Future Generation Computer Systems 127 (2022) 362–372

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

An adaptive federated learning scheme with differential privacy preserving

Xiang Wu [a,*], Yongting Zhang [a], Minyu Shi [a], Pei Li [b], Ruirui Li [c], Neal N. Xiong [d]

[a] School of Medical Information & Engineering, Xuzhou Medical University, Xuzhou 221000, China
[b] Affiliated Hospital of Xuzhou Medical University Information Department, Xuzhou 221002, China
[c] Xuzhou Hengjia Electronic Technology Co. LTD, Xuzhou 221100, China
[d] Department of Mathematics and Computer Science, Northeastern State University, OK 74464, USA

ARTICLE INFO

Article history:
Received 18 November 2020
Received in revised form 24 August 2021
Accepted 9 September 2021
Available online 15 September 2021

Keywords:
Federated learning
Differential Privacy
Adaptive gradient descent
Privacy preserving

ABSTRACT

Driven by the upcoming development of the sixth-generation communication system (6G), the distributed machine learning schemes represented by federated learning has shown advantages in data utilization and multi-party cooperative model training. The total communication costs of federated learning is related to the number of communication rounds, the communication consumption of each participants, the setting of reasonable learning rate and the guarantee of calculation fairness. In addition, the isolating data strategy in the federated learning framework cannot completely guarantee the privacy security of users. Motivated by the above problems, this paper proposes a federated learning scheme combined with the adaptive gradient descent strategy and differential privacy mechanism, which is suitable for multi-party collaborative modeling scenarios. To ensure that federated learning scheme can train efficiently with limited communications costs, the adaptive learning rate algorithm is innovatively used to adjust the gradient descent process and avoid the model overfitting and fluctuation phenomena, so as to improve the modeling efficiency and performance in multi-party calculation scenarios. Furthermore, in order to adapt to the ultra-large-scale distributed secure computing scenario, this research introduces differential privacy mechanism to resist various background knowledge attacks. Experimental results demonstrate that the proposed adaptive federated learning model performs better than the traditional models under fixed communication costs. This novel modeling scheme also has strong robustness to different super-parameter settings and provides stronger quantifiable privacy preserving for federated learning process.

1. Introduction

Mobile applications and technologies such as the Internet of Things (IoT) and social networks have led to staggering growth in data traffic and mobile devices, which brings challenges for data communication and secure computing to 6G technology [1–3].

Traditional privacy preserving methods are mainly based on data distortion and data encryption, but these methods are employed difficulty due to poor protection effects and complex implementation [4]. To improve the privacy protection, Dwork proposed the concept of differential privacy (DP) in 2006 [5]. Privacy can resist various new attacks under the assumption of maximum background knowledge [6]. In 2016, Google proposed federated learning (FL) as a secure machine learning framework to solve the data privacy problem in large-scale mobile device computing [7–10]. FL is one of the most popular Secure Multi-party Computing (SMC) frameworks in machine learning [11–16], which enables multiple clients to collaboratively train models under a central server, and has been applied to 5G communication scenarios [17–21]. This also provides a secure and efficient calculation method for 6G communication.

Among the existing research methods, many FL researches mainly focus on reducing communication costs, improving model performance and generalization capabilities, and further discussing the privacy security of federated learning. Many studies have focused on improving the performance of federated learning models [22], such as how to achieve better convergence within limited communication costs. Some scholars are committed to studying the universality of the federated learning framework in different scenarios [23]. There are other studies focusing on the cross-device deployment and setup of the federated learning model, such as how to balance the accuracy distribution of the model among multiple devices [24]. In the study of strengthening

* Corresponding author.
E-mail addresses: wuxiang@xzhmu.edu.cn (X. Wu), 301910911596@stu.xzhmu.edu.cn (Y. Zhang), 301810911545@stu.xzhmu.edu.cn (M. Shi), lipei@xzhmu.edu.cn (P. Li), geiwre@hotmail.com (R. Li), xiongnaixue@gmail.com (N.N. Xiong).

https://doi.org/10.1016/j.future.2021.09.015
0167-739X/

the privacy security of federated learning, some researches [25,26] have also proposed solutions to solve potential security problems in federated learning, using encryption, noise, and other methods to protect the privacy information. For example, Robin C. Geyer and others combined DP with FL [27]. In order to optimize the machine learning process in FL, many optimization methods are used [28–31]. However, the fixed learning rate of the existing FL model will cause the learning process to be greatly affected by the small batch datasets, resulting in local optimization during the convergence process, in addition, the impact of noise on the model performance in privacy-related research has not been resolved. Therefore, the self-adjusting ability of the FL model and the balance between privacy and usability are still the bottleneck in this field.

The traditional FL framework protects data privacy through parameter exchange under the encryption mechanism, while DP disturbs some sensitive characteristics of users through the noise mechanism, so that individual behaviors cannot be identified. From a theoretical perspective, DP obviously has better privacy protection capabilities and protects users' privacy more deeply [32].

In this context, we proposed an optimized federated model with privacy preserving and learning rate automatic adjustment function. Specifically, the DP mechanisms and scalability strategies are used to protect the privacy of each computing participant. The adaptive gradient descent algorithm is used to overcome the problems of local convergence and poor stability. The main contributions of this research are as follows.

(1) Proposed a federated learning optimization scheme with adaptive gradient descent function. This scheme can reduce overfitting and performance fluctuation in federated learning by adaptive algorithm and improve the generalization ability of the model.
(2) Improved the privacy performance of local training process by the differential privacy and the scaling of update volume. A reasonable mechanism is designed to strengthen the privacy security of each computing node in federated learning process.
(3) Proposed a cost-control scheme combining efficient optimization algorithm and distributed learning model. The autonomic adjustment ability of optimization algorithm and the distributed updating mechanism can reduce the communication cost effectively.

This research paper is organized as follows. Section 2 briefly introduced the background knowledge. Section 3 clarified the key steps to achieve user-level privacy security in the federated learning process, the limitations of the original SGD algorithm, the advantages of the adaptive gradient descent algorithm and the specific noise addition process. Section 4 compared the performance of the adaptive algorithm and the SGD algorithm, verified the sensitivity of different algorithms to changes in the initial learning rate, and analyzed model performance under different privacy budgets. Finally, we conclude this paper with some future work in Section 5.

2. Preliminaries

In this section, we give some basic notions and definitions of federated learning, then introduce the gradient descent algorithm in the machine learning optimization process. In addition, this section also explains the differential privacy method as the privacy security enhancement method in the process of federated learning.

2.1. Federated learning

Due to the data privacy, it is difficult to share the useful data between different enterprises or even different departments of one enterprise. Therefore, the secure computing framework represented by federated learning is proposed, which enables multiple organizations to conduct effectively joint federated modeling based on users, data and third parties.

According to the features of different datasets, FL [33,34] can be divided into three categories, including horizontal federated learning, vertical federated learning and federated transfer learning. If the data features $(f_{d1}, f_{d2}, \ldots)$ between the two datasets overlap more and the user characteristics $(f_{u1}, f_{u2}, \ldots)$ overlap less, it is called horizontal federated learning. If the user characteristics $(f_{u1}, f_{u2}, \ldots)$ between the two datasets overlap more and the data features $(f_{d1}, f_{d2}, \ldots)$ overlap less, it is called vertical federated learning. Otherwise, it is called federated transfer learning [35–37].

Decentralization is the key feature of FL. Shannon proposed the concept of information entropy [38,39], the uncertainty of information is described using a probability distribution function. Assume that the probability distribution of events $a, b, c, \ldots$ is $p = (p_a, p_b, \ldots, p_n)$, which meet $\sum_{i=a}^{n} p_i = 1$, $0 \leq p_i \leq 1$, the function of information entropy is defined as,

$$H(X) = H(p_1, p_2, \ldots, p_n) = -k \sum_{i=1}^{n} p_i log p_i, \tag{1}$$

Information entropy is a measure of information uncertainty, if all information at the i node event is obtained, the information entropy is 1. In the above formula, X can be understood as the total information entropy combined with events $a, b, c, \ldots$, and the information entropy corresponding to p_i can be understood as the micro information entropy of X at the node.

Classical FL [40] obtains the global statistical model from the data of multiple different devices, this model optimization process can be expressed with an objective function.

$$min\ F(X),\ \ where\ F(X) \coloneqq \sum_{k=1}^{n} p_k F_k(n), \tag{2}$$

where n represents the total number of devices involved in the calculation, F_k represents the local objective function of device k, p_k represents the upper limit of contribution for this device, $p_k \geq 0$ and $\sum_{k=1}^{n} p = 1$.

The basic process of multi-party collaborative modeling in federated learning process is as shown below:

Step 1: Clients selection. The central server M selects K node clients that meet the requirements from a group of clients. In this process, the central server selects sample users according to the client running status and network status.

Step 2: Broadcasting. The server distributes the central model M to the sample clients.

Step 3: Client-side calculation. Each client k updates the local computing model with the local data $m_k \leftarrow M_i + ClientData$. For example, run gradient descent algorithm to optimize the machine learning model, encrypt and transmit the local update $U_k = M_i - m_k$ to the center. In the process of sample alignment and model training, each client's data is stored locally.

Step 4: Aggregation. The server collects all the updating information $\{U_1, U_2, \ldots, U_K\}$. In order to improve efficiency, once enough devices report the results, the lagging ones will be discarded. This step includes multiple processes, including security aggregation, lossy compression, noise addition and update clipping.

Step 5: Model updating. The cumulative update U_i of all clients participating in the current round i is aggregated, and the server

shares the model based on this update center and releases the new model $M_{i+1} \leftarrow M_i + U_i$ to the selected clients in the next round.

Repeat the above steps until the loss function converges. In the federated learning process, multiple parties can collaboratively train the model, and the method reduced the risk of privacy disclosure, but more and more studies have shown that model parameters may still leak the privacy of related users [41,42].

2.2. Gradient descent and model update

Model optimization based on gradient descent is a key process of FL. As a machine learning framework for secure multi-party computing, the optimization process of FL is also closely related to the optimization process of each computing node. In the gradient descent model optimization algorithm, the gradient of the objective function $f(\theta)$ with respect to the parameters θ will be the fastest direction of the objective function rising. For the minimization optimization problem, it is only necessary to advance the parameters by one step in the opposite direction of the gradient to achieve the reduction of the objective function. This stepsize is also called the learning rate η, the choice of different learning rates may have different influence. After the model parameters and the objective function optimized are given, the algorithm is minimized by updating in the opposite direction of the gradient. The learning rate determines the update step size at each moment. The parameter update formula is as follows:

$$\theta \leftarrow \theta - \eta \bullet \Delta_\theta f(\theta). \tag{3}$$

If the learning parameter η is too large, it may cause the algorithm to hover around the optimal value and fail to converge. If η is too small, both parameter updating and model convergence are slow.

Stochastic Gradient Descent (SGD) [43–45] is a gradient descent algorithm commonly used in existing machine learning models, naive SGD is the simplest and has no concept of momentum,

$$v_t^1 = \eta g_t, \tag{4}$$

$$v_t^2 = I^2(\in = 0), \tag{5}$$

The updating step is

$$\theta_{t+1} = \theta_t - \eta g_t. \tag{6}$$

In other words, SGD updates the objective function at the same learning rate. A large number of parameters are often involved in deep learning models, and the updating frequency of different parameters is often different. The parameters that are updated infrequently should be set with a larger stepsize, while the parameters that are updated frequently should be set with a smaller stepsize, so as to make the parameters more stable and control the impact of the single sample on the model.

It can be seen that the fixed learning rate in SGD cannot meet the learning needs of each sub-model, and the unreasonable learning rate will cause the SGD algorithm to have the risks of slow convergence speed and oscillation of the optimization process near the saddle point. In order to better adapt to the optimization process of the objective function of different dimensions, the second order momentum recording mechanism is introduced, and several optimized versions of gradient descent algorithms are explored and analyzed.

Adam algorithm [46–48] has the advantages of both AdaGrad [49] and RMSProp [50]. Adam not only calculates the adaptive parameter learning rate based on the first-order moment mean like the RMSProp algorithm, but also makes full use of the gradient's second-order moment mean (uncentered variance).

Specifically, the Adam algorithm calculates the exponential moving average of the gradient, the hyperparameters b_1 and b_2, and controls the decay rate of moving averages. In addition to storing the average value of the exponential decay of the square v_t^2 of the past gradient like Adadelta and RMSprop, Adam also maintains the average value of the exponential decay of the past gradient v_t^1 like momentum. If v_t^1 and v_t^2 are initialized to 0 vectors, they will 0 offset, so the deviation correction is done, these deviations are offset by calculating the deviation corrected v_t^1 and v_t^2,

$$\hat{v}_t^1 = \frac{v_t^1}{1 - b_1^t}, \tag{7}$$

$$\hat{v}_t^2 = \frac{v_t^2}{1 - b_2^t}, \tag{8}$$

The update process of the algorithm is

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t^2} + \in} \hat{v}_t^1. \tag{9}$$

The Adabound [51] algorithm combines the advantages of Adam and SGD, which indicates the algorithm has a fast convergence speed in the early training stage, and a slow convergence speed in the later stage and a better final performance. In other words, Adabound looks like Adam early in training and more like SGD at the end of training. The control of the learning rate of the Adabound algorithm is similar to gradient cutting. To solve the gradient explosion problem, the algorithm cuts the learning that is greater than a certain threshold. The cutting process is expressed by the following formula,

$$Clip(\alpha/\sqrt{v_t}, \eta_l, \eta_u). \tag{10}$$

η_l, η_u are the upper and lower limits of the learning rate. In order to achieve a smooth transition of the learning rate, as the learning time changes, the lower limit of the learning rate η_l gradually increases from 0 to α and the upper limit η_u decreases from ∞ to α. In common situations, using its default parameters can achieve relatively excellent and stable final results.

2.3. Differential privacy

If there are two neighboring datasets D and D', that is $\left|D \Delta D'\right| = 1$, M is a randomization algorithm, $range(M)$ represents the set of all possible outputs of algorithm M, and S is any subset of $range(M)$, if M satisfies:

$$\frac{P\left[M(D) \in S\right]}{P\left[M(D') \in S\right]} \leq e^{\varepsilon}. \tag{11}$$

Differential privacy method is usually realized by adding fuzziness noise to the query results, and its essence is to protect the sensitive information by adding quantifiable randomness to data. The commonly used noise adding mechanisms are the Gaussian mechanism and the Laplace mechanism. In the Gaussian mechanism, there are

$$M(d) \triangleq f(d) + N(0, S_f^2 \cdot \sigma). \tag{12}$$

$M(d)$ represents the query result after noise addition, $N(0, S_f^2 \bullet \sigma^2)$ represents the noise generated by the Gaussian mechanism, its mean is 0, the standard deviation is $S_f \bullet \sigma$, and S_f is the sensitivity adjustment operator.

In the relevant definition of differential privacy, a smaller privacy budget means that data is more secure but less available. When the budget is used up, the data cannot be accessed anymore.

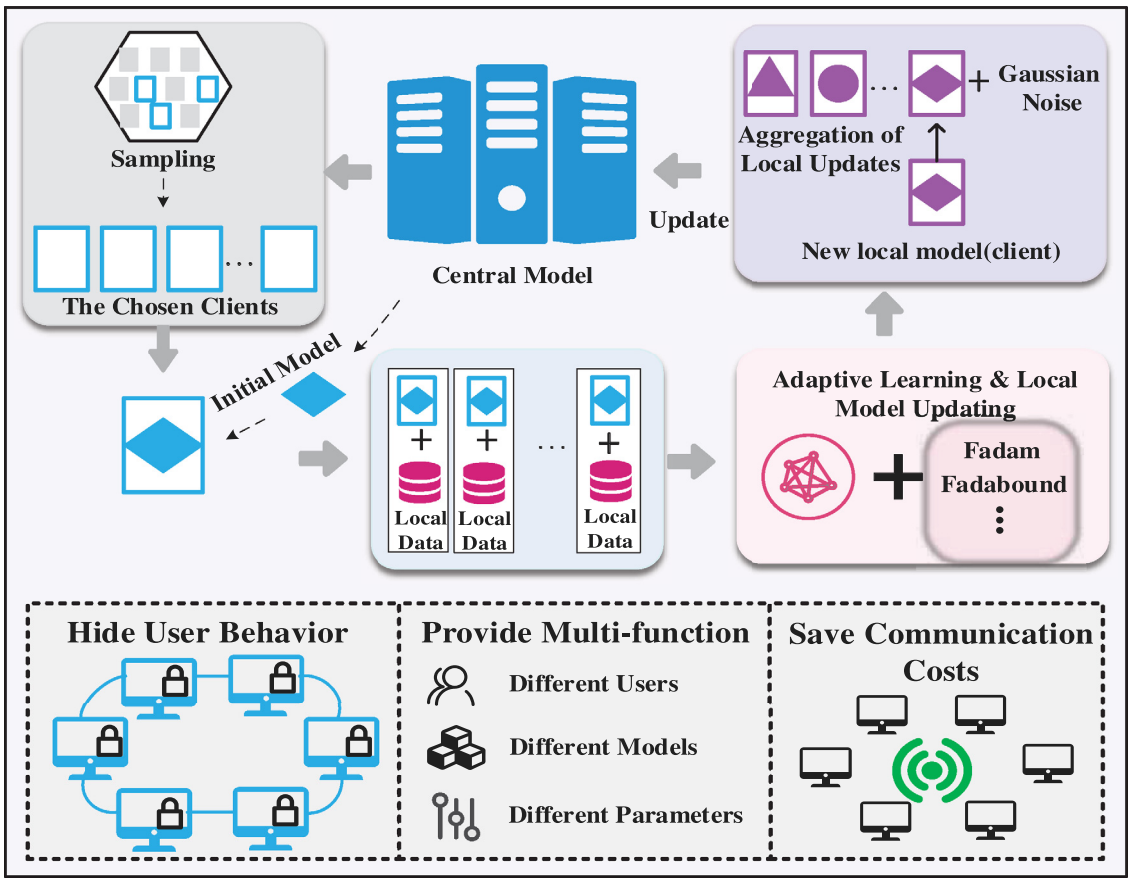


Fig. 1. The overall framework of federated adaptive learning scheme.

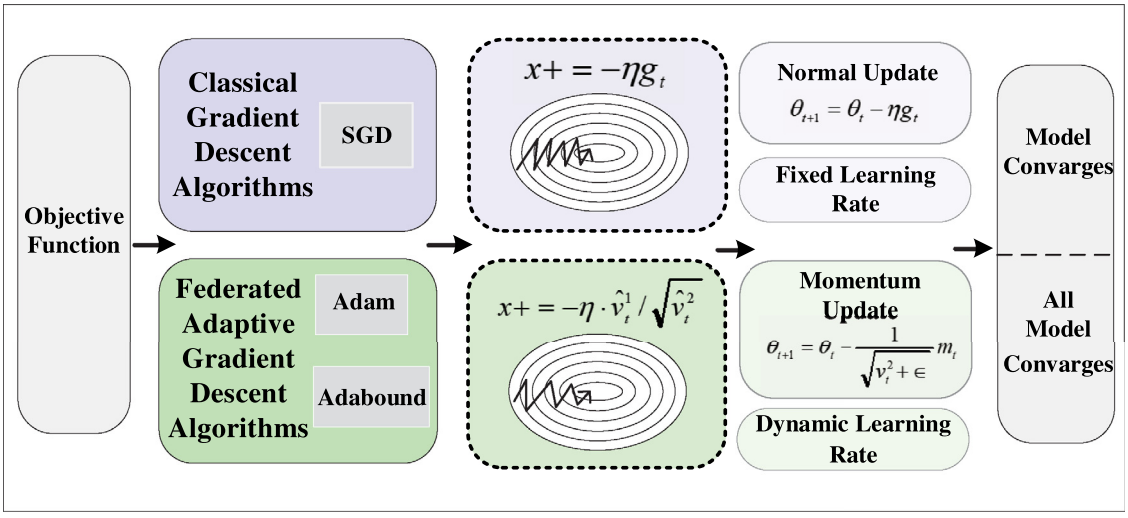


Fig. 2. Comparison of federated adaptive gradient descent algorithms and SGD algorithm.

The amount of added noise is related to the sensitivity of the algorithm to the data and the degree of privacy security. The global sensitivity $S_G(f)$ is defined as follows:

$$S_G(f) = \max_{d,d'} \|f(d) - f(d')\|_k. \tag{13}$$

In the specific application, it is necessary to select the noise disturbing mechanism according to the data features, and add noise reasonably according to the privacy security requirements and the global sensitivity.

3. Proposed method

This study apply the differential privacy mechanism to the updating process of the federated learning center model, and define the contribution of local updates with reference to the customer-level privacy. In addition, we use optimization algorithms that can adaptively adjust the learning rate to replace the original gradient descent algorithm, which can ensure that our model achieves global optimality and strong robustness on non-convex functions. The overall optimization framework is shown in Fig. 1.

3.1. Adaptive adjustment in federated learning

The learning rate and the number of iterations are the hyperparameters that need to be set before the machine learning process. The traditional gradient descent algorithm (such as SGD) [52] will often slow down the convergence rate and produce a local optimization solution due to the fixed learning rate. In view of the limitations of SGD, we propose the method that combines the adaptive gradient descent strategy with FL called Fadam. Fadam calculates the gradient of the objective function with the parameters, uses the first and second order momentum based on the historical gradient $v_t^1 = \phi(g_1, g_2, \ldots, g_t)$, $v_t^2 = \psi(g_1, g_2, \ldots, g_t)$ to update the local model parameters in each iteration:

$$\theta_{t+1} = \theta_t - \frac{1}{\sqrt{v_t^2 + \in}} v_t^1, \tag{14}$$

$$\theta_{t+1} = \theta_t - v_t^1. \tag{15}$$

The learning rate is reflected in the gradient descent of the model convergence. Usually, each local model trained at a certain learning rate contributes parameters to the central server and reduces the loss function value, achieving gradient descent. The central server adjusts to make all local models have higher accuracy and generalization ability. In order to avoid the model overfitting by adjusting the learning rate independently, we introduced the adaptive gradient descent algorithms.

Although SGD can reach a minimum value, it takes longer than other algorithms and may get stuck in saddle points in non-convex functions. We use an optimization method that is different from the traditional gradient descent algorithm, which calculates the first-order momentum estimation and second-order momentum estimation of the gradient to design independent adaptive learning rates for different parameters. Our method retains the learning performance advantage of the adaptive gradient algorithm (AdaGrad) on gradient sparse datasets, and the performance advantage of root mean square propagation algorithm(RMSProp) in dealing with non-steady state problems. The optimization process is shown in Algorithm 1.

In addition, since the second-order momentum in Fadam is recorded through a fixed time window, the data used for training may lose information because of the change of the time window, so it is difficult to obtain the optimal solution in the modeling training process. To obtain faster learning speed in the early stage and better generalization ability in the later stage, we applies the Adabound algorithm to the FL process, and proposes an algorithm, termed as Fadabound. Fig. 2 briefly introduces and compares the basic process of the classical stochastic gradient descent algorithm and the adaptive algorithm. The Fadabound process is shown in Algorithm 2.

3.2. Differential privacy implementation in federated learning

To further address the privacy security issues of each computing participant in the FL, we use the differential privacy method when the central server aggregates the parameter update information provided by the local models to avoid the privacy disclosed. Compared with the traditional privacy preserving mechanism, this method has the characteristics of quantifiable privacy budget and user-level sensitive behavior hiding. When the local model parameter updating information in each round is collected, we add noise to these updates based on data characteristics and the global sensitivity S_g, such as the number of clients participating in a training round is k, and the average noise obtained by each user is $\frac{1}{k}(0, \sigma^2, S^2)$. In this process, the updates provided by each user are normalized and scaled, and the privacy budget value is allocated based on the number of users.

Assume that the total number of users is n, and we select K users as the computing participants in the i round of communication. In the FL process, the local model updated by each user is m_k, and the existing central model is M_i. The partial update amount is

$$\Delta m_k = M_i - m_k, \tag{16}$$

In this round, the theoretically updated central model is

$$m_{i+1} = M_i + \sum_{k=0}^{k=K} \Delta m_k. \tag{17}$$

Due to traditional data collection and user contribution agreements may disclose specific user characteristics during model optimization, thus we introduce the Gaussian mechanism to the update amount to protect objective function.

In the Gaussian mechanism, the $l_2 - Sensitivity$ to the query function f also represents the maximum l_2 distance between the output of two adjacent datasets, which is expressed by the formula as:

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\|_2, \tag{18}$$

For $\forall \delta \in (0, 1)$, $\sigma > \frac{\sqrt{2ln\frac{1.25}{\delta}}\Delta f}{\in}$, there is the noise $y \sim N(0, \sigma^2)$ meets the principle of $(\in, \delta)-$ *differential privacy*, the query algorithm for two neighboring datasets D and D' is M. In formula (19), S represents the range of algorithm M, δ represents tolerable relaxation value for differential privacy mechanism:

$$P[M(D) \in S] \leq P[M(D') \in S] + \delta. \tag{19}$$

We add noise to the update results of each iteration in federated learning process and distort the update of the local model. After scaling update, each local model's contribution to the central model is

$$\Delta \hat{m}_k = \Delta m_k / \max(1, \frac{\|\Delta m_k\|_2}{S}), \tag{20}$$

After the iteration of round i, the central model is updated to

$$M_{i+1} = M_i + \frac{1}{K}(\sum_{k=0}^{k=K} \Delta \hat{m}_k + N_{GS}(\sigma^2 S^2)). \tag{21}$$

3.3. Iterative optimization of central model

In order to protect the participation behavior of specific users, we scale local updates to replace the original update amount $\Delta w^k = w^k - w_t$ with $\Delta \overline{w}^k = \Delta w^k / \max(1, \frac{\|\Delta w^k\|_2}{S})$, and return local zoom update amount to the center in each iteration. The central server does not collect data of each client, but updates the central model by collecting gradient updates, this training process can prevent attackers from backstepping the model data to a certain extent.

For each client involved in the calculation, the latest model parameters are obtained from the center as the initial model parameters before each round, and the local model is used to calculate gradient updates and sent to the central server.

The update process of the central model mainly includes several key steps,

(1) The central server selects user equipment k suitable for training and sends the original parameter model M_i or opens download permission.

Algorithm 1. Feadam

Input: Datasets, privacy budget ϵ, learning rate α, the number of iterations t.
Output: learning rate α, the parameter θ of the objective function.
1: Initialize the model parameter.
2: Begin $b_1 = 0, b_2 = 0, t = 0$:
3: **Do** $t \leftarrow t + 1$
4: The gradient update amount is $g_t \leftarrow \Delta_\theta f_t(\theta_{t-1})$
5: Calculate the first-order momentum $v_t^1 \leftarrow b_1 \cdot v_{t-1}^1 + (1 - b_1) \cdot g_t$
6: Calculate the second-order momentum $v_t^2 \leftarrow b_2 \cdot v_{t-1}^2 + (1 - b_2) \cdot g_t^2$
7: Update the first-order momentum estimation $\hat{v}_t^1 \leftarrow v_t^1 / (1 - b_1^t)$
8: Update the second-order momentum estimation $\hat{v}_t^2 \leftarrow v_t^2 / (1 - b_2^t)$
9: The updated model is $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{v}_t^1 / (\sqrt{\hat{v}_t^2} + \epsilon)$
10: Until the model converges on the local dataset
11: **Return** θ_t
12: Send the model parameters θ_t of this iteration to the central server
13: The central server calculates the contribution $\theta^i - \theta^{i-1}$ of the current iteration and delivers it
14: Iterate until all clients converge
15: **End.**

Algorithm 2. Fadabound

Input: Datasets, privacy budget ϵ, learning rate α, the number of iterations t.
Output: θ.
1: Begin set $v_0^1 = 0, v_0^2 = 0$.
2: **For** each round of iterations **do**
3: Gradient descent at time step t: $g_t \leftarrow \Delta_\theta f_t(\theta_{t-1})$
4: Calculate the first-order momentum $v_t^1 \leftarrow b_1^t v_{t-1}^1 + (1 - b_1^t) g_t$
5: Calculate the second-order momentum $v_t^2 \leftarrow b_2 v_{t-1}^2 + (1 - b_2) g_t^2$
6: Clip learning rates by $Clip(\alpha / \sqrt{v_t^2}, \eta_l, \eta_u)$
7: Update the first-order momentum estimation $\hat{v}_t^1 = \frac{v_t^1}{1 - b_1^t}$
8: Update the second-order momentum estimation $\hat{v}_t^2 = \frac{v_t^2}{1 - b_2^t}$
9: Until the model converges for the local dataset
10: Send the model parameters θ_t of this iteration to the central server
11: Central server calculates the contribution $\theta^i - \theta^{i-1}$ of the current iteration and delivers it
12: **End.**
13: **Return** resulting parameters $\theta_1^i, \theta_2^i, \ldots, \theta_k^i$ from clients to central server.

(2) The sample clients update the parameters with local data and calculate the updated gradient. In this process, the adaptive gradient descent algorithm is used to ensure good convergence.
(3) Scaling the gradient updated by a specific user to avoid excessive contribution from a single computing node.
(4) Add differential privacy noise to the overall aggregation update of a single communication process.
(5) The gradient update results are sent to the central server to update the central model and broadcast to participating clients.

4. Experiments

In this section, we tested the adaptive federated learning model with three experiments. Experiments were completed under Intel Core 9750, 16G memory and Windows 10 operating system, and model training was implemented in the TensorFlow framework. First, we analyzed the robustness of the model under different initial learning rates in multi-party computing scenarios. Second, compared the model convergence performance between different algorithms. Finally, we analyzed the privacy preserving capabilities of the model.

4.1. The Robustness comparison between different algorithms in the same initial learning rate

The sensitivity of different gradient descent algorithms to learning rate change is an important index for performance evaluation. The value of the learning rate parameter in the gradient descent process of the model is usually set to 0.01. Considering the important influence of the hyper-parameter of learning rate on the research algorithm, the initial learning rate was adjusted during the experiment, such as 0.1, 0.01, 0.001 and 0.0001. To study the sensitivity of the model to changes in the initial learning rate, we observed the performance of the three algorithms with the same initial learning rates under 1000 and 10,000 users, respectively.

In the training process with 1000 clients, there are about 24 rounds of communication iterations. As can be seen from Fig. 3, the overall change in accuracy and loss of SGD is the largest due to the influence of the initial learning rate.

The lower the initial learning rate, the worse the overall performance. For the adaptive algorithm, the accuracy and loss value of the Fadam algorithm change greatly when the learning rate changes from 0.1 to 0.01, and in other cases, the model performance is not much different, and the overall curve of the Fadabound algorithm under different learning rates has always been relatively good fit.

In the large-scale training process of 10,000 (more than 300 rounds of communication), SGD still fluctuates greatly under

Algorithm 3. Optimization process of adaptive federated learning

Input: the parameter M_j of center servers, clients number K, privacy budget ϵ, the learning rate α.
Output: α and other model hyper-parameter.
1: Initialize model parameters.
2: **For** each round of iterations **do**
3: Sample K clients from the total clients as the current round of samples
4: Distribute the existing central model parameters m_j to each sample client as the initial model parameters
5: Each client k use federated adaptive algorithm for local training (with local data)
6: Get local model parameters m_k
7: Calculate the difference Δm_k between m_k and the central model parameters, norm value $\|m_k\|_2$
8: Calculate the sensitivity of this round $S = median\|m_k\|_2$
9: Process the total amount of local updates $\Delta\hat{m}_k = \frac{\Delta m_k}{\max(1, \frac{\|\Delta m_k\|_2}{S})}$
10: Add noise to the total update, then the center model parameters becomes:

$$M_{t+1} \leftarrow M_t + \frac{1}{k}\left(\sum_{k=0}^{K} \Delta m_k / \max\left(1, \frac{\|\Delta m_k\|_2}{S}\right) + N(0, \sigma^2 S^2)\right).$$

11: **End.**

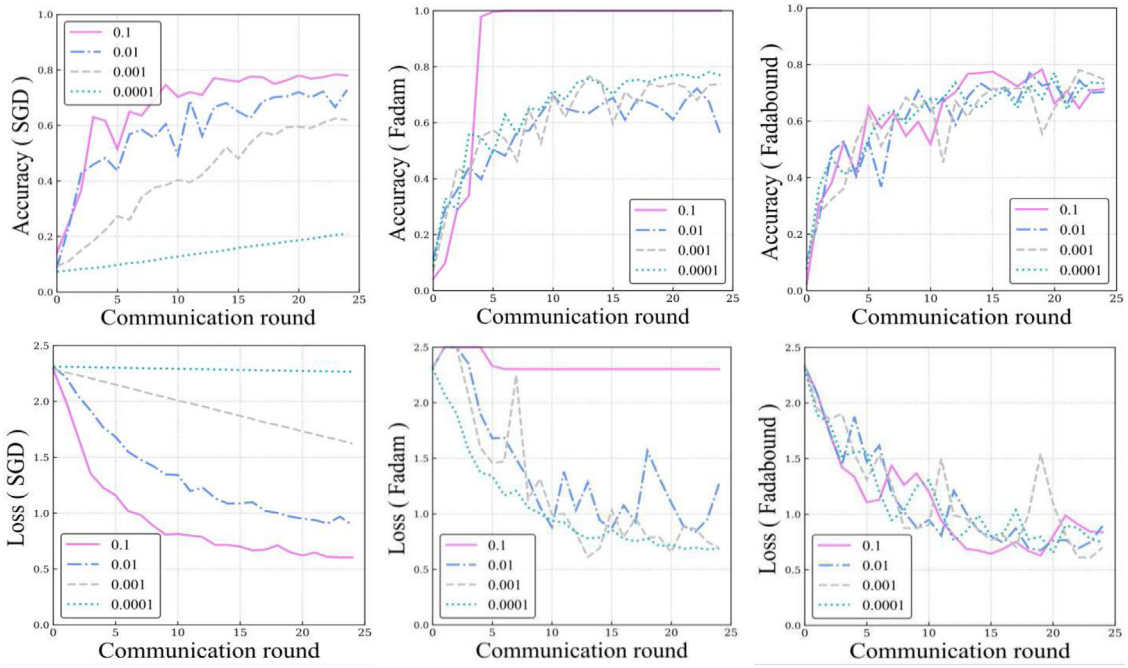


Fig. 3. The performance of gradient descent algorithms at different initial learning rates (1000 clients).

different learning rates, and the federated adaptive optimization algorithm has the better curve fitting under different initial rates. In general, the Fadam and Fadabound algorithms have smoother accuracy curves, which means that the optimization algorithm is more stable and the corresponding training process is more efficient. Although the Fadam and Fadabound models perform poorly at the learning rate of 0.1, they perform better than the SGD model at the smaller learning rates and have a high curve fit for accuracy, which indicates more consistent performance.

It can be concluded from the Fig. 4 that the optimization algorithm represented by the Fadabound algorithm has better adaptability to the change of the initial learning rate. Although the SGD algorithm performs better at a higher initial learning rate, it is more sensitive to changes in the learning rate. In addition, the performance of SGD is more stable at the same learning rate with fixed learning rate, which leads to a single gradient descent direction. And the momentum set by the adaptive optimization algorithm to avoid local optimization may be the reason for the large performance fluctuation. Overall, although Fadam produces larger fluctuations at a higher learning rate, it still performs better than SGD in other situations.

4.2. Performance comparison between different algorithms in terms of model convergence

To measure the impact of learning rate on the training process and the performance of the overall model, we take accuracy rate and loss rate as evaluation indexes. In order to test the actual performance of several federated optimization algorithms, we compared the performance of different algorithms and communication times. In general, the number of communications increases as the number of users increases. The lower the loss value, the higher the accuracy and the better the performance of the model. Figs. 5 and 6 show the accuracy and loss performance of the model under 1000 users and 10 000 users, respectively.

As can be seen from Fig. 5, the federated optimization algorithm drops faster than the SGD algorithm in the initial training period with 1000 clients. As the number of communication rounds increases, the convergence speed of the SGD algorithm

Table 1
Privacy security analysis the federated adaptive differential privacy model.

Measures	Upload model parameters during federated learning	Clipping local update contributions	Add DP noise to the aggregated update
Influences	Reduce the transmission of raw data	Hide specific user behavior	Randomized query
Respond to attack types	Password intrusion\Vulnerability intrusion\SQL injection attack\Untrusted third party	Black box attack\White box attack\Adversarial attack\Background knowledge attack	Link attack\Differential attack

slows down and the fluctuation increases. On the contrary, the accuracy of federated adaptive algorithms continues to improve and tends to stabilize, while the loss value continues to decline and stabilize. In Fig. 6, it can be seen that Fadabound's performance is better in most cases. Although the performance of SGD algorithm in early training as good as the federated optimization algorithms, the performance improvement of SGD model slows down or even stops at the later stage of the training process, and the performance is the worst in terms of accuracy and loss.

Experiments show that traditional gradient descent algorithm is available in some cases, but if there is no reference learning rate and the structure of the model is unknown, the traditional gradient descent algorithm has a poor late convergence. In contrast, the adaptive algorithms in this paper can optimize unknown models through autonomous adjustments (see Table 1).

4.3. Performance of the novel scheme in privacy preserving

4.3.1. The privacy security advantage of our model

Federated learning, as a multi-party modeling method, has the risk of exposing the user behavior of each client model during the training process. To achieve high performance of the federated model while protect the privacy of individual user behavior, in this model, a user-level privacy preserving method is introduced to ensure the privacy security in the processes of data transmission, model training and model used, improving the availability of the model maximally. This method avoids the risk of password intrusion, vulnerability intrusion and data disclosure caused by untrusted third parties in the central database.

The scheme proposed constrains and tailors the local update amount in the process of uploading gradients and weights from the computing node to the central server. Compared with adding noise for a single data, scaling and adding noise for a whole batch of data can ensure the availability of the model while blurring the specific user behavior. In addition to using the zoom update amount instead of the directly calculated update amount, a threshold is used to constrain the individual contribution of nodes to the central model, so as to protect the local user information and behavior. In this method, even if the attacker obtains the partial update amount, it is difficult to recover the real parameters, not to mention the behavior of specific users, and avoid the black box attacks, white box attacks and so on.

Add differential privacy noise to each round of global update during each iteration of the model. Differential privacy noise provides quantifiable privacy protection for the modeling process and enhances privacy security from the perspective of information theory. For datasets D_A and D_B, an appropriate amount of Gaussian noise can be added according to the principle of differential privacy to achieve quantifiable privacy protection at the entire datasets level, avoiding disclosure of multiple data tables and link attacks and differences caused by unlimited query attack.

4.3.2. Model performance under different privacy budgets

Our model further strengthens the privacy security of the model at the customer level. In order to test the anti-noise ability of the algorithm and the robustness under different noise scales, we compared the performance of the three algorithms in different privacy budget under four different scales. Generally, the larger the privacy budget, the larger the added noise scale, and the smaller privacy budget represents higher data availability.

It can be seen from Fig. 7, in the same privacy budget, the overall effect of our adaptive algorithm is significantly better than that of the SGD algorithm, which may be caused by the SGD algorithm falling into a local optimum or the learning rate being too slow. Specifically, in the early stage of the experiment, the accuracy of our model changes faster, while the accuracy of SGD improves slowly. In the later stage, the convergence speed and fluctuation of the federated adaptive algorithm tended to be stable, and the overall performance was better than the SGD algorithm.

The federated adaptive model is less sensitive to different privacy budgets. The experiment shows that the federated adaptive algorithm model has a faster convergence speed and better final results under the condition of fixed number of communications, which means that the federated adaptive algorithm may correspond to lower communication costs and better model performance, and the robustness of the scale noise makes the setting of the scale of noise more focus on the actual needs.

5. Conclusion and future work

The main contributions of this research are that the learning rate adaptive algorithm and differential privacy mechanism are combined to improve the availability and security of federated learning modeling process. In addition, this federated modeling scheme provides technical support for improving privacy security in the application process of the IoTs, and designs a method for adapting distributed computing process to the new communication standards.

Compared with previous studies, this scheme pay more attention to improve the model's insensitivity to hyperparameters and privacy security. The experiments proved that the federated adaptive learning rate gradient descent algorithms performs better in robustness and flexibility when the learning rate changes. At the same learning rate, especially at a lower learning rate, our model convergence effect is better than SGD, and at different learning rates, our model also shows strong robustness. In addition, we analyzed the privacy preserving advantages of the framework through theoretical cases, and experiments proved that our model performs less volatile and has better anti-noise ability under different privacy budgets. In a word, the introducing of adaptive algorithms and differential privacy mechanisms are not only increase the applicability and training accuracy of federated modeling, but also reduce the risk of privacy disclosure in the training process. Although the proposed scheme can improve the accuracy and stability of the model and reduce the influence of hyperparameter fluctuations in the case of limited training data, there are still limitations in the convergence efficiency of the model training and the availability for large-scale data scenarios.

In future research, we will optimize the model according to different data processing requirements, and the datasets of corresponding professional fields will be used for simulation experiments, so as to provide technical support for the intelligent electronic medical records, medical data segmentation, financial risk prediction and other related fields.

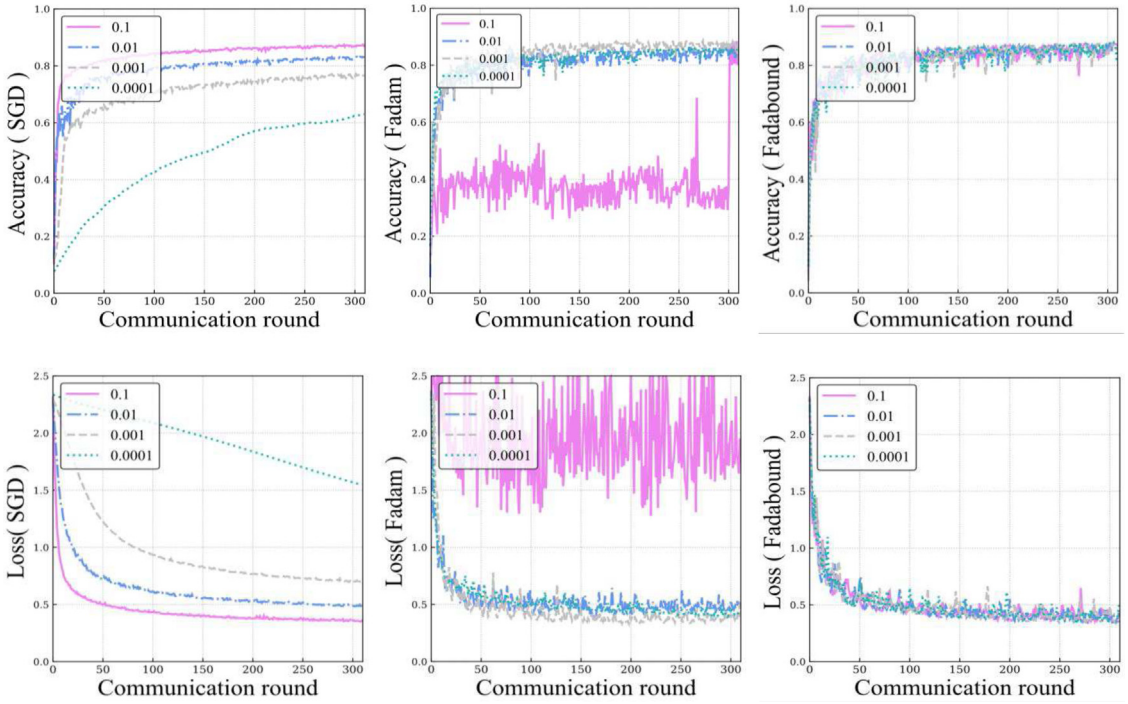


Fig. 4. The performance of gradient descent algorithms at different initial learning rates (10 000 clients).

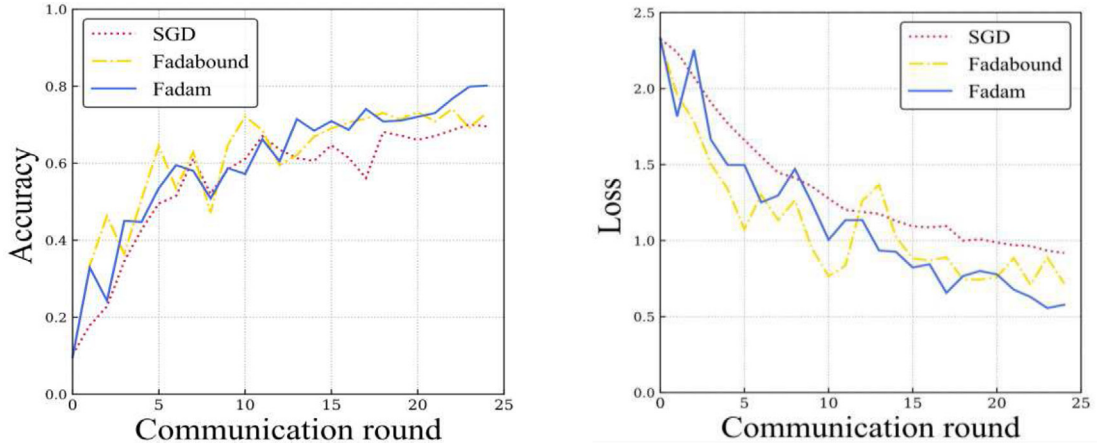


Fig. 5. The performance of different algorithms at fixed initial learning rate (1000 clients).

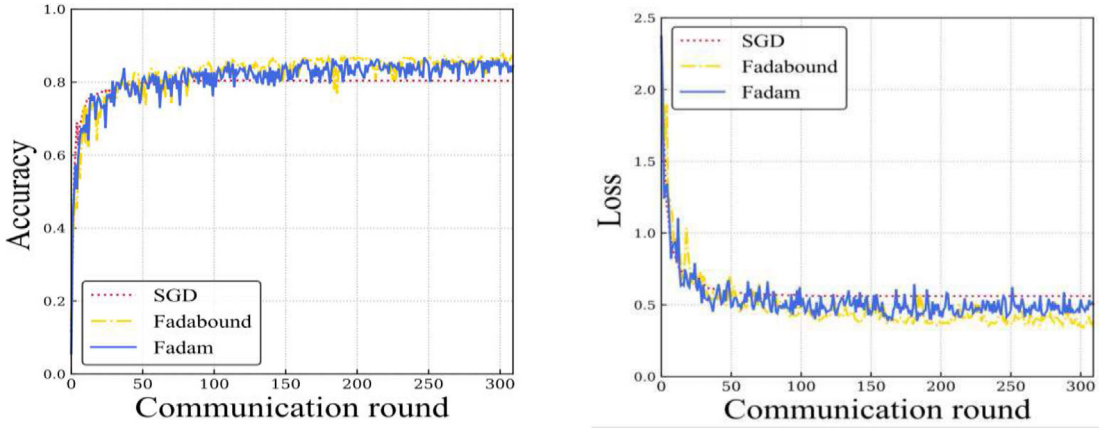


Fig. 6. The performance of different algorithms at fixed initial learning rate (10 000 clients).

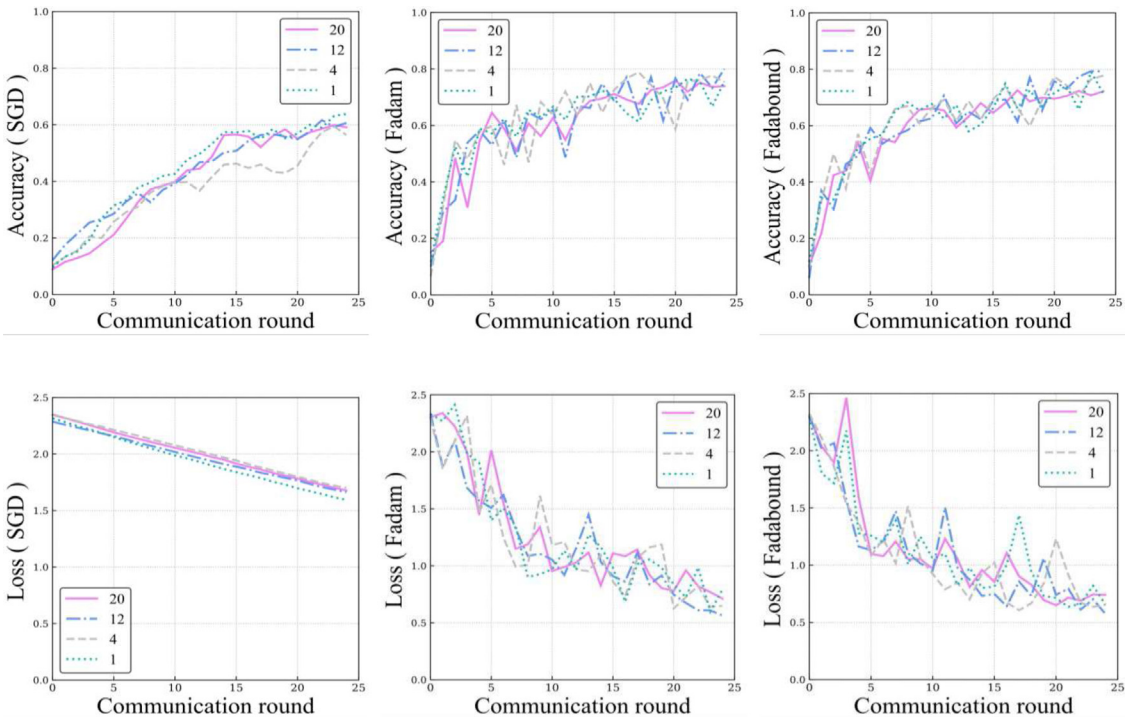


Fig. 7. The performance of different privacy budgets (1000 clients).

CRediT authorship contribution statement

Xiang Wu: Conceptualization, Schematic design, Formal analysis, Writing – review & editing, Supervision. **Yongting Zhang:** Investigation, Writing – original draft, Experimenting. **Minyu Shi:** Investigation, Writing – original draft, Writing – review & editing. **Pei Li:** Writing – software, Visualization, Writing – review & editing. **Ruirui Li:** Writing – experimenting, Writing –review & editing, Supervision. **Neal N. Xiong:** Writing – review & editing.

Declaration of competing interest

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant No. 62003291, the Xuzhou Science and Technology Project under grant No. KC20112, the Project of Philosophy and Social Science Research in Colleges and Universities in Jiangsu Province under grant No. 2020SJA1056, the Industry-University-Research Cooperation Project of Jiangsu Science and Technology Department under grant No. BY2018124 and the National Science and Technology Foundation Project under grant No. 2019FY100103.

References

[1] Q. Liu, Y. Peng, S.Y. Pei, J. Wu, T. Peng, G.J. Wang, Prime inner product encoding for effective wildcard-based multi-keyword fuzzy search, IEEE Trans. Serv. Comput. (TSC) (2020) http://dx.doi.org/10.1109/TSC.2020.3020688.

[2] X. Wu, H. Wang, M.Y. Shi, A.M. Wang, K. Xia, DNA motif finding method without protection can leak user privacy, IEEE Access (2019) 152076–152087.

[3] Q. Liu, P. Hou, G. Wang, T. Peng, S. Zhang, Intelligent route planning on large road networks with efficiency and privacy, J. Parallel Distrib. Comput. 133 (2019) 93–106.

[4] D. Riboni, Pareschi, JS-reduce: Defending your data from sequential background knowledge attacks, IEEE Trans. Dependable Secure Comput. 9 (2012) 387–400.

[5] C. Dwork, G.N. Rothblum, S.P. Vadhan, Boosting and Differential Privacy. Oct, Lsa Vegas, USA. The 51th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2010.

[6] X. Wu, Y. Wei, Y.Q. Mao, L. Wang, A differential privacy DNA motif finding method based on closed frequent patterns, Cluster Comput. (2018) 1–13.

[7] X.F. Wang, C.Y. Wang, X.H. Li, Federated deep reinforcement learning for internet of things with decentralized cooperative edge caching, IEEE Internet Things J. 7 (2020) 9441–9455.

[8] Q. Yang, Y. Liu, T. Chen, Federated machine learning: Concept and applications, Acm Trans. Intell. Syst. 10 (2019) 12.1–12.19.

[9] Q. Liu, Y. Tian, J. Wu, et al., Enabling verifiable and dynamic ranked search over outsourced data, IEEE Trans. Serv. Comput. (2019).

[10] Q. Liu, G.J. Wang, F.L.S.H. Yang, J. Wu, Preserving privacy with probabilistic indistinguishability in weighted social networks, IEEE Trans. Parallel Distrib. Syst. 28 (2017) 1417–1429.

[11] Y. Lu, X. Huang, Y. Dai, Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, IEEE Trans. Ind. Inf. 16 (2020) 4177–4186.

[12] Z. Li, V. Sharma, S.P. Mohanty, Preserving data privacy via federated learning: Challenges and solutions, IEEE Consum. Electron. Mag. 9 (2020) 8–16.

[13] N. Rodriguez-Barroso, G. Stipcich, D. Jimenez-Lopez, Federated learning and differential privacy: Software tools analysis, the Sherpa.ai FL framework and methodological guidelines for preserving data privacy, Inf. Fusion (2020) 270–292.

[14] Z. Yu, J. Hu, G. Min, Feb, Abu Dhabi National Exhibition Centre. Federated Learning Based Proactive Content Caching in Edge Computing. Presented at the 2018 IEEE Global Communications Conference (GLOBECOM) IEEE.

[15] Q. Liu, Y. Peng, J. Wu, T. Wang, G.J. Wang, Secure multi-keyword fuzzy searches with enhanced service quality in cloud computing, IEEE Trans. Netw. Serv. Manage. (TNSM) (2020) http://dx.doi.org/10.1109/TNSM.2020.3045467.

[16] Y. Qu, N. Xiong, RFH: A resilient, fault-tolerant and high-efficient replication algorithm for distributed cloud storage, in: The 41st International Conference on Parallel Processing, 2012, pp. 520–529.

[17] S. Niknam, H.S. Dhillon, J.H. Reed, Federated learning for wireless communications: Motivation, opportunities, and challenges, IEEE Commun. Mag. 58 (2020) 46–51.

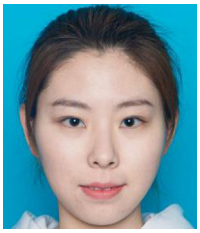
[18] Y. Liu, J.L. Peng, J.W. Wang, A secure federated learning framework for 5G networks, 27 (2020) 24–31.

[19] P. Florin, P. Butucaru, Maria, ARMCO: Advanced topics in resource management for ubiquitous cloud computing: An adaptive approach, Future Gener. Comput. Syst. (2016) 79–81.

[20] B. Yi, X. Shen, H. Liu, Z. Zhang, W. Zhang, S. Liu, N. Xiong, Deep matrix factorization with implicit feedback embedding for recommendation system, IEEE Trans. Ind. Inf. 15 (8) (2019) 4591–4601.
[21] M. Wu, L. Tan, N. Xiong, A structure fidelity approach for big data collection in wireless sensor networks, Sensors 15 (2015) 248–273.
[22] Jakub Konečný, H.B. Mcmahan, F.X. Yu, Federated Learning: Strategies for Improving Communication Efficiency, April, Vancouver, Canada. International Conference on Learning Representations.
[23] G. Wood, Ethereum: A Secure Decentralized Generalised Transaction Ledger, Ethereum Project Yellow Paper, 2014, pp. 1–32.
[24] T. Li, M. Sanjabi, V. Smith, Fair resource allocation in federated learning, 2019.
[25] S. Hardy, W. Henecka, H. Ivey-Law, Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption, 2017.
[26] J.W. Leng, M. Zhou, J.L. Zhao, Y.F. Huang, Y.Y. Bian, Blockchain security: A survey of techniques and research directions, IEEE Trans. Serv. Comput. (2020).
[27] R.C. Geyer, T. Klein, M. Nabi, Differentially private federated learning: A client level perspective, 2017.
[28] J.W. Leng, S.D. Ye, M. Zhou, J.L. Zhao, Q. Liu, W. Guo, W. Cao, L.J. Fu, Blockchain-secured smart manufacturing in industry 4.0: A survey, IEEE Trans. Syst. Man Cybern.: Syst. 51 (2021) 237–252.
[29] K. Xia, X. Wu, Y.Q. Mao, H. Wang, Secure DNA motif finding method based on sampling, ACM Trans. Internet Technol. (2020).
[30] M. Chen, H.V. Poor, W. Saad, Performance Optimization of Federated Learning over Wireless Networks, Dec, Waikoloa Village, USA. the 2019 IEEE GLOBAL COMMUNICATIONS CONFERENCE (GLOBECOM), 2019.
[31] S. Hua, K. Yang, Y. Shi, Sep, Hawaii, On-Device Federated Learning via Second-Order Optimization with Over-the-Air Computation. USA. The 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall). IEEE.
[32] T. Vu, N. Tron, N.H. Tran, Cell-free massive MIMO for wireless federated learning, IEEE Trans. Wireless Commun. 19 (2020) 6377–6392.
[33] M. Duan, D. Liu, X.Z. Chen, Self-balancing federated learning with global imbalanced data in mobile systems, IEEE Trans. Parallel Distrib. Syst. 32 (2021) 59–71.
[34] J. Konen, H.B. Mcmahan, D. Ramage, Federated optimization: Distributed machine learning for on-device intelligence, 2016, CoRR, abs/1610.02527.
[35] D. Ye, R. Yu, M. Pan, Z. Han, Federated learning in vehicular edge computing: A selective model aggregation approach, IEEE Access (2020) 23920–23935.
[36] X. Wu, H. Wang, D.S. Wei, M.Y. Shi, Dynamic allocation strategy of VM resources with fuzzy transfer learning method, Peer-to-Peer Netw. Appl. (2020).
[37] Y. Liu, Y. Kang, C.P. Xing, A secure federated transfer learning framework, IEEE Intell. Syst. 35 (2020) 70–82.
[38] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. (1948) 27.
[39] J.B. Gao, F.Y. Liu, J.F. Zhang, Information entropy as a basic building block of complexity theory, Entropy 15 (2013) 3396–3418.
[40] H.B. Mcmahan, E. Moore, D. Ramage, Fort APR, Lauderdale, Communication-efficient learning of deep networks from decentralized data, in: Presented at the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), 2017, pp. 1273–1282.
[41] R. Hu, Y.X. Guo, H.N. Li, Personalized federated learning with differential privacy, IEEE Internet J. 7 (2020) 9530–9539.
[42] B. Lin, F. Zhu, J. Zhang, J. Chen, X. Chen, N. Xiong, J.L. Mauri, A time-driven data placement strategy for a scientific workflow combining edge computing and cloud computing, IEEE Trans. Ind. Inf. 15 (2019) 4254–4265.
[43] F. Niu, B. Recht, C. Re, HOGWILD! A lock-free approach to parallelizing stochastic gradient descent, in: Advances in Neural Information Processing Systems, 2011, pp. 693–701.
[44] H. Li, J. Liu, R.W. Liu, N. Xiong, K. Wu, T. Kim, A dimensionality reduction-based multi-step clustering method for robust vessel trajectory analysis, Sensors 17 (2017) 1792.
[45] Q. Mercier, F. Poirion, J.A. Desideri, A stochastic multiple gradient descent algorithm, European J. Oper. Res. 271 (2018) 808–817.
[46] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, Comput. Sci. (2014).
[47] W. Fang, X. Yao, X. Zhao, J. Yin, N. Xiong, A stochastic control approach to maximize profit on service provisioning for mobile cloudlet platforms, IEEE Trans. Syst. Man Cybern.: Syst. 48 (2016) 522–534.
[48] Y.F. Zhou, M.C. Zhang, J.L. Zhu, A randomized block-coordinate adam online learning optimization algorithm, Neural Comput. Appl. 32 (2020) 12671–12684.
[49] R. Ward, X. Wu, L. Bottou, AdaGrad stepsizes: Sharp convergence over nonconvex landscapes, from any initialization, 2018.
[50] T. Kurbiel, S. Khaleghian, Training of deep neural networks based on distance measures using RMSProp, 2017.
[51] L. Luo, Y. Xiong, Y. Liu, Adaptive gradient methods with dynamic bound of learning rate, 2019.
[52] R. Ge, F. Huang, C. Jin, Escaping from saddle points-online stochastic gradient for tensor decomposition, Mathematics (2015).

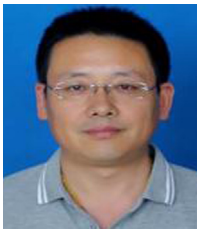
Xiang Wu received the B.Eng. degree in Information Engineering, the M.S. and Ph.D. in communication and information system all from China University of Mining and Technology, Xuzhou, China, in 2007, 2010 and 2014, respectively. He is currently pursuing a post-doctoral position with the School of Safety Engineering, China University of Mining and Technology, Xuzhou, China. His research interests include privacy protection and information security.
Email: wuxiang@xzhmu.edu.cn.

Yongting Zhang received the B.S. degree from Nanjing University of Chinese Medicine Hanlin College, Taizhou, China, in 2019. She is pursuing the M.S. degree in medical informatics with Xuzhou Medical University, Xuzhou, China. Her research interest includes privacy protection and information security.
Email: 301910911596@stu.xzhmu.edu.cn.



Minyu Shi received the B.S. degree in management from Shanxi Medical University, Taiyuan, China, in 2018. She is currently pursuing the MS degree in medical informatics, Xuzhou Medical University, Xuzhou, China. Her research direction is medical privacy protection.
Email: 301810911545@stu.xzhmu.edu.cn.

Pei Li received B.Eng. degree in mechanical and electrical engineering from Jiangsu Normal University in 2003, the M.S. in computer technology from China University of Mining and Technology in 2013, XuZhou, China. His research interests include medical information and information security.
Email: lipei@xzhmu.edu.cn.



Ruirui Li once worked in internationally renowned companies, including German WOLF brand, STORZ, etc., and has accumulated rich experience in medical imaging. In 2010, he registered the GEIWRE brand and began to devote himself to researching medical imaging system and practical applications.
Email: geiwre@hotmail.com.

Neal Naixue Xiong received the Ph.D. degree from the Japan Advanced Institute of Science and Technology. Before he attends SWOSU, he worked in Colorado Technical University (Full Professor, four years), Wentworth Institution of Technology, and Georgia State University for many years. He is currently with the Department of Business and Computer Science, Southwestern Oklahoma State University (SWOSU), OK, USA. His research interests include cloud computing, business networks, security and dependability, parallel and distributed computing, and optimization theory. He has published over 100 international journal articles and over 100 international conference articles.
Email: xiongnaixue@gmail.com.