk-0.1	71.05	19.34	70.56	14.24
FedProx	topk-1.0	72.09	105	71.55	95
	topk-0.5	71.98	59	71.68	51
	topk-0.1	71.03	18	71.13	15
	Atopk-1.0	72.65	100.09	71.64	85.96
	Atopk-0.5	72.75	54.50	71.77	47
	Atopk-0.1	71.16	20.65	71.16	17.17
ImpWA	topk-1.0	71.82	112	72.40	97
	topk-0.5	72.10	54.50	72.21	49.50
	topk-0.1	70.75	19	71.57	15.80
	Atopk-1.0	71.17	100.93	73.44	93.42
	Atopk-0.5	72.52	53	72.13	48.50
	Atopk-0.1	71.22	18.02	71.83	19.54

TABLE IX
COMMUNICATION AND TEST ACCURACY OF 50 CLIENTS IN CIFAR-10 DATASET.

Aggregation	top-p_0	IID		Non-IID	
		ACC(%)	up50	ACC(%)	up50
FedAvg	topk-1.0	50.71	2290	52.82	1920
	topk-0.5	50.59	1135	52.89	960
	Atopk-1.0	50.74	2106.62	52.56	1876.91
	Atopk-0.5	50.74	1109.84	53.16	930.10
FedProx	topk-1.0	51.16	2180	52.34	2010
	topk-0.5	49.14	1140	52.54	970
	Atopk-1.0	51.08	2060.47	52.43	1887.81
	Atopk-0.5	50.56	1114.88	52.98	925
ImpWA	topk-1.0	50.84	2250	51.86	2070
	topk-0.5	50.55	1145	51.45	1060
	Atopk-1.0	51	2079.01	51.65	1991.72
	Atopk-0.5	50.72	1115	51.96	1015.01

Columns "up98", "up95" and "up93" in Table V, VI and VII show the experimental results of communication cost comparison when the number of clients is 5, 50 and 100, respectively,

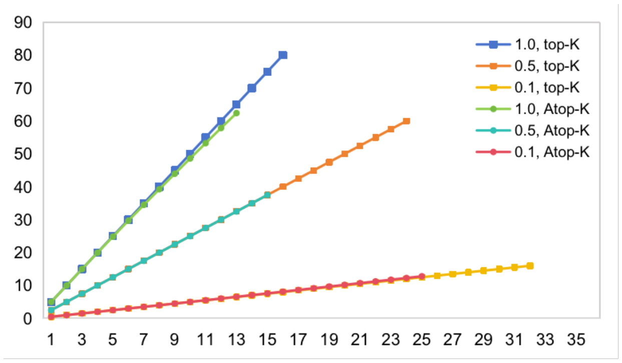


Fig. 4. Communication cost comparison of our upload rate adjustment method when there are 5 clients on the MNIST dataset. In this figure, we set the initial upload rate to 1.0, 0.5, and 0.1, respectively, to compare the communication bits needed to reach 98% accuracy with and without the dynamic upload rate adjustment method proposed in this paper.

on the MNIST dataset. That is, the communication costs required to achieve 98%, 95% and 93% accuracy, respectively. Columns “up70” and “up50” in Table VIII and IX show the experimental results of communication cost comparison when the number of clients is 5 and 50, respectively, on the CIFAR-10 dataset. That is, the communication costs required to achieve 70% and 50% accuracy, respectively. The lower the upload rate, the higher the degree of sparsification. At an upload rate of 1.0, $\Delta p = 0.05$ and $\varrho = 0.2$; at upload rates of 0.5 and 0.1, $\Delta p = 0.02$ and $\varrho = 0.1$. The values in the table represent the number of model parameters.

We measure communication costs in the number of bits transmitted. A model parameter is 32 bit. For the MNIST dataset, with a total of 10,650 model parameters, a value of 40 corresponds to a communication cost of $40 \times 10,650 \times 32$ bit $= 13,632,000$ bits. For the CIFAR-10 dataset, with a total of 86,346 model parameters, the same value corresponds to a communication cost of $40 \times 86,346 \times 32$ bit $= 110,522,880$ bits. For ease of comparison and analysis, only the number of units is shown in the tables. From these tables, it can be seen that after adjustment using our proposed adaptive method, the communication costs are reduced at upload rates of 1.0 and 0.5, achieving a compact lower bound of the compression rate. Compared to no sparsification, it saves 86% of communication costs, and compared to non-adaptive sparsification, it further saves 18% of communication costs. At an upload rate of 0.1, the communication cost increases, but the accuracy improves after the same 50 epochs training. This indicates that with a fixed Top-K threshold, when set too low, more rounds are needed to achieve the same accuracy, giving an illusion of reduced communication costs. Adaptive adjustment of the upload rate changes this by increasing the per-round communication cost but reducing the number of training rounds.

Fig. 4. shows the cumulative communication overhead of reaching 98% test accuracy when the number of clients is 5 on the MNIST dataset, among which the communication overhead of Atop-K is significantly reduced. It is further proved that the proposed method is effective in the minimization of sparse communication cost.

Columns ”ACC” in Table V, VI and VII show the test accurancy after training 50 epochs when the number of clients is 5, 50 and 100, respectively, on the MNIST dataset. Columns ”ACC” in VIII and IX show test accuracy after training 50 epochs when the number of clients is 5 and 50, respectively, on the CIFAR-10 dataset.

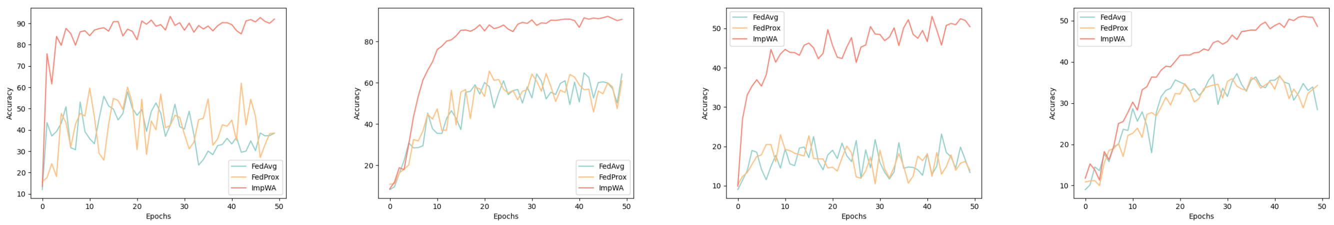
As for Table V-IX, when the same aggregation method is the same initial upload rate, the values of top-K and Atop-K are mainly observed in the ”up” column. It can be seen that the value of Atop-K is smaller than top-K in most cases, which indicates that the communication cost is reduced and the lower limit of communication is more compact through adaptive upload rate adjustment. This phenomenon is especially obvious when p_0 is 1.0 and 0.5, because the parameter upload rate at this time still has room to decrease relative to the acceptable parameter distortion of the model. However, when p_0 is 0.1, the number of uploaded parameters is small, which makes the distortion degree of the local model relatively large. Although the communication cost is saved, the accuracy of the model is sacrificed. The proposed Atop-K method aims to restrain the communication cost while ensuring the accuracy, so the communication cost of Atop-0.1 is slightly higher than top-0.1, and the accuracy of Atop-0.1 is usually higher than top-0.1.

It should be noted that there is no necessary relationship between the reduction of communication costs and accuracy. The reason for this increase in accuracy is that according to our communication cost measure, a reduction in the communication cost of reaching a certain accuracy before 50 rounds means that fewer rounds have reached the same accuracy, which also means that there are more rounds remaining from 50 rounds. Therefore, there is a high probability that the accuracy will be improved, but there will be accidents when the training state of the model itself is unstable. For example, when the number of clients is 50 or 100, the number of samples each client has becomes less, and the training effect of the local model is not good. At this time, the training process of the model will introduce great instability, and the accuracy of the model will be reduced in a certain probability.

C. *ImpWA for Robustness*

In this section, we verify the robustness of the proposed weighted aggregation method. As can be seen from Fig. 1, robustness is related to communication, so the ImpWA method is compared with the classical FedAvg [2] and FedProx [35] method under different communication Settings (Atop-K and top-K, where K (p_0) values are 1.0, 0.5 and 0.1, respectively). In addition, robustness is directly related to the amount of noise, which is divided into predictable noise (Gaussian noise in differential privacy) and unforeseeable noise (poison attacks by malicious attackers). Unforeseen noise has a particularly strong impact on model robustness, so this section verifies the performance of ImpWA, FedAvg, and FedProx methods in defending against model poisoning attacks.

From the results of the Table V-IX, it can be seen that in most cases, ImpWA achieves the highest accuracy compared to FedAvg and FedProx when there is adaptive upload rate adjustment. This is because the ImpWA method takes into account the parameter upload rate. With the dynamic adjustment



(a) MNIST, 5 clients (b) MNIST, 50 clients (c) CIFAR-10, 5 clients (d) CIFAR-10, 50 clients

Fig. 5. Test accuracy performance on MNIST and CIFAR-10 datasets with Non-IID distribution for different aggregation methods in response to model poisoning attacks.

TABLE X
TEST ACCURACY AGAINST MODEL POISONING ATTACKS OF 5 CLIENTS IN MNIST DATASET.

Aggregation	top-p_0	IID	Non-IID
FedAvg	topk-1.0 topk-0.5 topk-0.1	76.56 78.75 86.71	38.87 39.03 56.10
FedProx	topk-1.0 topk-0.5 topk-0.1	77.17 74.66 85.58	37.54 37.15 58.57
ImpWA	topk-1.0 topk-0.5 topk-0.1	94.62 95.36 95.96	90.14 90.95 92.59

TABLE XI
TEST ACCURACY AGAINST MODEL POISONING ATTACKS OF 50 CLIENTS IN MNIST DATASET.

Aggregation	top-p_0	IID	Non-IID
FedAvg	topk-1.0 topk-0.5 topk-0.1	84.06 85.29 84.82	63.42 63.19 69.10
FedProx	topk-1.0 topk-0.5 topk-0.1	83.77 83.89 85.22	63.68 63.47 69.78
ImpWA	topk-1.0 topk-0.5 topk-0.1	92.98 92.91 90.91	89.75 89.86 88.14

TABLE XII
TEST ACCURACY AGAINST MODEL POISONING ATTACKS OF 100 CLIENTS IN MNIST DATASET.

Aggregation	top-p_0	IID	Non-IID
FedAvg	topk-1.0 topk-0.5 topk-0.1	84.46 85.32 83.44	60.35 64.76 63.99
FedProx	topk-1.0 topk-0.5 topk-0.1	83.61 83.87 83.54	61.16 63.39 60.39
ImpWA	topk-1.0 topk-0.5 topk-0.1	91.53 90.60 88.24	86.85 88.68 80.70

TABLE XIII
TEST ACCURACY AGAINST MODEL POISONING ATTACKS OF 5 CLIENTS IN CIFAR-10 DATASET.

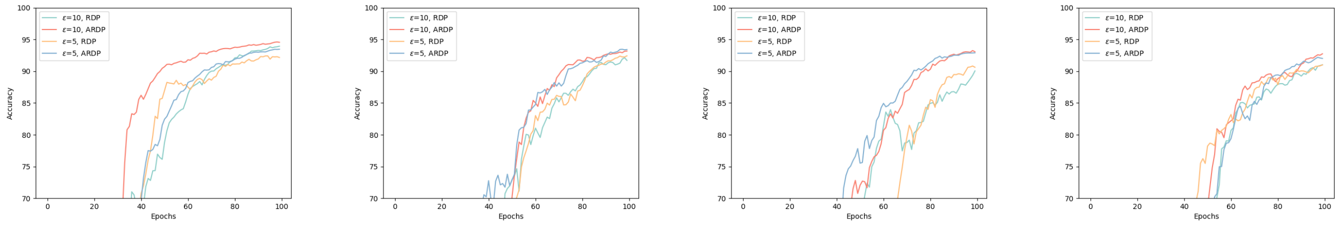
Aggregation	top-p_0	IID	Non-IID
FedAvg	topk-1.0 topk-0.5 topk-0.1	36.62 39.45 51.44	17.45 16.03 29.14
FedProx	topk-1.0 topk-0.5 topk-0.1	36.98 38.43 50.94	16.41 15.88 30.97
ImpWA	topk-1.0 topk-0.5 topk-0.1	62.44 62.79 63.10	50.23 52.17 57.38

TABLE XIV
TEST ACCURACY AGAINST MODEL POISONING ATTACKS OF 50 CLIENTS IN CIFAR-10 DATASET.

Aggregation	top-p_0	IID	Non-IID
FedAvg	topk-1.0 topk-0.5 topk-0.1	47.22 48.09 45.88	33.95 34.42 40.00
FedProx	topk-1.0 topk-0.5 topk-0.1	47.79 47.46 46.24	33.29 34.57 43.99
ImpWA	topk-1.0 topk-0.5 topk-0.1	52.18 51.11 47.17	49.59 49.29 45.92

of the upload rate, the weight of the aggregated local model is also dynamically adjusted, so as to achieve more efficient aggregation.

However, the original intention of our proposed aggregation method is to enhance the robustness of the federated learning architecture in the presence of parameter noise or external malicious attackers. Therefore, we use the most common model poisoning attack to verify the robustness of the ImpWA method. We simulate model poisoning attacks by modifying parameters to values sampled from a uniform distribution $(-0.25, 0.25)$. Table X -XII show the test accuracy of ImpWA in dealing with model poisoning attacks on the MNIST dataset. When the number of clients is 5, the number of attacked clients is 20% of the total; when the number of clients is 50 and



(a) DP-Fedavg, IID (b) DP-Fedavg, Non-IID (c) RDP-ImpWA, IID (d) RDP-ImpWA, Non-IID

Fig. 6. Test accuracy of DP-Fedavg and RDP-ImpWA with and without Adaptive Privacy Budget adjustment (ARDP) on the MNIST dataset for 5 clients. The overall privacy budget is 5 and 10, respectively.

100, the number of attacked clients is 40% of the total. Table XIII and XIV show the test accuracy of ImpWA in dealing with model poisoning attacks on the CIFAR-10 dataset. In these cases, the number of attacked clients is consistently 20% of the total. From the experimental results in these tables, it can be seen that the accuracy of the FedAvg and FedProx aggregation methods is severely affected, especially under the No-IID distribution, where the FedAvg and FedProx methods are almost completely ineffective. This is because these two methods only consider the local data volume factor when aggregating. Models with large amounts of local data normally contribute more to the global model, so when these local clients are malicious, the impact on the global model is also greater. The ImpWA method weakens the influence of data volume by multi-factor weighting, and evaluates the reliability of local model parameters, which can effectively reduce the influence of malicious parameters. Moreover, the higher the upload rate, the greater the impact. This is because the higher the upload rate, the higher the proportion of poisoned parameters, naturally leading to a greater impact.

The precision performance of the three aggregation methods against the model poisoning attack under the Non-IID distribution is shown in the Fig. 5a.-5d. We can intuitively see the robustness of our proposed ImpWA method.

D. ARDP for Privacy

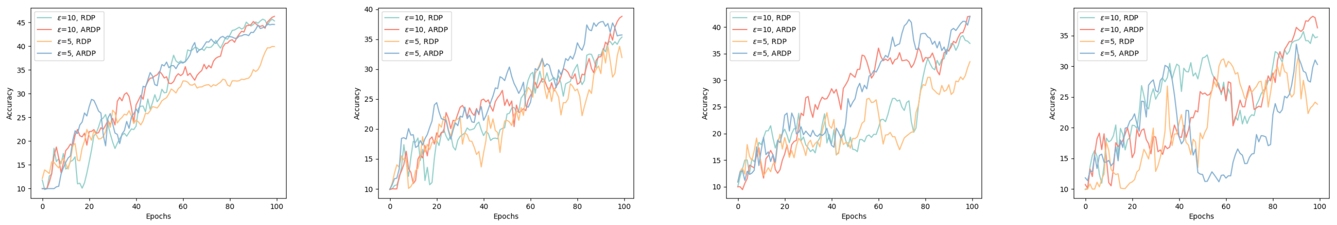
In this section, we validate the utility of the overall system model in terms of privacy. The privacy is mainly realized by ARDP method, and the effectiveness of ImpWA and Atop-K method has been verified in previous sections. Therefore, the system model adopted in this section is set as ImpWA and Atop-K (K (p_0) is 0.1). Compared with DP-FedAvg [44] and PEDPFL [34], both of which use FedAvg method for aggregation, the ImpWA method is also validated for reducing the impact of predictable noise on model robustness.

The total rounds is set 100. The clipping threshold C is set as the median of the parameters.In practical applications, the selection of parameters c, d, l is crucial for model training. we choose $c = \frac{\epsilon_{avg}}{50}, d = 1, l = 11$, where ϵ_{avg} represents the average remaining privacy budget.

The more noise is added, the greater the degree of distortion of the original model parameters. For (ϵ, δ)-DP, the addition of specific noise is determined according to formula 3. δ is usually set to a constant value, and s is also set to a constant value if the function and scenario are determined. Thus, the more stringent the privacy constraint, the smaller the ϵ value and the larger the σ^2 value, the greater the amount of noise added. For example, for $\epsilon = 1$, $\sigma^2 = 2s^2 ln(1.25/\delta)$. In the case of function and scenario determination, this value can be expressed by the constant c, that is, $\sigma^2 = c$ for $\epsilon = 1$. When $\epsilon = 5$, $\sigma^2 = \frac{c}{25}$. When $\epsilon = 10$, $\sigma^2 = \frac{c}{100}$. It can be seen that if the privacy constraint is 10 times stricter (to 1/10), the noise needs to be 100 times the original, the degree of distortion of the model parameters will be greatly increased, and the natural accuracy will be reduced. When $\epsilon = 1$, although the privacy is the strictest, the noise is too large, the accuracy is seriously decreased, and the model cannot be trained normally. Therefore, we choose $\epsilon = 5$ and 10 for the experiment.

Fig. 6a. and 6c. show the test accuracy on the MNIST dataset under IID distribution for DP-FedAvg and RDP-ImpWA with and without Adaptive Privacy Budget Adjustment (ARDP), respectively. Fig. 6b. and 6d. show the test accuracy under Non-IID distribution. Fig. 7a. and 7c. show the test accuracy on the CIFAR-10 dataset under IID distribution for DP-FedAvg and RDP-ImpWA with and without Adaptive Privacy Budget Adjustment (ARDP), respectively. Fig. 7b. and 7d. show the test accuracy under Non-IID distribution. The overall privacy budget is set at 5 and 10 respectively. From Fig. 6, it can be seen that ARDP effectively reduces the impact of noise on the training process while maintaining the same level of privacy protection. Noise in Fig. 7 has a relatively large impact on the training process, mainly because the training task for FICAR-10 is more complex than MNIST and is very sensitive to noise. But It can still be seen that the ARDP method is much more accurate.

Fig. 8. and 9. respectively show the performance of different privacy-enhanced architectures on the MNIST dataset under IID and non-IID distributions. And Fig. 10. and 11. respectively show the performance of different privacy-enhanced architectures on the CIFAR-10 dataset under IID and non-IID distributions. The effectiveness of the Atop-K and ARDP methods has already been discussed. To validate the performance of our proposed overall architecture, we set the initial upload rate to 0.1. Both DP-FedAvg and ARDP-ImpWA architectures employ Atop-K and ARDP settings. The PEDPFL architecture retains its original settings, i.e., top-K and RDP without adaptive adjustment. Fig. 8 and 9 clearly show that the PEDPFL architecture performs poorly. In Fig. 10 and 11,



(a) DP-Fedavg, IID (b) DP-Fedavg, Non-IID (c) RDP-ImpWA, IID (d) RDP-ImpWA, Non-IID

Fig. 7. Test accuracy of DP-Fedavg and RDP-ImpWA with and without Adaptive Privacy Budget adjustment (ARDP) on the CIFAR-10 dataset for 5 clients. The overall privacy budget is 5 and 10, respectively.

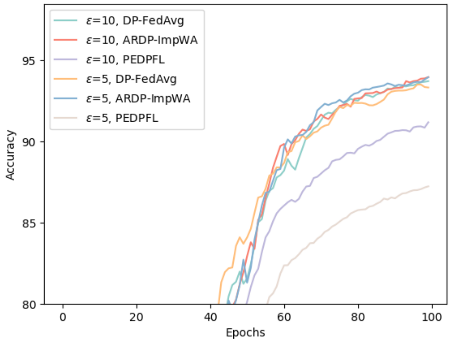


Fig. 8. Performance of different privacy enhancement architectures on MNIST datasets under IID distribution

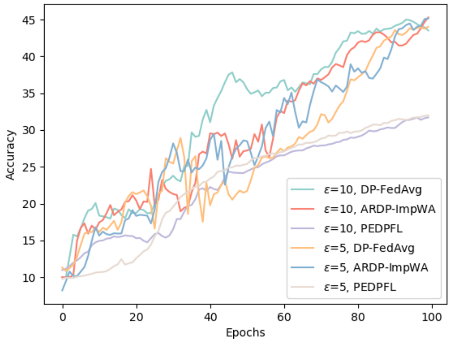


Fig. 10. Performance of different privacy enhancement architectures on CIFAR-10 datasets under IID distribution

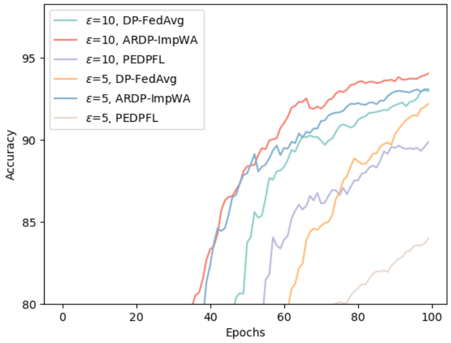


Fig. 9. Performance of different privacy enhancement architectures on MNIST datasets under Non-IID distribution

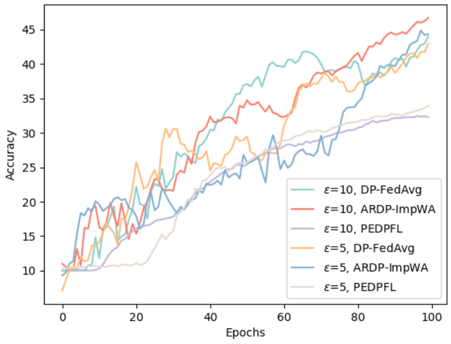


Fig. 11. Performance of different privacy enhancement architectures on CIFAR-10 datasets under Non-IID distribution

PEDPFL architecture makes the training process more stable through regularization, but the accuracy is obviously lower. In differential privacy, a smaller epsilon value means a larger amount of added noise, so generally speaking, the model performs better under the setting $\epsilon = 5$ than $\epsilon = 10$. The ARDP-ImpWA and DP-FedAvg shown in Fig. 8 to 11 both follow this rule. PEDPFL shows the same behavior in Fig. 8 and 9, but the opposite in Fig. 10 and 11. This is because the experiments in Fig. 10 and 11 are conducted on the CIFAR-10 dataset. When the training task is relatively complex, the instability of the training process is also conducive to the model generalization to some extent. However, the addition of regularization factors in the PEDPFL algorithm to force the training process to be stable makes the training process very stable and slow. As can be seen from the comparison results of Fig. 10 and 11, PEDPFL algorithm has obvious effect and low learning ability. In this case, the moderately large noise makes the training process more active, which helps the model improve the training speed.

While the advantage of the ARDP-ImpWA architecture is not very apparent under the IID setting, it shows a significant advantage under the Non-IID distribution. This is because both ARDP-IMPWA and DP-FedAvg use Atop-K and ARDP settings, so there are only differences in polymerization methods.

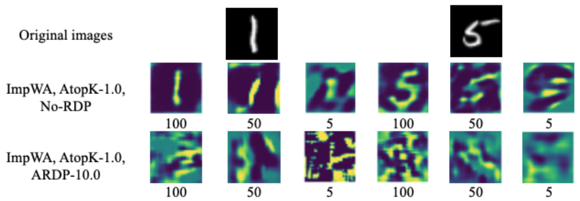


Fig. 12. Results of defending against GRNN privacy attacks on MNIST datasets

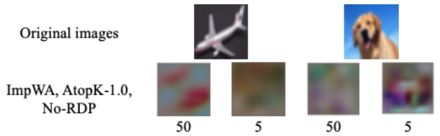


Fig. 13. Results of defending against GRNN privacy attacks on CIFAR-10 datasets.

If the data distribution is IID, the amount of data between clients is the same. In the absence of malicious attacks, the reliability of parameters among clients is roughly the same. The three factors in ImpWA approximate degenerate to only consider the parameter upload rate, so there is no obvious difference in the aggregation effect between ImpWA and FedAvg method under IID setting.

To further prove the effectiveness of the proposed federated privacy-enhanced architecture, we implemented a gradient leakage attack. Gradient leakage attacks can infer a client's local training samples and are currently the most significant privacy threat. GRNN [13] is the most advanced gradient leakage attack method, capable of recovering sample gradients in larger batchsize. Previous DLG [11] and iDLG [12] attacks were only effective for batchsize of 10 or even 1. Therefore, we used GRNN to validate our method. Gradient leakage attacks work by approximating the gradients uploaded by the client, which requires the locally uploaded parameters to be gradients. In our algorithm, the uploaded parameters are the differences in weights. According to formula (13), the parameters uploaded by client k can be considered as the negative of the gradients calculated by the global model with the client's sample size as the batch, where the weights can be considered as the learning rate of the global model. The second row of Fig. 12 shows the recovery of MNIST samples without differential privacy. From the figure, it can be seen that the recovery is more effective when there are more clients, as each client is allocated fewer samples, making the attack more successful. When the number of clients is 5, the attack is almost ineffective. This also shows that uploading the difference in weights inherently provides a certain level of privacy under the current level of privacy attacks. The third row of Fig. 12 shows the recovery of MNIST samples using the ARDP method with a privacy budget of 10, where the attack is completely ineffective. Fig. 13. shows the recovery of CIFAR-10 samples. Even in the absence of differential privacy settings, relying solely on the protection provided by the difference in weights is sufficient.

VI. Conclusion

This paper presents a federated privacy-enhancing algorithm that integrates local differential privacy, parameter sparsification, and weighted aggregation for cross-silo setting. The primary objective of this method is to enhance the privacy of the cross-silo federated learning model training process and to strike a balance between privacy, accuracy, communication cost and robustness. However, it is worth noting that the adjustment process may introduce some instability, and further refinements are required to achieve a more stable and robust training scheme in the future.

VII. Acknowledgement

This work is supported by the Shandong Science and Technology Major Project (2020CXGC10103) and National Natural Science Foundation of China (62101159).

References

[1] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321, 2015.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[3] M. Polato, R. Esposito, and M. Aldinucci, "Boosting the federation: Cross-silo federated learning without gradient descent," in *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, IEEE, 2022.

[4] C. Huang, J. Huang, and X. Liu, "Cross-silo federated learning: Challenges and opportunities," *arXiv preprint arXiv:2206.12949*, 2022.

[5] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, "Ffd: A federated learning based method for credit card fraud detection," in *Big Data–BigData 2019: 8th International Congress, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019, Proceedings 8*, pp. 18–32, Springer, 2019.

[6] S. R. Pfohl, A. M. Dai, and K. Heller, "Federated and differentially private learning for electronic health records," *arXiv preprint arXiv:1911.05861*, 2019.

[7] M. Yang, X. Wang, H. Qian, Y. Zhu, H. Zhu, M. Guizani, and V. Chang, "An improved federated learning algorithm for privacy preserving in cybertwin-driven 6g system," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 6733–6742, 2022.

[8] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pp. 603–618, 2017.

[9] J. Sun, A. Li, B. Wang, H. Yang, H. Li, and Y. Chen, "Soteria: Provable defense against privacy leakage in federated learning from representation perspective," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9311–9319, 2021.

[10] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE symposium on security and privacy (SP)*, pp. 691–706, IEEE, 2019.

[11] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[12] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

[13] H. Ren, J. Deng, and X. Xie, "Grnn: Generative regression neural network—a data leakage attack for federated learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, may 2022.

[14] R. Xu, N. Baracaldo, Y. Zhou, A. Anwar, and H. Ludwig, "Hybridalpha: An efficient approach for privacy-preserving federated learning," in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, CCS '19, ACM, Nov. 2019.

[15] W. Jin, Y. Yao, S. Han, C. Joe-Wong, S. Ravi, S. Avestimehr, and C. He, "Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system," 2023.

[16] F. Yu, H. Lin, X. Wang, S. Garg, G. Kaddoum, S. Singh, and M. M. Hassan, "Communication-efficient personalized federated meta-learning in edge networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1558–1571, 2023.

[17] L. Yin, J. Feng, H. Xun, Z. Sun, and X. Cheng, "A privacy-preserving federated learning for multiparty data sharing in social iots," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2706–2718, 2021.

[18] R. Hu, Y. Gong, and Y. Guo, "Federated learning with sparsification-amplified privacy and adaptive optimization," *arXiv preprint arXiv:2008.01558*, 2020.

[19] A. Cheng, P. Wang, X. S. Zhang, and J. Cheng, "Differentially private federated learning with local regularization and sparsification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10122–10131, 2022.

[20] Y. Shi, Y. Liu, K. Wei, L. Shen, X. Wang, and D. Tao, "Make landscape flatter in differentially private federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24552–24562, 2023.

[21] X. Yuan, W. Ni, M. Ding, K. Wei, J. Li, and H. V. Poor, "Amplitude-varying perturbation for balancing privacy and utility in federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1884–1897, 2023.

[22] J. Zhao, M. Yang, R. Zhang, W. Song, J. Zheng, J. Feng, and S. Matwin, "Privacy-enhanced federated learning: a restrictively self-sampled and data-perturbed local differential privacy method," *Electronics*, vol. 11, no. 23, p. 4007, 2022.

[23] M. Wu, D. Ye, J. Ding, Y. Guo, R. Yu, and M. Pan, "Incentivizing differentially private federated learning: A multidimensional contract approach," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10639–10651, 2021.

[24] Q. Yang, X. Du, A. Liu, N. Wang, W. Wang, and X. Wu, "Adastopk: Adaptive federated shuffle model based on differential privacy," *Information Sciences*, vol. 642, p. 119186, 2023.

[25] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pp. 1–11, 2019.

[26] X. Liu, H. Li, G. Xu, R. Lu, and M. He, "Adaptive privacy-preserving federated learning," *Peer-to-peer networking and applications*, vol. 13, pp. 2356–2366, 2020.

[27] L. Sun, J. Qian, and X. Chen, "Ldp-fl: Practical private aggregation in federated learning with local differential privacy," *arXiv preprint arXiv:2007.15789*, 2020.

[28] D. Yu, H. Zhang, W. Chen, and T.-Y. Liu, "Do not let privacy overbill utility: Gradient embedding perturbation for private learning," *arXiv preprint arXiv:2102.12677*, 2021.

[29] S. Wu, M. Yu, M. A. M. Ahmed, Y. Qian, and Y. Tao, "Fl-mac-rdp: Federated learning over multiple access channels with renyi differential privacy," *International Journal of Theoretical Physics*, vol. 60, pp. 2668–2682, 2021.

[30] T. Zhang, A. Song, X. Dong, Y. Shen, and J. Ma, "Privacy-preserving asynchronous grouped federated learning for iot," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5511–5523, 2021.

[31] N. Ström, "Scalable distributed dnn training using commodity gpu cloud computing," 2015.

[32] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[33] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[34] S. Lu, R. Li, W. Liu, C. Guan, and X. Yang, "Top-k sparsification with secure aggregation for privacy-preserving federated learning," *Computers & Security*, vol. 124, p. 102993, 2023.

[35] X. Shen, Y. Liu, and Z. Zhang, "Performance-enhanced federated learning with differential privacy for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 24079–24094, 2022.

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

[37] Y. Lu and L. Fan, "An efficient and robust aggregation algorithm for learning federated cnn," in *Proceedings of the 2020 3rd International Conference on Signal Processing and Machine Learning*, pp. 1–7, 2020.

[38] Z. Wu, Q. Ling, T. Chen, and G. B. Giannakis, "Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4583–4596, 2020.

[39] H. Yang, X. Zhang, M. Fang, and J. Liu, "Byzantine-resilient stochastic gradient descent for distributed learning: A lipschitz-inspired coordinate-wise median approach," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5832–5837, IEEE, 2019.

[40] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

[41] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[42] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pp. 486–503, Springer, 2006.

[43] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[44] I. Mironov, "Rényi differential privacy," in *2017 IEEE 30th computer security foundations symposium (CSF)*, pp. 263–275, IEEE, 2017.

[45] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.