A Secure Sharing Method for University Personnel Archive Data Based on Federated Learning

Xinwei Li[1(✉)], Yue Zhao[1], and Min Zhou[2]

[1] Human Resources Department, Changchun University of Architecture and Civil Engineering, Changchun 130000, China
lixinwei58666@163.com

[2] Department of Employmeng and Enterpreneurship, Xiamen Ocean Vocational College, Xiamen 361000, China

Abstract. In response to the complex data trust evaluation process in the current process of secure sharing of university personnel archive data, which leads to long data encryption time and poor data sharing and distribution performance, a federated learning based method for secure sharing of university personnel archive data is proposed. Build a data federation learning module to provide a platform for subsequent data processing. Optimize federated learning algorithms and complete incremental federated learning of archive data. Federated incremental learning of archival data. Improve data privacy and security. Apply Kalman filtering technology and data mapping technology to achieve secure sharing of archival data. The experimental results show that this method can effectively reduce data encryption time and provide data sharing and distribution performance.

Keywords: Federated learning · Archive data processing · Data distribution · Data sharing

1 Introduction

With the rapid development of modern computer science and significant improvement of computing power, machine learning is widely used in various fields of production and life, such as intelligent medical treatment, automatic driving and face recognition. These applications of machine learning are changing the way people live, learn and produce. The development of the Internet of Things makes the data of the Internet of things more and more huge, huge data can mine a large number of valuable information. GSMA predicts that by 2025, the number of connected Internet of Things devices in the world will reach 25 billion. The massive data collected by Internet of Things devices can generate meaningful information for human production and life through the mining of artificial intelligence and other technologies [1, 2]. For example, in smart medicine, doctors diagnose and analyze human health information collected by wearable devices to make scientific diagnosis and treatment suggestions for patients. In the scenario of

L. Yun et al. (Eds.): ADHIP 2023, LNICST 547, pp. 184–197, 2024.
https://doi.org/10.1007/978-3-031-50543-0_13

university Internet of Things, personnel file data collected by Internet of Things devices can help universities optimize education effect, improve teaching efficiency and reduce the cost of manual file management through intelligent analysis.

Under the influence of big data technology, the data size and model complexity of university personnel archive data are gradually increasing. In large-scale machine learning applications, the amount of data can be so large that a single computer's computing power cannot afford it. At this point, distributed machine learning has attracted the public's attention. Unlike model training on a single computer, distributed machine learning divides data and computing tasks that were originally concentrated on the server into different nodes, and multiple nodes collaborate to assist the server in training the model, thus supporting large-scale data training [3, 4]. Today's Internet of Things and edge computing, based on distributed training, have accelerated the pace of AI industrialization on the premise of ensuring data security.The effective union of production data between data nodes can realize the joint optimization of archival data, and further develop the technology of personnel data processing and sharing in colleges and universities. Due to the limited computing and storage capacity of Internet of Things devices, it is often necessary to outsource computing and storage requirements to cloud service providers. Therefore, before blockchain technology was widely studied by the academic community, Internet of Things data sharing was mainly realized through cloud services, supplemented by cryptography algorithm to realize data access control and ensure data security. However, the current proposed security sharing method of personnel files data in colleges and universities is complicated to set the link of data credit score, which leads to a long time in data security processing and low accuracy in data distribution. Therefore, a security sharing method of personnel files data in colleges and universities based on federal learning is proposed to optimize the shortcomings of the current security sharing method in the application process.

2 Build a Data Federation Learning Module

Literature research shows that federated learning can ensure the sharing of gradient information among all participants. Currently, in the process of data security sharing, the method of calculating the weighted average of all parties is commonly used. Therefore, the upload of client encryption gradients, aggregation of server encryption gradients, and distribution of global shared models are involved, and the overall operation time is long, and the operation security factor is poor [5, 6]. Therefore, in this study, federated learning algorithms were used to optimize the current method and compensate for its shortcomings. In the specific design, the following considerations were made in this study:

① Improved timeliness of data encryption:

Use Differential privacy technology: Differential privacy is a method to protect individual privacy, which can provide high data encryption timeliness while protecting data privacy. The Differential privacy technology can be used to encrypt and Data anonymization sensitive data.

Optimize encryption algorithms and key management: Choose efficient encryption algorithms and key management strategies to improve the efficiency of encryption and decryption, and reduce the impact of the encryption process on system performance.

② Improve data distribution accuracy:

Introduction of model aggregation and parameter update mechanism: in Federated learning, all participants jointly train models and share updated parameters, which can improve the accuracy of data distribution through model aggregation and parameter update mechanism. Participants share some model parameters rather than complete original data, thus protecting data privacy.

Data standardization and cleaning: Before data distribution, standardize and clean the data to ensure consistency and accuracy. This will help improve the accuracy and availability of data distribution.

③ Building Federated learning module:

Build a secure federated learning platform: Establish a secure and trustworthy federated learning platform to ensure security and privacy protection during data transmission and model training processes.

Design a reasonable participant collaboration mechanism: clarify the roles and responsibilities of each participant, develop a reasonable participant collaboration mechanism, and ensure the smooth progress of data sharing and model training.

When completing personnel file data sharing in colleges and universities, A data set for data sharing processing is usually given, in which each sample is represented as a number pair (E, F) composed of data feature E and data label F, and an initial model parameter u and a given form of loss function $\alpha(E, F, u)$ are given. Then the loss function can be used to calculate the loss of each data sample (E, F) on model u, so as to obtain the gap between the model performance and the training target based on the current data set and the current model parameter training. Finally, minimizing the gap between the reality and the target is taken as an important criterion throughout the process of data sharing to guide the updating direction of the model and realize the training of the model. Take the r sample in the training data set as an example, the feature of this data point is E_r and the label is F_r. If model parameter u is given, the loss function $f(u)$ on the whole training data set is defined as the form of sum of finite terms:

$$f(u) = \frac{\sum_{i=1}^{n} f_r(u)}{n} = \frac{\sum_{i=1}^{n} f_r(E_r, F, u)}{n} \tag{1}$$

where, n represents the number of training data, u represents the parameters of the neural network model, and f_r represents the loss function of a single sample. Use this function to control the loss during data processing. For the other half of the privacy budget δ, in each iteration, δ is divided into two parts, the gradient budget δ_i and the step budget δ_j. δ_i is used to calculate gradients with noise, and δ_j is used to find the optimal gradient descent step size. At the same time, add an initialization weight vector β. At this point, in the second row of federated learning calculation, set t to record the number of iteration rounds initialized to 0, and randomly set δ to δ_i and δ_j. The third line, when the privacy budget is not exhausted, performs a loop, which is model training. On line 4, record the index value of the gradient step with δ_j and initialize it to O. There are:

$$h_t(s) \leftarrow \nabla\alpha(f(u), n) \tag{2}$$

$$h_t(s) \leftarrow \frac{h_t(s)}{\max(1, \frac{\|h_t(s)\|_2}{Y})} \tag{3}$$

$$\widetilde{h_t(s)} \leftarrow \frac{\sum (h_t(s) + N(\frac{0, Y}{\delta_j}))}{N} \tag{4}$$

Y indicates the data processing threshold. The above function can be applied to the data credit score, according to the score results, determine the security level of the data. The above Settings are sorted out and constructed as a federal learning framework, so that archive data users can sort and update data in this framework and realize data cloud interaction.

3 Design of Security Sharing Method of Personnel File Data in Universities

In federated learning, a model is jointly trained among multiple users in a decentralized manner, and the model learning process is transformed to the user side. The user only uploads the model parameters to the cloud platform, which then integrates the parameters and sends them back to each user. Due to the potential involvement of sensitive information in training data, publishing the model during or after training poses a risk for federated learning frameworks to leak user privacy [7, 8]. The personnel files of universities contain a large amount of personal privacy data, and any abnormal encryption will cause irreparable losses. Therefore, based on previous research results and the data

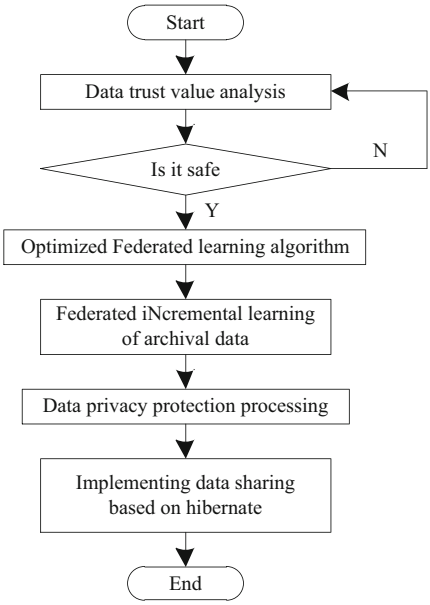


Fig. 1. Security Sharing Process of Personnel Archive Data in Universities

federation learning process mentioned earlier, the secure sharing process of archive data is set in the following form.

Next, follow the process shown in Fig. 1 for data processing and distribution. The specific data sharing operation process is set as follows.

3.1 Federated Incremental Learning of File Data

The object of this study is the highly complex personnel file data of colleges and universities. How to mine the newly generated state data for the exponential growth of new data and combine it into the existing mining mode has become a new focus. The model trained by the traditional data processing algorithm repeatedly extracts data features from the local data for training and learning, but it cannot adapt and incrementally modify the industry association model with the newly added data T in real time, resulting in an increase in time cost and a decrease in model diagnostic accuracy. This study uses incremental weighting to solve the federated incremental learning problem. The number of samples reflects the diversity of samples to some extent, and the model based on high-complexity data training has better scalability [9]. The process of model training can be understood as the "learning" process of the model. Generally, as time goes by, the model gets closer to the optimal solution of the problem, but more new data will increase the distance between the computing cloud and the optimal solution of the learning problem. Therefore, it is obviously unreasonable for the computing cloud with uneven new data to update the model parameters in the file association end in the same amount. Therefore, it is targeted to optimize.

The incremental weight value represents the proportion of the number of newly added samples in the cloud to the total number of original samples. The incremental weight of cloud c can be calculated from the number of newly added samples and the total number of samples:

$$\varepsilon_c = \frac{|U_c|}{|V_c|} \tag{5}$$

where, $|U_c|$ is the number of newly added samples in the computing cloud, and $|V_c|$ is the total number of original samples in the computing cloud. In the process of parameter optimization, there is a certain depth value. Let the parameter depth value be:

$$\varpi_{i+1}^c = \varpi_i^c - (\varepsilon_c * \varpi_i^c) \tag{6}$$

During the time interval between downloading and uploading parameters from the cloud, new training data will be generated, and the parameter depth value will also be updated to a certain extent. The parameter depth value represents the impact of newly added data from the local dataset on model performance during the completion of an iterative learning process in the computing cloud, reflecting the update level of the computing cloud. In order to reduce the parameter weighting of cloud computing with larger parameter depth values and relatively smooth attenuation process, this study chose the arctangent function as the incremental weighted attenuation function:

$$\sigma_{i+1}^c = \frac{2 \arctan \varpi_{i+1}^c}{\pi} \tag{7}$$

Under the above Settings, only the computing cloud in the participating subset is updated in each round. The contribution of the model to the aggregation operation can be determined according to the parameter depth value of the computing cloud model, which can effectively utilize the historical information and distinguish the utilization value of the local model, which is expected to improve the effectiveness of the aggregation operation. Therefore, the parameter weighting of the local model is further paid attention to, and the improved aggregation strategy is proposed as follows:

$$\lambda_{i+1} \leftarrow \sum_{c=1}^{c} \frac{n_c}{n} * \sigma_{i+1}^{c} * \varpi_{i+1}^{c} \tag{8}$$

In the process of federated incremental learning, the model parameters submitted by the computing cloud need to be incrementally weighted to participate in industry joint model optimization. The modified parameters are updated on the data joint processing end based on specific optimization algorithms to update the model parameters. After optimization, the latest industry joint model parameters are obtained from the cloud and covered by local parameters for the next round of iterative learning.

3.2 Data Privacy Protection Processing

The general process of federated learning was set in the previous section. However, the process of encryption and decryption of gradient or local model weights was ignored in the description. If the gradient or current local model weight is not encrypted, the above process can train a model normally. However, it will lead to the disclosure of user privacy data. Although in the general process of federated learning, data does not flow directly out of the local, only out of the gradient or when the local model weight, however, some research results have shown that such information contains user privacy data. Specifically, user data can be backderived through gradient, and the leakage gradient is almost a direct disclosure of user privacy data [10–12]. Therefore, the gradient or local model weights uploaded by users per round need to be protected by relevant techniques. Common protection schemes include security aggregation, homomorphic encryption and differential privacy. In this study, security aggregation is selected to build the protocol of federal learning privacy protection, and the specific main contents are set as follows.

Any user of this agreement needs to communicate with all remaining users, with the purpose of enabling Diffie Hellman key negotiation between any two users to obtain a secret that only both parties know. Any user, in order to ensure the security and robustness of the protocol, will share their secrets twice, that is, split their secrets into "secret fragments" and distribute them to all remaining users[13–15]. Only when they have a certain number of "secret fragments" can the shared secrets be reconstructed. Before sending data to the server, users will apply two masks locally, and the two masks will be removed during the aggregation phase in two queries from the server.

To ensure the security of the communication content on the channel, symmetric encryption can be used to encrypt the communication content. Symmetric encryption means that the same key is used to encrypt and decrypt information. To ensure information security, the key cannot be disclosed and only the communication parties know it. In the public key cryptosystem, the public key is used for encryption and the private key for

decryption. A public key is different from a private key. Public keys can be made public, and anyone can use them to encrypt; however, public keys cannot be decrypted. The private key is not public, and only the person with the private key can decrypt it. Let's say we're using a symmetric cryptosystem. Before the communication starts, both parties do not know the key and cannot perform symmetric encryption. Then, there needs to be a way for both parties to know this key securely. Therefore, in this study, a temporary secure private channel was established to transfer the key used by both parties for the convenience of subsequent processing.

3.3 Sharing File Data Securely

Based on the above settings, a new method for secure sharing of archival data is proposed. Due to the fact that archive data is mostly heterogeneous, Hibernate will be used in this study to achieve data sharing. Through the research on Hibernate based MCDMS heterogeneous data sharing technology, starting from the heterogeneous data source of archives, this paper explores the data mapping of each archive and the inheritance relationship of each archive's data table after saving the heterogeneous data source data to the Oracle9i database. In addition, this study will also describe the underlying data filtering technology of a single discipline and the Lucen data indexing technology currently used in the system. By using open source components and Java Web programming knowledge, the data indexing function will be implemented, providing a data level shared data query solution for the sharing of personnel archives data in universities.

Set the data in the file to $O = \{\mu 1, \mu 2, \mu 3, ..., \mu n\}$; Inherit the dependency relationship as the parent class PriorDiscipline, all derived subclasses as Derived Discipline N, and $N \in$ as any natural number. Then, in the relational database, you can use additional fields to represent the recorded ID of a specific subclass in the PrioritDiscipline table, and save the data segments that all subclasses need to share. When doing SQL or HQL (Hibernate Query Language) CRUD, you can use this ID to read the specific subclass, obtain the attribute values of the subclass saved in the parent class PrioritDiscipline, and complete the inheritance mapping. According to this setting, an additional field filter can be used for the PriorDiscipline parent class in the archive database (for example, the filter can be equal to the numerical value 1 to identify the parent class, and if it is NULL, it is not the parent class). The PriorDiscipline table does not store the data segments that all subclasses need to share. Instead, by passing the ID field of the PriorDiscipline parent class to the Derived DisciplineN, the ID is both the primary key of the Derived DisciplineN and the foreign key of the Derived DisciplineN. When using SQL or HQL for CRUD, the first step is to determine whether the filter is NULL. If it is not, the ID of PriorDiscipline is taken to compare all subclass IDs. If a subclass with a matching ID value is found, the attribute values of the subclass are obtained to complete the inheritance mapping.

Some noise is generated in the mapping processing process. In order to avoid the impact on the archive data and increase the process, Kalman filtering technology is used in this study to complete this part of operation. Kalman filter can deal with normally distributed noise in linear system. The noise added to the model parameters after differential privacy processing is Gaussian noise, which meets the condition of normal distribution, and the state update equation is linear. Therefore, Kalman filter can be used to de-noise

the model parameters to reduce the impact of differential privacy on model accuracy. The data update parametric equation can be expressed as:

$$\begin{cases} g_{t+1} = g_t - \rho \sum h_i \\ g_{t+1,1} = g_{t+1} - \rho g_{t+1} + \wp(0, \zeta^2 D) \\ \cdots \\ g_{t+1,N} = g_{t+N} - \rho g_{t+N} + \wp(0, \zeta^2 D) \end{cases} \tag{9}$$

Among them, g_t represents the global model parameters for the t round of global training, and g_{t+N} represents the model parameters for the target data sent to different data nodes in the t round. The noise added during the differential privacy process uses the Gaussian mechanism, which satisfies the Gaussian distribution $\wp(0, \zeta^2 D)$. Transform the parameter update equation of model training into matrix form. In the t round of federated learning, there are N sending nodes with model parameters $g_{t+1,1}$ and global model parameters $g_{t+1,N}$. The parameter vector composed of all node model parameters is $G = \left[g_{t,1}\ g_{t,2}\ \ldots\ g_{t,n} \right]$, and its gradient value can be expressed as $g_{t,n}$. At this point, the gradient vector composed of the gradient values of all data distribution nodes is $g_{t,n}$, and the data filtering linear coefficient matrix can be written as:

$$L = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & \frac{G'}{N} & 0 & \ldots & 0 \\ 0 & 0 & \frac{G'}{N} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \frac{G'}{N} \end{bmatrix} \tag{10}$$

Through this formula, the noise in the data is filtered and the content set above is followed to realize the safe sharing of archive data. So far, based on the federal learning university personnel file data security sharing method design is completed.

4 Experimental Analysis

This study proposes a secure sharing method for university personnel archive data based on federated learning, and improves the data user credit scoring section of the method. In this stage, the method will be tested and analyzed, including data encryption testing, simulation application performance testing, and credit scoring performance testing.

4.1 Experimental Data

This experiment uses two commonly used open source datasets, MNIST and FMNIST, as experimental data. Below are brief introductions.

The MNIST dataset records handwritten archive images from 100 different individuals, with numbers ranging from 0 to 9 and 10 categories. The statistical purpose is to train recognition models for handwritten digits. The MNIST dataset contains a total of 10000 grayscale images with channel 1, of which 5000 were divided into training

datasets and the remaining 5000 were used for this experiment. The data size of each handwritten image is 56 pixels * 56 pixels.

The EMNIST dataset is an extension of the MNIST dataset. Based on handwritten numerical pictures of different people, the relevant attribute data is recorded. The handwritten digits are 0 to 9, the attribute data characters are set as lowercase letters a to z and uppercase letters A to Z, and the number of categories is 62. The FMNIST dataset contains 10000 pieces of data. There are a large number of categories in the FMNIST data set, and the upper and lower case forms of some handwritten letters are similar, so the FMNIST data set can be divided into five categories, which are detailed as shown in Table 1.

Table 1. Classification results of FMNIST dataset

Category name	Training set data quantity/item	Test set data quantity/item	Total number of data sets per/item
Z1	458	1042	1500
Z2	2000	1000	3000
Z3	4500	1500	6000
Z4	1000	1000	2000
Z5	1350	1150	2500

Organize the above content and use it as the data source for this experiment to analyze the application effect of the methods in the text.

4.2 Experimental Environment

All experiments in this chapter were performed on four high-performance CPU servers with the same configuration. Table 2 shows the specific configurations of each server.

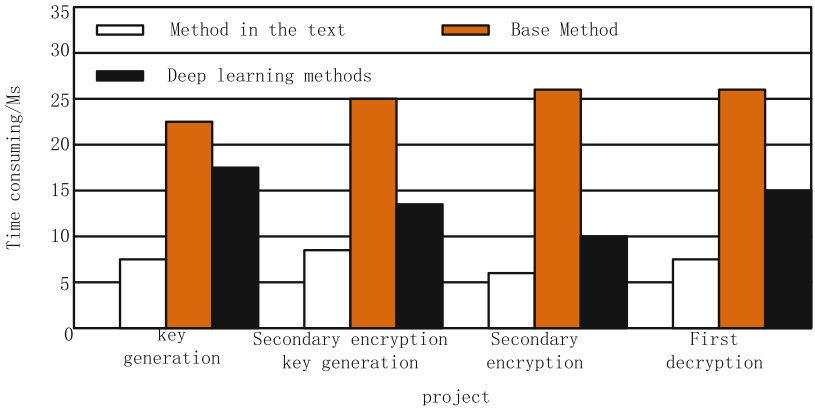
Table 2. Configuration information of the experimental server

Information sequence number	Message name	Specific content
1	CPU	8 nuclei, Intel Xeon Processor (Skylake, IBRS)@2194.848 MHz
2	Memory	32 GB
3	Disk	1TB
Tb	Network card	Red Hat, Inc. Virtio network device
5	Operating system	Centos 7.2
6	Kernel version	3.10.0?1160.15.2.e17.x86–64

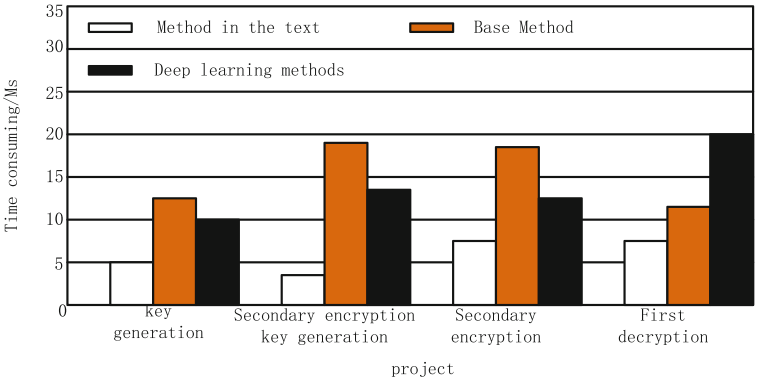
In terms of software, all implemented applications in this experiment were deployed in the form of containers using Docker software; The development language includes Java and Golang; The IBPRE algorithm is implemented based on the official JPBC library of Java (Java Pairing BasedCryptography); The blockchain platform uses Tendermint and Hyperledger Fabric. Fabric is used for comparison with Tendermint; The functional testing software used Postman performance testing software and Apache's open-source JMeter project.

4.3 Encryption Timeliness Test

On the premise of fixed experimental data, compare and analyze the data encryption effects of the basic method, deep learning method, and the method in the text. The encryption effect is reflected by the encryption time, as shown in Fig. 2.



(a) Data set 1



(b) Data set 2

Fig. 2. Encryption experiment results

Analyzing Fig. 2, it can be seen that there are significant differences in encryption time among the three methods for different databases. For database 1, the encryption process of this method takes significantly less time, and it can complete the encryption of university personnel file data within 10ms. Compared with the methods in this article, the encryption efficiency of the two comparison methods is relatively low. Compared to Database 1, the three methods generally take less time to encrypt Database 2, but the encryption time of this method is still less than that of the two comparative methods. Therefore, the encryption timeliness of this method is higher.

4.4 Simulation Application Performance Test

This experiment sets the performance testing index of the secure sharing method as the accuracy of data distribution, and the specific calculation formula is set as follows:

$$A = \frac{a+d}{a+b+c+d} \tag{11}$$

where, a represents the correct number of samples distributed; b represents the correct number of negative samples distributed; c represents the number of samples wrongly distributed; d represents the negative number of samples distributed incorrectly.

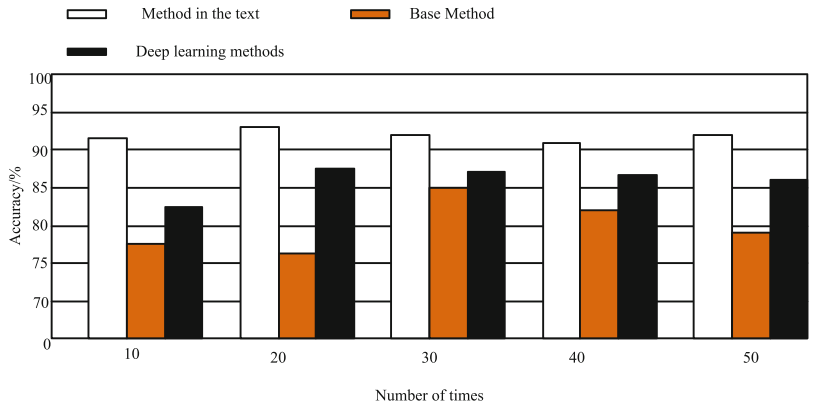
According to formula (11), complete 50 data sharing and distribution tests. The accuracy of data distribution using different methods is shown in Fig. 3.

Analyzing Fig. 3, it can be seen that the data distribution capability of the method proposed in this paper is far superior to the other two methods. During 50 tests, the method proposed in this article can maximize the accuracy of data distribution, with a distribution accuracy rate consistently above 90%, significantly higher than the two comparative methods. This indicates that the method proposed in this article can effectively avoid the problem of abnormal data sharing and ensure the reliability of archival data applications.

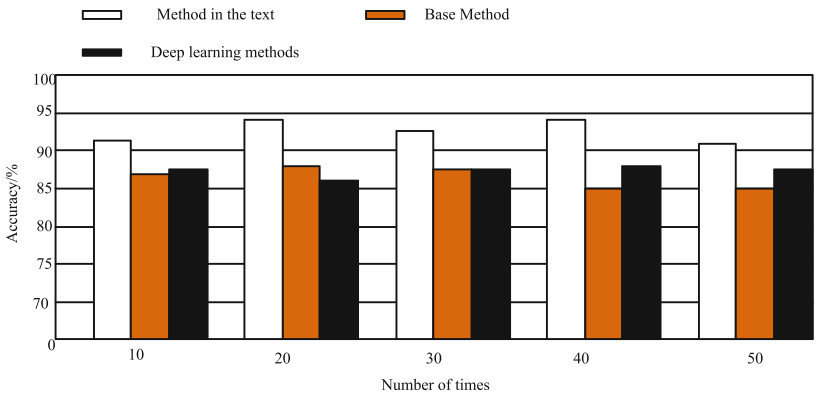
4.5 Credit Score Performance Test

In this study, the credit score part of the security sharing method is optimized. This experiment will compare the data transaction processing performance after the sharing method is applied in the case of malicious nodes and credit scoring algorithm, so as to verify the effectiveness of the method proposed in this study (Fig. 4).

Comparing the content in the above figure, it can be seen that when the data volume is constant, the data processing capacity of each node after the application of the method in the article is relatively large, indicating that this method can effectively control the credit value of the data and avoid data distribution and sharing problems caused by abnormal credit scores. The credit scoring process of the other two methods has poor application results when there are malicious nodes, and the credit scoring process affects the efficiency of data processing and transactions. Based on the above experimental results, it can be determined that the credit scoring stage setting of the method in the article is relatively reasonable.



(a) Data set 1



(b) Data set 2

Fig. 3. Simulation application performance test results

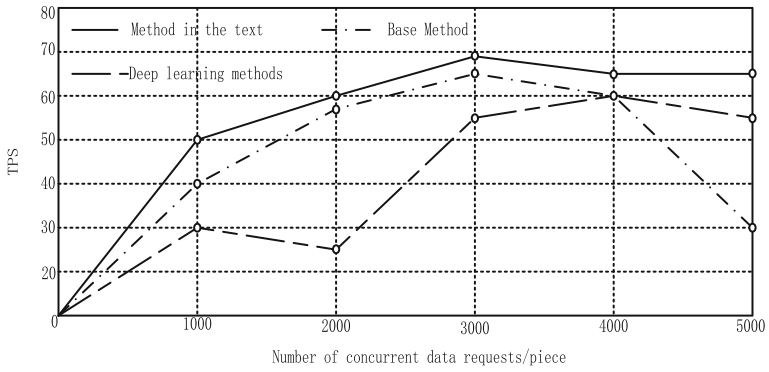


Fig. 4. Credit score performance test results

5 Conclusion

In this study, from the part of data encryption and data security management, the current sharing method of personnel files in colleges and universities was optimized, and the federal learning module was added to conduct credit assessment on the original data, so as to improve the security of data and the ability to resist malignant interference.

The method proposed in this article has achieved good application results at the current stage. This method can not only effectively shorten the encryption time, but also improve the distribution accuracy and credit rating of the data sharing process. However, during the testing process, it was found that there is still room for further improvement in the scalability of this method. Therefore, in future research, it is necessary to optimize the application process of this method, including increasing the number of supported users, data scale, and flexibility of the architecture, in order to further ensure the application effect of this method.

References

1. Guo, L., Zhu, Q., Zheng, J., et al.: The research on data sharing and security protection for students' comprehensive literacy evaluation based on federated learning. China Educ. Technol. **10**, 56–63 (2022)
2. Lu, C., Deng, S., Ma, W., et al.: Clustered federated learning methods based on DBSCAN clustering. Comput. Sci. **49**(z1), 232–237 (2022)
3. Liu, W., Xu, X., Zhang, X., et al.: Federated learning based method for intelligent computing with privacy preserving in edge computing. Comput. Integr. Manuf. Syst.. Integr. Manuf. Syst. **27**(9), 2604–2610 (2021)
4. Wen, Y., Chen, M.: Medical data sharing scheme combined with federal learning and blockchain. Comput. Eng. **48**(5), 145–153,161 (2022)
5. Mo, H., Zheng, H., Gao, M., et al.: Multi-source heterogeneous data fusion based on federated learning. J. Comput. Res. Dev. **59**(2), 478–487 (2022)
6. Zhao, Y., Wang, L., Chen, J., et al.: Network anomaly detection based on federated learning. J. Beijing Univ. Chem. Technol. (Natural Science Edition) **48**(2), 92–99 (2021)
7. Zheng, J., Li, W., Liu, X., et al.: Research on the federal sharing technology of remote sensing image artificial intelligence datasets. Spacecraft Recovery Remote Sensing **43**(4), 12–24 (2022)
8. Su, Y., Zhang, H., Liu, J.: Federated recommendation algorithm based on equilibrium learning. J. Hefei Univ. Technol. (Natural Science) **45**(5), 625–632 (2022)
9. Liu, X., Yin, Y., Chen, W., et al.: Secure data sharing scheme in Internet of Vehicles based on blockchain. J. Zhejiang Univ. (Eng. Sci.) **55**(5), 957–965 (2021)
10. Yang, Y., Lin, D., Huang, F., et al.: Blockchain based secure data sharing system for internet of things. J. Fuzhou Univ. (Natural Sci. Edition) **49**(6), 739–746 (2021)
11. Jia, Z., Zhang, J., Yi, K.: Mobile education resource sharing method for wireless broadband connection. Secur. Commun. Networks **16**(5), 127–135 (2021)
12. Noreen, H.S., Thar, B., Ghulam, A., Haq, A.Z.: MACRS: an enhanced directory-based resource sharing framework for mobile ad hoc networks. Electronics **11**(5), 725–732 (2022)
13. da Costa, L.A.L.F., Kunst, R., de Freitas, E.P.: Intelligent resource sharing to enable quality of service for network clients: the trade-off between accuracy and complexity. Computing **14**(7), 1–13 (2022)

14. Erqi, Z.: Application of conditional random field model based on machine learning in online and offline integrated educational resource recommendation. Math. Probl. Eng.Probl. Eng. **20**(1), 551–5526 (2022)
15. Qian, X., Li, F.: Evolutionary game analysis of information sharing in closed loop supply chain based on cloud platform. J. Univ. Shanghai Sci. Technol. **43**(06), 606–616 (2021)