Computers, Materials & Continua
DOI: 10.32604/cmc.2023.034571
Article

Tech Science Press

Research on Federated Learning Data Sharing Scheme Based on Differential Privacy

Lihong Guo*

Department of Information and Communications Engineering, Nanjing Institute of Technology, Nanjing, 211167, China
*Corresponding Author: Lihong Guo. Email: guolihongnj@163.com

Received: 20 July 2022; Accepted: 03 October 2022

Abstract: To realize data sharing, and to fully use the data value, breaking the data island between institutions to realize data collaboration has become a new sharing mode. This paper proposed a distributed data security sharing scheme based on C/S communication mode, and constructed a federated learning architecture that uses differential privacy technology to protect training parameters. Clients do not need to share local data, and they only need to upload the trained model parameters to achieve data sharing. In the process of training, a distributed parameter update mechanism is introduced. The server is mainly responsible for issuing training commands and parameters, and aggregating the local model parameters uploaded by the clients. The client mainly uses the stochastic gradient descent algorithm for gradient trimming, updates, and transmits the trained model parameters back to the server after differential processing. To test the performance of the scheme, in the application scenario where many medical institutions jointly train the disease detection system, the model is tested from multiple perspectives by taking medical data as an example. From the testing results, we can know that for this specific test dataset, when the parameters are properly configured, the lowest prediction accuracy rate is 90.261% and the highest accuracy rate is up to 94.352. It shows that the performance of the model is good. The results also show that this scheme realizes data sharing while protecting data privacy, completes accurate prediction of diseases, and has a good effect.

Keywords: Federated learning; C/S mode; differential privacy; gradient descent; prediction accuracy

1 Introduction

It is said that the past decade is the “Internet 2.0 era” which was brought about by the mobile Internet, and mobile and big data have brought huge changes to people’s lifestyles. The QR code during the epidemic is a case of the widest impact of big data applications on disease prevention and control. By analyzing the LBS (Location Based Services) data of individual cell phones, operators can map out the travel addresses of individuals and give red code alarm alerts whether they are in high-risk areas. If separated from big data, it is obvious that such efficient social order governance cannot be achieved. Now people have long been accustomed to the convenience brought by big data analysis and use, which profoundly affects people's lives [1]. At present, the rapid development of the Internet and computer technology, makes all kinds of data flow, and these flowing data break the restrictions of time and space and accumulate in different fields to form data treasures, which breed huge commercial value and unlimited potential. Therefore, with big data as the background and data sharing as the foundation, the intelligent use of data and the maximization of data value have become the focus of competition in various industries.

1.1 Privacy Security and Data "Island"

In the face of the complex network environment, people must first ensure the security of users' data while pursuing the value of data. Therefore, how to protect data security and prevent privacy leakage has become a major challenge at present. In the past, different industries and departments focused on data security, mainly to solve their own data storage security and intrusion prevention problems, establishing their data "island", data barriers exist between industries and departments, and data cannot be shared safely. However, with the development of AI (Artificial Intelligence), it has been found that the process of data circulation and usage requires multi-dimensional data, whether in terms of the number of data or the characteristic dimensions and requires the collaborative sharing of data among multiple organizations. But each organization is unwilling to disclose too much data due to privacy and security issues, which has led to lacking available data for AI enterprises, especially in the analysis and processing of big data. Due to the lack of data volume and feature dimensions, the global optimum cannot be achieved, which limits the development of AI [2].

1.2 Privacy Computing and Data Sharing

With the cross-border integration of data and the need for collaborative sharing, data security has expanded from "defense-oriented" to "circulation-oriented". At present, the core technology for data circulation is privacy computing, which solves the security problem of data circulation from enterprise to enterprise. Privacy computing refers to a type of information technology that implements data analysis while protecting the data itself from external leakage and is divided into two main areas: trusted hardware and cryptography. Trusted hardware refers to the trusted execution environment, and the core idea is to build a hardware security environment in which data is computed only within that security region. Trusted hardware is represented by Intel-SGX, ARM-TrustZone, Ucloud-safe house, etc., [3]. Cryptography technology is currently represented by MPC (Multi-Party Secure Computing), it offers cryptographically strong guarantees on the secrecy of data used in collaborative computing among untrusted parties, and each participating entity cannot get any input information from other participating entities except the computation result [4].

1.3 Data Security Sharing Scheme

As digital technology enters a period of rapid development, data diversification, informatization, and diversification become the theme of era. As a result, various data security sharing schemes have emerged, in summary, which is divided into two modes.

(1) Centralized Processing Mode

For the storage and processing of massive data, people rely more on cloud services, which is a typically centralized data processing mode. This centralized training approach is to upload all the client's data to the cloud server, and the model deployed on the server is trained based on the uploaded data [5]. This model has two drawbacks.

1) It is impossible to guarantee the data privacy of clients. The service provider collects all the client's data on the server for unified management. This approach will be increasingly restricted under the increasingly strict regulatory control of personal data privacy.
2) Real-time performance is difficult to guarantee. In practical applications, the cloud model needs to be requested through the network. In the case of network delay or no network, the model cannot play its role.

The centralized processing mode is not only inefficient but also prone to additional bandwidth overhead and network latency. At the same time, the demand for data privacy protection is becoming increasingly apparent, which undoubtedly reduces the possibility of sharing data among different entities.

(2) Distributed Learning Mode

To solve the drawbacks of the above-mentioned centralized processing mode and effectively solve the problem of data island, Google has proposed a new "distributed model training", i.e., federal learning model. In this model, user data does not leave the local area, and all model training is performed locally on the device. After the local model is trained, the model parameters are uploaded to the cloud, where the cloud model receives and combines all the parameters for a unified aggregation, and then re-distributes the new results to the local level, where a new model is updated [6].

Federated learning is essentially distributed machine learning, and its most important feature is that user data is stored locally by the user so that the original data of each participant is not leaked during the cooperation training of the model. Federated learning organizes the process of model training through distributed, and the whole training process only moves the model, not the data, so that modeling using the framework of federated learning can ensure that multi-institutional data are jointly modeled under the premise of security without revealing privacy [7,8].

In this paper, based on the distributed learning model, medical data is used as the prototype data, and a differential privacy-based data sharing scheme is studied and designed through federated learning from feasibility and effectiveness. This scheme can protect the clients' private data and realize secure collaborative data sharing.

1.4 The Contribution and Organization

In summary, the contributions of this paper are as follows:

(1) Propose a data security sharing scheme based on the C/S communication mode, and construct a federated learning architecture that uses differential privacy technology to protect training parameters.
(2) Introduce a distributed parameter update mechanism in the process of training, and the server is responsible for issuing training commands and parameters, and aggregates the local model parameters uploaded by the client. And the client uses the stochastic gradient descent algorithm for gradient trimming and updating.
(3) Taking medical data as the test dataset, a series of experiments are designed to verify the performance of the scheme, and the influence factor is analyzed from the predicted results.

Organization: The rest of the paper is organized as follows. In Section 2, this paper discusses and summarizes the related work. Related preliminaries are provided in Section 3. Section 4 describes the construction, and then the implementation and performance are shown in Section 5. Finally, Section 6 concludes the paper and discusses future directions.

2 Related Work

The development of data security sharing mode is closely related to the development of network environment, cloud computing, and AI. In the early days, with no network environment, people focused on local data storage security and managed security. With the popularity of the Internet and the rapid development of information technology, more and more companies and organizations began to use the Internet and mobile communication systems to deal with various information. In the face of a large amount of data, people hosted data storage and computing to a cloud server to save local storage space and arithmetic power. However, users lose control of outsourced data because of the centralized cloud storage, and they suffer from low efficiency, data leakage, and other problems. Therefore, how to ensure the security of sharing data in the cloud is an urgent task, so data encryption and access control become the focus in this period.

However, with the rapid development of the Internet and the deepening of informationization, the world has crossed into the era of big data. Due to the singularity of user data and the limitation of processing capacity, data sharing has become an important way of data utilization, which can expand the scale of data and improve the efficiency of data mining. In the Global Internet Trends Report released in 2018, data sharing has become an inevitable trend in the development of the Internet and big data. For the sharing of ordinary data, distributed database storage and blockchain on-chain records can be used to directly make the shared data public [9]. The sharing of high-privacy data, it involves the security management of data during the sharing process, which includes anonymous sharing, encrypted storage, ciphertext search, threshold access, provable security, permission security, etc., [10]. All of them require cryptography technology to ensure the security of sharing data. Therefore, attribute encryption, blockchain technology, and various centralized data sharing protection schemes emerged in this period.

Furthermore, data sharing can increase the value of data, and in the process of the human pursuit of intelligence, AI as a theory of intelligence has widely penetrated various fields of economic, political, cultural, social, etc. AI needs big data as the basis for "thinking" and "decision-making", and the rapid development of AI has given rise to a series of emerging applications of machine learning algorithms, among which federated learning is a hot spot for the current research. Federated learning has recently attracted a lot of attention from the academic community. It was first proposed by Google in 2016 and was originally used to solve the problem of updating models locally by android phone users. The goal of the design is to carry out efficient machine learning among multiple participants under the premise of guaranteeing information security. This technology is a kind of distributed cryptographic technology where all participants can share the underlying data. Its most important feature is to keep the data in the local area so that the original data of each participant will not be leaked during the cooperation training. Now, it has shown strong vitality and good prospects in more and more scenarios.

In February 2019, Webank made public FATE (Federated AI Technology Enabler) architecture, a high-performance, privacy-secure computing framework that provides a solution platform for common industrial applications [11]. In the past two years, besides the Federated Bank research team leading the promotion of federated learning, there is also the PingAn Technology R&D team, which mainly focuses on the research of privacy security and federated incentives for federated learning, and Jingdong Digital AI Lab for the research of asynchronous federated learning. As the concept of federated learning has become popular, the applications of a federated learning [12–15] have been gradually developed. In the federated learning scenario, clients do not need to share local data, and they only need to upload the trained model parameters. However, using model parameters as an interactive medium, there may lead to data privacy leakage during the learning process. At present, established federal learning works [16,17] show that private information may still be leaked when the model parameters of distributed users are uploaded piecewise with the model structure. Based on the model parameters uploaded by each user, the original data owned by the local user can be inferred. What is more serious is that when an attacker who may be an honest but curious server or a malicious client, or a malicious third party directly intercepts the model parameters uploaded by each client, he can further infer the victim's private information, so the model parameters also need protection.

Based on the research of the above-related literature, we find that centralized data sharing schemes have the risk of user's local data leakage. In addition, its security depends on the centralized server, and there are certain hidden safety risks. Distributed data sharing mode is an inevitable trend and has a good prospect in more and more applications, especially since the use of federated learning is increasing. However, using model parameters as the interaction medium in a federated learning model still suffers from the problem of data privacy leakages. So in this paper, a federated learning data sharing scheme was designed based on differential privacy. The federated learning model is used to ensure the security of local data and differential privacy is used to protect the security of model parameters in training.

3 Preliminaries

To clarify the scheme proposed in this paper, some relevant theoretical knowledge needs to be introduced here.

3.1 Differential Privacy

Differential privacy is a new definition of privacy proposed by Dwork in 2006 in response to the problem of privacy leakage in statistical databases [18]. Under this definition, the results of computational processing in a database are insensitive to the changes in a specific record, and the presence or absence of a single record in a dataset has a negligible effect on the computational results. Therefore, the risk of privacy disclosure is kept within a very small and acceptable range, and an attacker cannot obtain accurate information about an individual by observing the computation results. In practice, there are two methods commonly used. One is the Laplace mechanism applied to numerical output, which adds the noise of the Laplace distribution to the query results. And the other one is the exponential mechanism applied to non-numerical output, which adjusts the probability with exponential distribution in the query result.

(1) (ε, δ)-differential privacy

For two data sets D and D' that differ by only one record, a randomized algorithm O, and for any output $S \subset Range(O)$, only when it satisfies Eq. (1).

$$P_r[O(D) \in S] \leq P_r[O(D') \in S] \times e^{\varepsilon} + \delta \tag{1}$$

Here, to claim that the randomized algorithm O provides (ε, δ)-differential privacy protection, where ε denotes the privacy budget and δ denotes the failure probability. When $\delta = 0$, the (ε, δ)-differential privacy protection with better performance is obtained.

(2) The probability density function of Laplace distribution

The Laplacian mechanism achieves ε-differential privacy protection by adding random noise obeying the Laplace distribution to the exact query result. To make the query result satisfy the requirement of differential privacy, a random noise η satisfying Laplace distribution is added to the query result to obtain the noise-added query result $f(D) + \eta$. The probability density function of Laplace distribution in Eq. (2).

$$p(\eta) = \frac{1}{2\lambda} e^{-\frac{|\eta|}{\lambda}} \tag{2}$$

From Eq. (2), Laplace distribution has a mathematical expectation of 0 and a variance of $2\lambda^2$. The Laplace noise parameter λ indicates the magnitude of added noise, and this parameter defines the strength of privacy protection, and the larger the value of λ, the larger the noise magnitude and the higher the strength of differential privacy protection.

(3) The global sensitivity

Given a function set F, if every function query results in the function set is a real number, the sensitivity of F is defined as Eq. (3).

$$S(F) = \max_{T,T'}\left(\sum_{f \in F} |f(D) - f(D')|\right) \tag{3}$$

where D and D' are any pair of sibling data tables. The weakness of differential privacy is obvious: it needs to include a lot of randomization in the query results, however, because of too strong assumptions about background knowledge it leads to a sharp drop in data usability. Especially for those complex queries, sometimes the randomization results almost obscure the real results. Currently, differential privacy can be applied to recommendation systems, social networks, location-based services, Apple's input system, etc.

3.2 Machine Learning Algorithms

(1) Linear regression algorithm

Linear regression [19] is the simplest basic type of supervised learning model, and it is the basis for many complex models. Linear regression has to deal with a class of problems: given a set of input samples and the target value corresponding to each sample, it is necessary to find (learn) the functional relationship between the target value and the input value under a certain loss criterion, so that when a new sample arrives, it can predict what the corresponding target value is. Linear regression and linear classification are very similar but differ in that the target value for linear regression is a continuous variable, and the target value for linear classification is a discrete variable.

The linear regression model predicts the label value y by making a linear combination of the eigenvalues $X{=}(x_1, x_2, \ldots x_n)$, i.e., satisfying Eq. (4).

$$y = w_1x_1 + w_2x_2, \ldots + w_nx_n + b \tag{4}$$

It is usually expressed in a simplified form using vectors as Eq. (5).

$$y{=}W^TX + b \tag{5}$$

where $W{=}(w_1, w_2, \ldots w_n)$, $X{=}(x_1, x_2, \ldots x_n)$.

(2) Logistic regression algorithm

Logistic regression [20] is the most commonly used binary classification algorithm, which belongs to the family of generalized linear models. It is widely used because of its simplicity and good results.

The y-value obtained using Eq. (5) is a continuous value, and the output of the dichotomy method is a discrete value containing only 0 and 1. For this reason, a nonlinear mapping can be performed on top of the continuous value output of Eq. (5), i.e., a differentiable nonlinear function f is found to relate the discrete label value y, and the predicted continuous value of linear regression is shown in Eq. (6).

$$y=f\left(W^{T}X+b\right) \tag{6}$$

In logistic regression, the logistic function is generally used to act as this nonlinear mapping, and the logistic function is expressed in the form of Eq. (7).

$$f\left(z\right)=\frac{1}{1+e^{-z}} \tag{7}$$

When using logistic regression for classification prediction, if the prediction value of linear regression $W^{T}X+b\geq 0$, then it is judged to be a positive case and the output is 1; otherwise, it is judged to be a negative case and the output is 0.

(3) Random gradient descent algorithm

The gradient descent algorithm [21] is currently one of the most used algorithms in machine learning. The gradient is a vector that represents the direction of weights, or more precisely, how to change the weights so that the loss changes the fastest. This research called this process a gradient descent because it used the gradient to bring the loss curve down to a minimum value. The core of random gradient descent is that the gradient is the mathematical expectation, and the expectation can be estimated using a small sample size approximation. At each step of the algorithm, we draw a small batch of samples $B=\left\{x^{(1)},\ldots x^{(\mathrm{m}')}\right\}$. When the size of training set m grows, m' is usually fixed. With one sample at each update, i.e., one example in the sample is used to approximate all samples to adjust θ, as shown in Eq. (8).

$$\theta_{j}=\theta_{j}+\alpha\left(y^{(i)}-h_{\theta}\left(x^{i}\right)\right)x^{j} \tag{8}$$

Therefore, random gradient descent cannot exactly find an optimal gradient, and in the optimal case, its loss function is not always in the direction of the optimum, but in the direction of the global optimum.

3.3 The Aggregation Algorithm in Federated Learning

Federated learning differs from general machine learning models in that the main function on the server side is to perform model aggregation, so some typical model aggregation algorithms are needed to update the global model by accepting models uploaded by the client using the aggregation algorithm. There are two typical federated model aggregation algorithms: FedAvg and FedProx [22–24].

(1) FedAvg algorithm

The algorithm is the most fundamental gradient aggregation method in the field of federation learning. Compared to traditional distributed machine learning methods that only compute the gradient on the client side, the FedAvg method expects the client side to do more operations to get a better descent direction than the gradient. Since this descent direction is better than the gradient, it can converge faster. If the convergence is faster, then the number of communications is naturally less. The essence of the FedAvg idea is that the client uses a random gradient descent algorithm to get the weight parameters, and the server integrates each user's trained weights for averaging, as shown in Eq. (9).

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^{k} \tag{9}$$

(2) FedProx algorithm

The FedProx algorithm focuses on improving FedAvg from two directions: system heterogeneity and statistical heterogeneity; on the one hand, different devices have different computational capabilities, and simple iteration will overstress some devices; on the other hand, we want to keep the local model from deviating the global model, and it will affect the convergence of the global model.

The local iterations of epochs performed by each node may not be guaranteed, so adding a proximal term in the optimization objective function of the client. It makes the optimization algorithm more stable and ultimately, makes FedProx converge faster even under statistical heterogeneity. The equation is shown in Eq. (10).

$$Q^{t+1}{=}Q^{t} + \lambda \sum_{i=1}^{m} \left(L_i^{t+1} - Q_i^{t}\right) \tag{10}$$

where Q^t denotes the global model parameters of the t^th^ round of aggregation, Q_i^{t+1} denotes the model of the i^th^ client after the $t+1$ round of local update, and Q^{t+1} denotes the global model after the $t+1$ round of aggregation.

4 A New Data Security Sharing Scheme

In this paper, the research proposed a new data security sharing scheme based on federated learning and differential privacy technology. Different from the previous centralized data sharing scheme, here, all the training data was local and belonged to the client, the C/S mode was used for cooperation training of the model. The server was responsible for issuing training commands, while the client was responsible for training and uploading the training parameters. In the end, all participants shared the cooperation training results to implement data sharing.

In this part, the paper takes the medical data as the training data, and hopes to expand the training sample space with the participation of multiple parties, thus, improving the prediction accuracy.

4.1 Our Construction

In this part, the search built a federated learning model based on differential privacy that not only considered the privacy security at the data level but also considered the security issues at the client level. This model not only ensures the privacy security of local data in each client, but also ensures the information security between clients, that is, the server receives the training parameters of the local model from the client, and can neither determine which client uploaded it. It is also not possible to infer whether a client is participating in the current federated training. The architecture of the model is shown in Fig. 1.

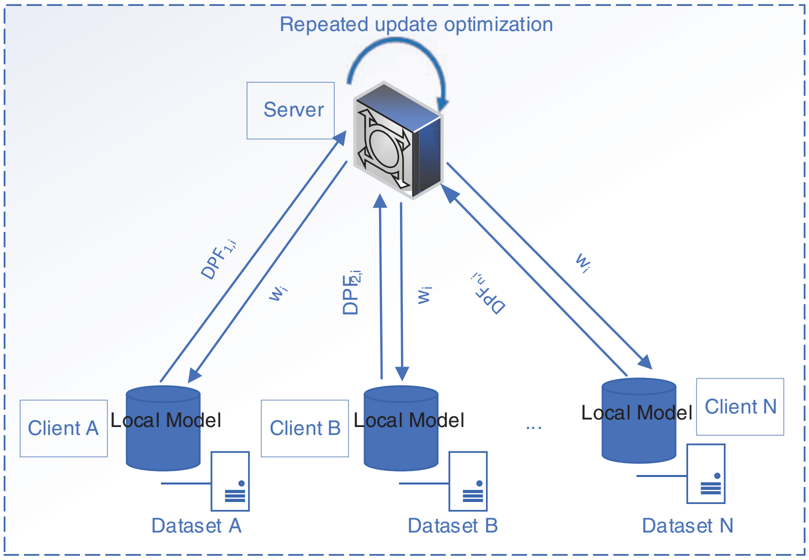


Figure 1: The architecture of the model

The architecture of the model is mainly composed of a server and multiple clients. Each client uses its local dataset for training, and they have its local models. In the process of training, the server sends the training command and weight to every participant, and the clients participating in training upload the trained parameter to the server. On the server side, it estimates the training performance. If the result meets the training requirements, it will end the process, and further provide the predicted result of the test example.

4.2 Implementation of Data Sharing Scheme

Here, the medical data was taken as the sharing data. To realize data sharing under the premise of protecting data privacy, we use horizontal federated learning technology, so each participant has the same data features [25–28].

4.2.1 The Pre-processing of Data

To realize the differential privacy-based federated learning, the paper used an open-source dataset from Wisconsin Center for Scientific Research [29]. For the convenience of processing, here we preprocess the data features and extract 30 main features. The information after feature extraction is shown in Table 1, and the 31st column represents the label data (The value 1 represents the tumor, and the value 0 represents the malignant tumor).

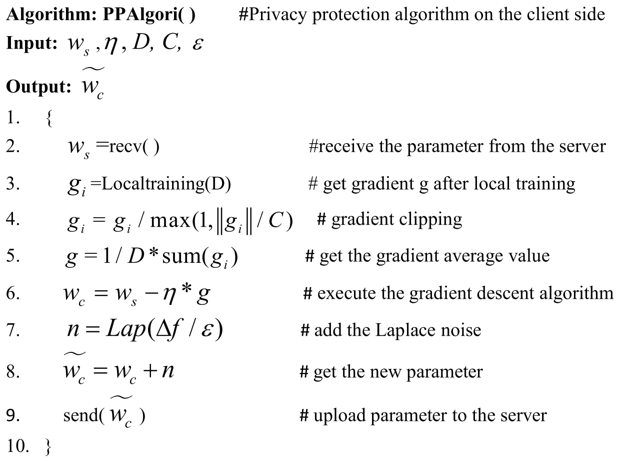
Since some feature values are greater than 100, and some feature values are less than 1, the dataset is first normalized and preprocessed to reduce the dimension and difference of each value. Data standardization mainly scales the value of each dimension according to a certain proportion, so that it falls into a specific interval so that the feature value of different units or magnitudes can be weighted and compared.

Table 1: The main features of medical data

F1	F2	F3	F4	F5	F6	...	31_label
1.0961	−2.07151	1.268817	0.98351	1.567087	3.280628	...	0
1.828212	−0.35332	1.684473	1.90703	−0.82624	−0.48664	...	0
1.578499	0.455786	1.565126	1.557513	0.941382	1.052	...	0
−0.76823	0.253509	−0.59217	−0.76379	3.280667	3.399917	...	1
1.748758	−1.1508	1.775011	1.824624	0.280125	0.538866	...	0
...	...	...	...	...	...	...	...

4.2.2 The Design of the Algorithm

In this scheme, on the server side, a federated aggregation algorithm is used for model aggregation, while on the client side, a gradient clipping algorithm based on differential privacy is used. The detailed steps on the client side are shown in Fig. 2.



```
Algorithm: PPAlgori( )      #Privacy protection algorithm on the client side
Input: $w_s, \eta, D, C, \varepsilon$
Output: $\widetilde{w_c}$
{
    $w_s$ =recv( )                              #receive the parameter from the server
    $g_i$ =Localtraining(D)                     # get gradient g after local training
    $g_i = g_i / \max(1, \|g_i\| / C)$          # gradient clipping
    $g = 1/D * \mathrm{sum}(g_i)$               # get the gradient average value
    $w_c = w_s - \eta * g$                      # execute the gradient descent algorithm
    $n = Lap(\Delta f / \varepsilon)$           # add the Laplace noise
    $\widetilde{w_c} = w_c + n$                 # get the new parameter
    send( $\widetilde{w_c}$ )                   # upload parameter to the server
}
```

Figure 2: Gradient clipping algorithm on the client side

The specific steps of the privacy protection algorithm on the client side are as follows: First, in the normal communication environment, the client receives the training parameters (or weights as they are called) from the server side, and it will start the local training according to the training requirements. Secondly, the client gets the gradient after local training, and then it will perform the gradient clipping. Thirdly, after several rounds of local training, it will calculate the average value of the gradient, and then execute the gradient descent algorithm to get the trained parameter w_c. Next, to ensure the security of the parameter, the Laplace noise is introduced by the Lap function and used in the trained parameter. Finally, the model parameter $\widetilde{w}_c$ is uploaded by the client to the server, which can effectively prevent the differential attack.

In the above algorithm, some variables or parameters are described in Table 2, and $||g_i||$ is the norm of the gradient. Gradient clipping ensures the maximum norm of the gradient vector, which can help gradient descent and keep it reasonable even when the model's loss function is irregular.

Table 2: The description of variables or parameters in the algorithm

Variables or parameters	Description
w_s	Initial parameter from the server
η	Learning rate
C	Hyperparameter
ε	Privacy budget
D	Dataset
g	The gradient
g_i	The gradient of the client i
w_c	The parameter of gradient descent
$\widetilde{w}_c$	The parameter of adding LAP noise

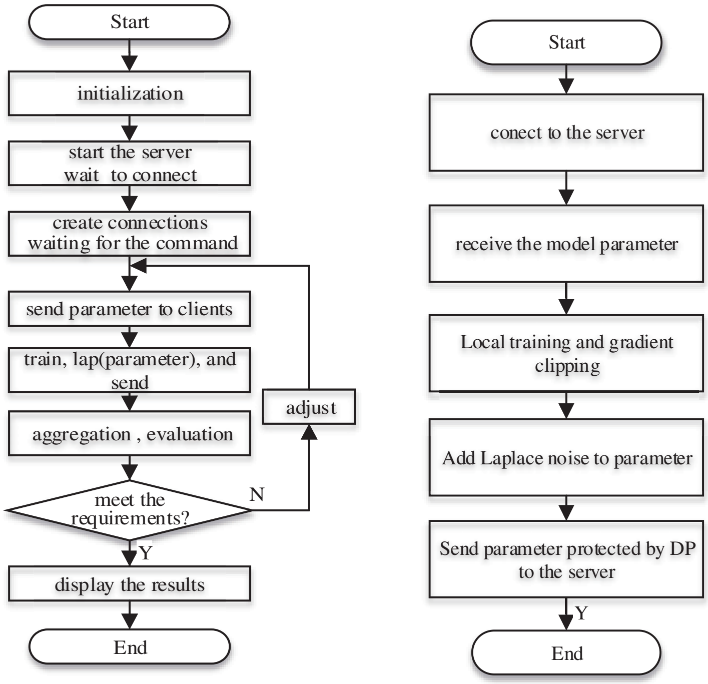
$\widetilde{w}_c$ is the encrypted model parameters uploaded by the participating clients to the server, which can effectively prevent the differential attack, and the added noise has little impact on the overall model prediction performance. The reason for gradient clipping of model parameters is that g is a value that cannot be fixed without clipping, that is, the range of its first norm cannot be determined, and the range of the first norm cannot be determined, which means that the sensitivity cannot be calculated. If the sensitivity cannot be calculated, then the differential noise cannot be added, because sensitivity is a very important parameter for differential privacy. According to the definition of adjacent datasets and the sensitivity, it can be known that the norm of gradient clipping should be limited to a certain range. According to the derivation: $|g - g'| \leq \frac{2C}{|D|}$, the global sensitivity Δf is: $\Delta f = \eta * \frac{2C}{D}$.

The difference between differential federated training and federated training is that gradient clipping and noise processing are performed on the client. The added Laplace noise can not only prevent the differential attack but also further protect the privacy of the user gradient.

4.2.3 The Execution Process

In this part, the server and the clients are based on the C/S communication mode. The whole execution process of the scheme is shown in Figs. 3a and 3b is the execution process of the client.

The whole execution process is described as follows: First, as the initialization of a system, it mainly finishes the initialization of variables and the creation of a socket. And then, the server starts the listening function and waits for the connection of the clients. Further, the server sends the training commands and parameters to all the participating clients. On the client's side, they perform the train, gradient clipping, and differential privacy transform, etc. Later, the clients upload the parameters added the Laplace noise to the server, and finish the first aggregation and evaluation. If it hasn't met the training requirements, then repeated the process. It includes the sending, training, adding noise and uploading parameters, etc. In the process of training, the client needs to train many rounds in the local area, and the client and server need to go through many rounds of communication and interaction until it meets the requirement of training.



(a) The whole execution process (b) The execution process of the client

Figure 3: (a) The whole execution process (b) The execution process of the client

The execution process of the client is below: First, on the client side, the client receives the model parameter and the start command for local training sent by the server. Secondly, the client starts the local training, and after getting the trained parameter, it will execute the gradient clipping and add the Laplacian noise to the trained parameter. Finally, the client uploads the parameter to the server for parameter aggregation and performance evaluation. If the training requirements are not met, the above process is repeated; if the requirements are met, the training objectives are completed.

5 Test Experimental Results and Discussion

To assess and test the performance of the data sharing scheme, a serials of experiments is conducted. The test is in Windows 10 operating system with Inter (R) Core(TM) i5-6500 CPU 3.20 GHz and 16 GB RAM. PyCharm is used as the integrated development environment, and python is used as the programming language. In addition, many third-party libraries are installed, such as sklearn, pytorch, numpy, etc. The test dataset is from the Wisconsin Center for Scientific Research [29].

5.1 The Impact of the Privacy Budget

This experiment mainly tests the impact of the privacy budget on model performance, and sets the privacy budget as 0.2, 0.5, and 1.0, respectively. The other test parameters are: the global training epoch is 30, the learning rate is 0.01, and the batch size is 64.

The test result is shown in Fig. 4, the abscissa and the ordinate represent the global training epoch and the prediction accuracy, respectively. When the global training epoch is small, the smaller the privacy budget, the worse the prediction accuracy. Because the privacy budget gets smaller, it equates to an increase in Laplacian noise, which can make the accuracy worse and worse. In conclusion, when the privacy budget is 0.5, it has good results in terms of smoothness, accuracy, etc. When the global training epoch continues to gradually increase, the privacy budget has little impact on the model performance, and it also proves the reliability of this model proposed in this paper.

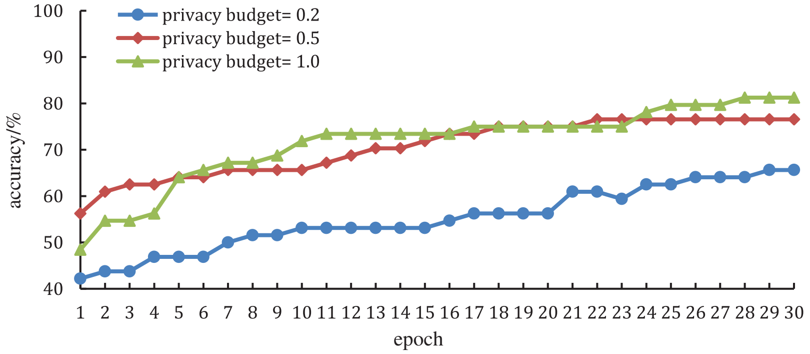


Figure 4: The impact of the privacy budget

5.2 *The Effect of the Learning Rate*

In this part, we mainly test the effect of learning rate on model performance. Under the same test conditions, we compare the performance of the model when the learning rates are 0.01, 0.015, 0.02, and 0.03, respectively. The other test parameter is: the global training epoch is 30, the privacy budget is 0.5, and the batch size is 64.

From the result shown in Fig. 5, we can know that when the learning rate is large, the model can reach the convergence value faster, that is, the prediction accuracy changes faster with the increase in training times. In addition, since the global sensitivity of differential privacy is proportional to the learning rate, the higher the learning rate, the smaller the noise, and the more stable the training, and the model performance is better.

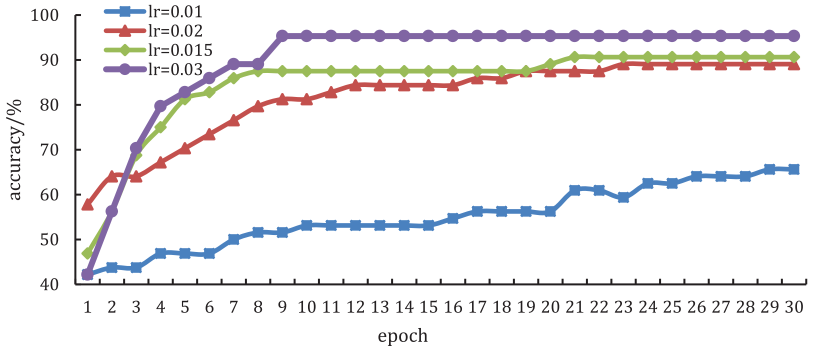


Figure 5: The effect of the learning rate

5.3 *The Effect of Global Training Epoch*

In this part, we set the learning rate to 0.01, the privacy budget to 0.5, and the batch size to 64. Comparing the performance of the model when the global training epoch is 10, 50 and 100, respectively. The test result is shown in Fig. 6. For the best and fastest training result, the parameter is global_epoch = 50 and batch_size = 64.

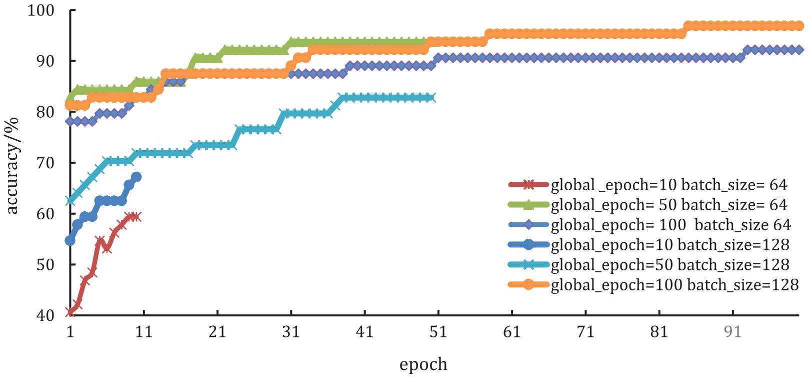


Figure 6: The effect of global training epoch

From the result, we can know that as the global training epoch gradually increases, the influence of noise on the performance is gradually decreased. Therefore, there is a similar conclusion, that is: as the global training epoch increases, the prediction accuracy is also getting better and better. When the training epoch reaches a certain value, the effect of added noise on the performance can be almost ignored.

5.4 *The Effect of Batch Size*

In this experiment, the search set the global training epoch to 30, the learning rate to 0.01, and the privacy budget to 0.5, and compare the performance when the batch sizes are 32, 64, 128, and 192, respectively.

Essentially, federated learning based on differential privacy adds the noise to the weight (training parameter), and adding noise to the weight itself is a common way to prevent model overfitting, that is to say, it is equivalent to adding a regularization term. From the equation of global sensitivity we can know: $\Delta f = \eta * \frac{2C}{D}$, where C is the privacy budget, its value is C = 0.5, so $\Delta f = \eta * \frac{1}{D}$, we can know that: with the gradual increase of batch size D, there is a decreasing trend in the global sensitivity, that means the Laplacian noise gradually decreases.

From the test result in Fig. 7, we can know that when the batch size is small, it is equivalent to adding less noise. Although the prediction accuracy is better in the first few rounds, due to the local optimum phenomenon during the training process, sometimes it occurs the overfitting phenomenon. For example, when the batch size is 192 and the training epoch is small, its prediction accuracy is low, which means the model performance is greatly affected by the noise of the first few rounds. However, when the training epoch reaches a certain value, the effect of the added noise on the model performance is almost negligible, the model is not prone to overfitting, and the prediction accuracy increases.

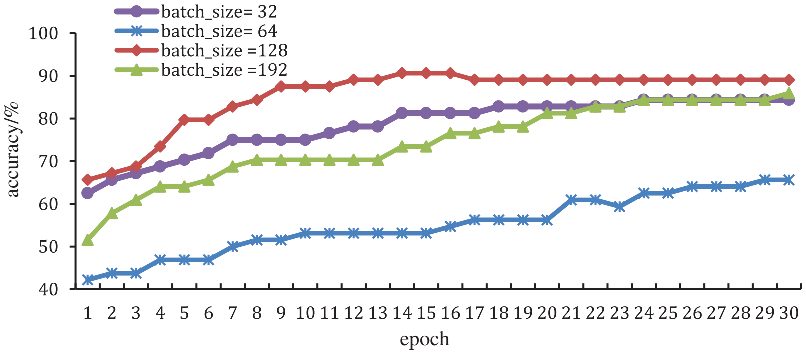


Figure 7: The effect of batch size

5.5 The Analysis and Discussion

Combining the above experimental results, we properly adjust the parameter values. After many rounds of testing, for this test dataset, the parameters are configured in the following way: privacy budget is 0.5, the learning rate is 0.015, and batch size is 128, when the global training epoch is 50, the lowest prediction accuracy rate is 90.261% and the highest accuracy rate is up to 94.352. Although there are some fluctuations, it is relatively steady. The experimental results fully demonstrate the validity of the scheme.

The implementation process of this data sharing scheme is a process of parameter tuning and finding the best balance, especially the introduction of noise. Therefore, in a sense, adding the differential noise of the Laplacian mechanism to federated learning can effectively prevent the model from overfitting, and at the same time, its communication cost or computational cost is smaller compared with other traditional privacy-preserving technologies. That is, the differential privacy technology controls the global sensitivity by controlling the batch size, and then controls the size of the added noise. Adding noise can not only protect the gradient privacy, but also it can prevent the federated learning model from overfitting, but it comes at the cost of reducing the prediction accuracy of the model in previous rounds [30].

6 Conclusions

Due to the multi-source heterogeneity of distributed user data, the contradiction between data privacy protection and data availability is difficult to balance. To a certain extent, it leads to the existence of data “island”. Federated learning, as a way to protect privacy, is a good solution to the problem of data sharing under the premise of privacy protection. So in this paper, we design and implement a data security sharing model based on federated learning and differential privacy mechanism.

First, to achieve a balance between data availability and data privacy protection, we design a kind of data sharing model based on federated learning, thereby mapping distributed raw data into data models. By analyzing and processing the data model, the original data is kept locally, which significantly reduces the risk of data leakage.

Secondly, we design the architecture of the data sharing model based on federated learning, and it avoids the transmission of original data and improves the privacy protection of client data.

Further, to prevent parameter leakage during federated learning, differential privacy technology is introduced to protect model parameters in communication. By introducing localized differential privacy and using a distributed parameter update mechanism in the training process, privacy protection of model parameters is achieved. This research builds a data model based on federated learning, converts the original data-oriented computational analysis process into a data model-oriented processing process, and it sinks the training to the user side, which can effectively improve the privacy protection of original data.

Finally, a series of experiments was practiced to evaluate the performance. The testing results indicated that for this specific test dataset, when the parameters are properly configured, the lowest prediction accuracy rate is 90.261% and the highest accuracy rate is up to 94.352. It shows that the performance of the model is good. At the same time, from the test process, we can know that differential privacy adds noise data to the model parameters, the added noise will directly affect the performance of the model. When the noise is small, the performance loss of the model will be small, but the security will be poor; on the contrary, when the noise is large, the performance loss will be higher, but the security becomes stronger. In general, this is a good data sharing scheme that balanced in data sharing and privacy protection, it can be applied to many practical scenarios with high prediction accuracy. But in the face of the complex network environment, communication exceptions are also a problem we have to consider. In addition, in this scheme, each client uses the same local model and the same amount of training data, in practical applications, how to design different local models and how to balance the data volume on the client side are the problems to be solved in the future.

Funding Statement: This work was supported by Funding of the Nanjing Institute of Technology (No. KE21-451).

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

References

[1] K. K. Kumar, M. A. Rao and G. L. Narayana, “Influence of big data,” *International Journal of Advanced Engineering and Management*, vol. 2, no. 4, pp. 56–59, 2020.

[2] Y. Chen, C. Shen, Q. Wang, Q. Li, C. Wang *et al.,* “Security and privacy risks in artificial intelligence systems,” *Journal of Computer Research and Development*, vol. 56, no. 10, pp. 2135–2150, 2019.

[3] M. A. Mukhtar, M. K. Bhatti and G. Gogniat, “Architectures for security: A comparative analysis of hardware security features in Intel SGX and ARM TrustZone,” in *2019 2nd Int. Conf. on Communication, Computing and Digital Systems*, Islamabad, Pakistan, IEEE, pp. 299–304, 2019.

[4] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C. Gao *et al.,* “Secure multi-party computation: Theory, practice and applications,” *Information Sciences*, vol. 476, pp. 357–372, 2019.

[5] M. Waseem, A. Lakhan and I. A. Jamali, “Data security of mobile cloud computing on cloud server,” *Open Access Library Journal*, vol. 3, no. 4, pp. 1–11, 2016.

[6] C. Gonçalves, R. J. Bessa and P. Pinson, “Privacy-preserving distributed learning for renewable energy forecasting,” *IEEE Transactions on Sustainable Energy*, vol. 12, no. 3, pp. 1777–1787, 2021.

[7] N. Peng and H. Wang, *Federated Learning Technology and Practice*. Beijing, China: Electronic Industry Press, pp. 186–194, 2021.

[8] J. Wang, Z. Li and A. He, *Dive into Federated Learning Principle and Practice*. Beijing, China: Machinery Industry Press, pp. 18–49, 2021.

[9] L. Yang, Y. Zou, M. Xu, Y. Xu, D. Yu *et al.,* “Distributed consensus for blockchains in internet-of-things networks,” *Tsinghua Science and Technology*, vol. 27, no. 5, pp. 817–831, 2022.

[10] H. S. Rhee, “Chosen-ciphertext attack secure public-key encryption with keyword search,” *Computers, Materials & Continua*, vol. 73, no. 1, pp. 69–85, 2022.

[11] Q. Yang, A. Huang, Y. Liu and T. Chen, *Practicing Federated Learning*. Beijing, China: Electronic Industry Press, pp. 26–30, 2021.

[12] A. Hard, K. Rao, R. Mathews, F. Beaufays and D. Ramage, “Federated learning for mobile keyboard prediction,” arXiv preprint arXiv:1811.03604, 2018.

[13] D. Shi, L. Li, R. Chen, P. Prakash, M. Pan *et al.,* “Towards energy efficient federated learning over 5G+ mobile devices,” arXiv preprint arXiv:2101.04866, 2021.

[14] D. Jiang, Y. Tong, Y. Song, X. Wu, W. Zhao *et al.,* “Industrial federated topic modeling,” *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 1, pp. 1–22, 2021.

[15] Y. Luo, H. Zhou, W. Tu, Y. Chen, W. Dai *et al.,* “Network on network for tabular data classification in real-world applications,” in *The 43rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, Virtual Event, China, pp. 2317–2326, 2020.

[16] A. AitMlouk, S. Alawadi, S. Toor and A. Hellander, “Fedqas: Privacy-aware machine reading comprehension with federated learning,” *Applied Sciences*, vol. 12, no. 6, pp. 1–12, 2022.

[17] Y. Aono, T. Hayashi, L. Wang and S. Moriai, “Privacy-preserving deep learning via additively homomorphic encryption,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[18] C. Dwork, “Differential privacy,” in *International Colloquium on Automata, Languages and Programming*, Berlin, Heidelberg: Springer, pp. 1–12, 2006.

[19] J. Shen, *Linear Regression*. Boston, MA: Springer Press, pp. 1622, 2009.

[20] Z. Zhou, *Machine Learning*. Beijing, China: Tsinghua University Press, pp. 53–60, 2016.

[21] C. Jia, H. Wang and D. Zhou, “Gradient learning in a classification setting by gradient descent,” *Journal of Approximation Theory*, vol. 161, no. 2, pp. 674–692, 2009.

[22] C. Luo, X. Chen, C. Ma and S. Zhang, “Improved federated average algorithm based on tomographic analysis,” *Computer Science*, vol. 48, no. 8, pp. 32–40, 2021.

[23] J. Wang, L. Kong and Z. Huang, “Research advances on privacy protection of federated learning,” *Big Data Research*, vol. 7, no. 3, pp. 130–149, 2021.

[24] X. Li, K. Huang, W. Yang, S. Wang and Z. Zhang, “On the convergence of FedAvg on non-IID data,” arXiv preprint arXiv, pp. 4–8, 2019.

[25] A. Alsirhani, M. Ezz and A. M. Mostafa, “Advanced authentication mechanisms for identity and access management in cloud computing,” *Computer Systems Science and Engineering*, vol. 43, no. 3, pp. 967–984, 2022.

[26] M. Ragab, H. A. Abdushkour, A. F. Nahhas and W. H. Aljedaibi, “Deer hunting optimization with deep learning model for lung cancer classification,” *Computers, Materials & Continua*, vol. 73, no. 1, pp. 533–546, 2022.

[27] X. Zhang, W. Zhang, W. Sun, X. Sun and S. Jha, “A robust 3-D medical watermarking based on wavelet transform for data protection,” *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.

[28] Y. Y. Ghadi, I. Akhter, S. A. Alsuhibany, T. A. Shloul, A. Jalal *et al.,* “Multiple events detection using context-intelligence features,” *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp. 1455–1471, 2022.

[29] W. H. Wolberg, “*Breast Cancer Wisconsin Data Set*,” Madison, Wisconsin, USA, 2021. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29.

[30] J. Li, C. Zhao, Q. Yang, W. Qiu, Q. Chen *et al.,* “Federated learning-based short-term building energy consumption prediction method for solving the data silos problem,” *Building Simulation*, vol. 15, no. 6, pp. 1145–1159, 2022.