Digital Communications and Networks 8 (2022) 636–643

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

An intelligent and privacy-enhanced data sharing strategy for blockchain-empowered Internet of Things

Qinyang Miao [a,b], Hui Lin [a,b,*], Jia Hu [c], Xiaoding Wang [a,b,**]

[a] *College of Computer and Cyber Security, Fujian Normal University, Fuzhou, 350117, China*
[b] *Engineering Research Center of Cyber Security and Education Informatization, Fujian Province University, Fuzhou, 350117, China*
[c] *University of Exeter, EX4 4RN Exeter, UK*

ARTICLE INFO

Keywords:
Data sharing
Federated learning
Blockchain
Privacy protection
IoT

ABSTRACT

With the development of the Internet of Things (IoT), the massive data sharing between IoT devices improves the Quality of Service (QoS) and user experience in various IoT applications. However, data sharing may cause serious privacy leakages to data providers. To address this problem, in this study, data sharing is realized through model sharing, based on which a secure data sharing mechanism, called BP2P-FL, is proposed using peer-to-peer federated learning with the privacy protection of data providers. In addition, by introducing the blockchain to the data sharing, every training process is recorded to ensure that data providers offer high-quality data. For further privacy protection, the differential privacy technology is used to disturb the global data sharing model. The experimental results show that BP2P-FL has high accuracy and feasibility in the data sharing of various IoT applications.

1. Introduction

With the development of the Internet technology, the Internet of Things (IoT) is widely used in various industries [1]. Sensors are an important part of the IoT and the most important data source for the IoT system. The perception data collected by a single sensor often cannot meet users' needs, and the real value of the IoT lies in the comprehensive utilization and sharing of various data and information [2–4]. For example, in healthcare, data sharing can provide valuable health records, including treatment and physical examination information, and can offer more targeted treatments for patients. In industry, by analyzing the collected data, data sharing can accurately understand the preferences of tourists and predict future tourism hot spots to improve the quality of service. However, data sharing in IoT may face various problems. First, it is very difficult for each pair of organizations to build mutual trust. As a result, it is unlikely to share reliable local data. Second, data privacy has become a big problem that hinders data sharing because data owners suffer from privacy leakage. Therefore, achieving effective data sharing is a challenge, especially when these two problems have not been solved.

Machine learning [5] technologies are widely used in data sharing. Traditional machine learning technologies collect data first and then focus on model training. However, data collection is often difficult because data owners are worried about privacy leakage. Federated learning is a distributed machine learning framework, which not only reduces the computing burden of centralized devices by aggregating the local training model of data owners rather than the original data, but also protects the data privacy of data owners [6]. As a distributed shared ledger and database, the blockchain [7,8] has the characteristics of decentralization, non-tampering, tracing, collective maintenance, openness, and transparency, which can provide reliable technical supports for the privacy protection of data sharing. For example, the blockchain can record the sharing behavior of each participant who provides a data model, thus forcing the participants to provide a reliable data model.

According to abovementioned analysis, we propose herein a secure data sharing mechanism, called BP2P-FL, using peer-to-peer federated learning with the privacy protection of data providers. The contributions of this paper are summarized as follows:

- A data sharing mechanism based on federated learning is proposed. This mechanism transforms the data sharing problem into a model sharing problem and realizes team-based data sharing. In addition, the reward and punishment mechanism is introduced. Specifically, the data requester will reward and punish each team according to the results of data sharing, such that team members can complete the data

* Corresponding author.
** Corresponding author.
E-mail addresses: 18291972715@163.com (Q. Miao), linhui@fjnu.edu.cn (H. Lin), j.hu@exeter.ac.uk (J. Hu), wangdin1982@fjnu.edu.cn (X. Wang).

https://doi.org/10.1016/j.dcan.2021.12.007
Received 15 April 2021; Received in revised form 18 December 2021; Accepted 28 December 2021
Available online 3 January 2022

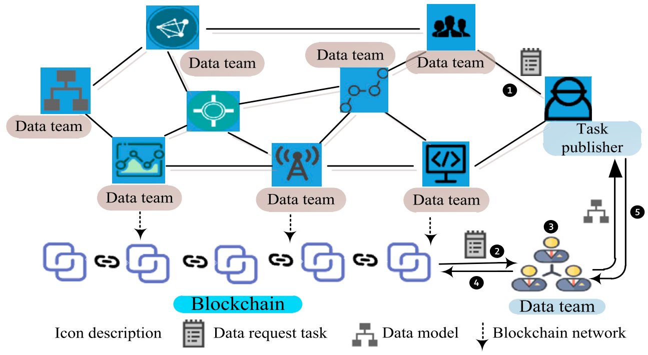


Fig. 1. System architecture.

sharing with high quality and reliability. Moreover, the "mortgage-penalty" mechanism is introduced to further punish members who provide unreliable data. Each team can further manage and supervise members, such that they can efficiently and reliably complete the data sharing tasks.

- Differential privacy is applied to data sharing by adding a Laplacian to the global data sharing model, preventing the inference attack initiated by data requesters and providing further privacy protection to the data.
- The experiment results show that BP2P-FL achieves high accuracy and feasibility for privacy-enhanced data sharing in IoT.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 introduces the system model. Section 4 elaborates the proposed strategy BP2P-FL. Section 5 presents the experiment. Section 6 concludes this paper.

2. Related work

Secure data sharing in IoT has drawn an ever-growing interest, and many data sharing mechanisms have been proposed.

In [9], the author proposed a secure data sharing framework using the blockchain and the proxy re-encryption technology. To perform fine-grained control on visitors and prevent privacy leakage [10], proposed a new encryption algorithm based on hierarchical attributes, which assigns attributes to authorization centers based on blockchain to realize data security sharing. In Ref. [11], the authors embedded access control rules into smart contracts to control user access to data and divided the blockchain into multiple channels to protect data privacy and security. To solve the problem of insecure data sharing caused by an untrusted environment [12], proposed an efficient and secure data sharing model based on the blockchain, which was based on attribute encryption and can resist multiple attacks.

Although the encryption and decryption technologies can effectively protect the privacy and security of data sharing, they are inefficient in large-scale computing environments, and federated learning [13] methods bring new opportunities for data security sharing. In Ref. [14], the authors proposed an efficient federated learning scheme to ensure data privacy. This scheme can resist collusion attacks in a distributed environment and, at the same time, prevent personal data privacy leakage. To solve the communication overhead of model training, the authors in Ref. [15] proposed a sparse compression framework suitable for broadband constrained environments. In Ref. [16], the authors improved federated learning by evaluating the participant's model feedback and the update method of participant weights. In Ref. [17], the authors combined data sharing, machine learning, blockchain, and federated learning to solve the privacy protection problem in data sharing. Meanwhile, to further optimize federated learning, the authors in Ref. [18] used deep reinforcement learning to select the participating nodes of federated learning, thereby improving the efficiency of the data sharing process. In Ref. [19], the authors combined federated learning and cryptography to protect the data privacy of data sharing participants in the social IoT, and used a sparse differential gradient to improve data transmission and storage efficiency. To solve the security problem of resource sharing under the Internet of Vehicles, the authors in Ref. [20] constructed a safe and hierarchical federated learning scheme to protect the privacy of the local data model. Although the abovementioned work has made positive contributions to privacy protection, how to ensure the reliability of the data sharing process still needs further research. Therefore, this study proposes a data sharing mechanism based on federated learning to realize the safe and reliable sharing of data without trust.

3. System model

We considered the collaborative data sharing scenario in this study. That is, after the data requester sends a data sharing request, multiple data providers collaboratively train a data model to realize data sharing. Therefore, two entities should be considered in this scenario, namely task receivers and the task publisher. Specifically, a team of users receive data sharing tasks, and each user will participate in the training process of federated learning. In this paper, we call these users data nodes, and they will complete the blockchain consensus process. Each team has a team leader who is responsible for receiving data sharing tasks, supervising the federated learning process in data sharing, and sending the global model combined with differential privacy to the task publisher, to prevent the task publisher from inferring the privacy information of data providers. A data requester is also called the task publisher, that is, the party who needs data, and usually publishes its own data sharing request tasks on the blockchain.

The blockchain-based data sharing can trace each data sharing node, ensuring the traceability of data sharing. Therefore, we consider the alliance chain and federated learning modules. The blockchain module establishes a secure connection between all nodes, and all transactions are packaged into blocks by miners. Considering audibility, the alliance chain will record all data sharing records to track the nodes participating in the data request task and the data usage. The architecture of the scheme proposed in this article is shown in Fig. 1. Our proposed architecture includes the following processes: the data requester issues the request task; the team responds to the task; shared transaction records are generated; the data nodes reach a consensus; and the data requester issues credit rewards are generated. The data requester specifically sends the data request task to the nearby blockchain node. If it is a new data request, the blockchain node broadcasts it on the blockchain. The data nodes respond to tasks in the form of teams. All shared records between the data requester and the data node are packaged into shared transactions by the transaction record node. Finally, the task publisher will give corresponding credit rewards based on the completion of the work. The flowchart of this process is shown in Fig. 2.

Data providers and requesters are not trusted, which may lead them to act dishonestly. The proposed architecture is vulnerable to two threats. First, dishonest data providers may provide false or malicious models, resulting in unreliable training results. Unreliable data providers may temporarily withdraw, which will adversely affect the quality and efficiency of the global model. Second, the data requester may attempt to infer the data providers' privacy information from the data model, resulting in privacy leakage.

4. BP2P-FL implementation

4.1. Management of team members

The traditional blockchain allows individual nodes that do not trust each other to register; however, this method is not suitable for tasks that require collaboration. In this article, to ensure the high-quality completion of the data request task, we propose a new registration method in which the mutual trust node is registered as a team. When the data requester issues the requested task, the team that meets the credit rating requirements on the blockchain responds to the task. In other words, the credit rating of each node in the team meets the requirements.

It is an efficient and mutually beneficial way for multi-data nodes to complete the assigned tasks through collaboration. First, the team sponsor initiates the member recruitment information, and the team leader S_p sets the deadline for the recruitment response, work tasks, and member requirements. The member requirements are formulated according to the specific tasks to be solved, such as quantity and capacity requirements. The data node registers for the election according to its own abilities and sends the work preference to the team sponsor. The team leader evaluates the capabilities of the data node and comprehensively selects the team members that meet the requirements. The teaming algorithm is shown in Algorithm 1.

Algorithm 1 Team Construction.

```
The team leader releases team aggregate information;
Interested data nodes register;
Each data node evaluates its job preference $W_p$ and responds to the team leader;
The team leader evaluates the capabilities $A_b$ of each data node based on the registration materials;
The team leader calculates the integrated value $I_V = 0.5W_p + 0.5A_b$;
The first $N$ data nodes are comprehensively selected;
```

In reality, it is easy to form a team when data nodes have similar work preferences or complementary work abilities. However, forming a team based on these alone will be unreliable because data nodes may be selfish(e.g., temporary withdrawal or lazy behavior in collaborative tasks), which affects the global work quality and efficiency of the entire team. In response to this problem, we designed an internal team management mechanism based on "mortgage-punishment"; that is, a certain amount of mortgage must be provided when forming a team, and nodes with the abovementioned bad behaviors must be "punished" to make up for the losses of the other nodes. The mortgage is determined by the team to ensure the honesty of each team member. The mortgage will set the minimum value but not the maximum value. The greater the mortgage of the team member, the higher the cost of malicious behavior, and the better the honesty of the team member. Rewards are provided by the task publisher after the task is completed and distributed based on contributions. The penalty mechanism within the team is presented as follows:

$$punish(\mathrm{N_i}) = N_{i_{\mathrm{mortgage}}} \cdot k \tag{1}$$

where, $N_{i_{\mathrm{mortgage}}}$ is the mortgage of the data node i, and the penalty is executed by the team leader. In a real scenario, the team leader may also be malicious because this situation will cause greater losses to the team; therefore, the team leader should mortgage more. When the team sponsor has a malicious behavior, the punishment process will be performed by the other team members. k is the penalty coefficient defined as

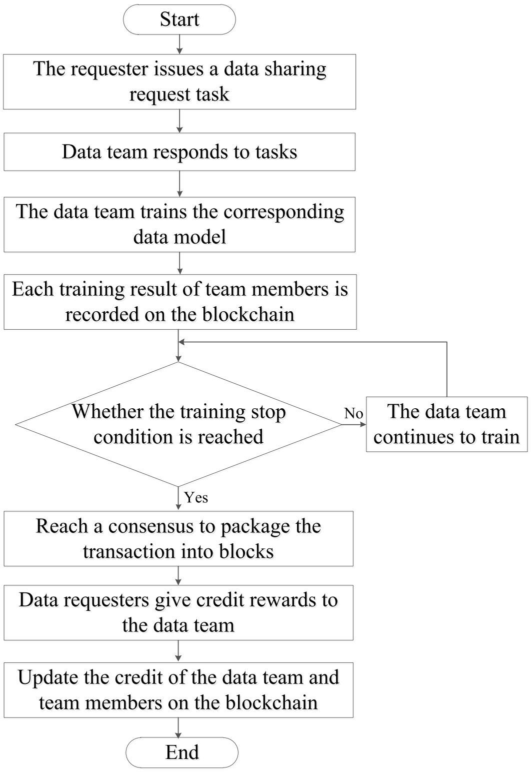

Start
The requester issues a data sharing request task
Data team responds to tasks
The data team trains the corresponding data model
Each training result of team members is recorded on the blockchain
Whether the training stop condition is reached
No
The data team continues to train
Yes
Reach a consensus to package the transaction into blocks
Data requesters give credit rewards to the data team
Update the credit of the data team and team members on the blockchain
End

Fig. 2. Architecture flow diagram.

$$\mathrm{k} = \frac{p+q}{v} \tag{2}$$

where v is the total number of work rounds to complete the collaborative task, p is the number of temporary exits, and q is the number of laziness. Thus, the compensation that each other member can get is

$$C = punish(N_i) \cdot \frac{1}{N} \tag{3}$$

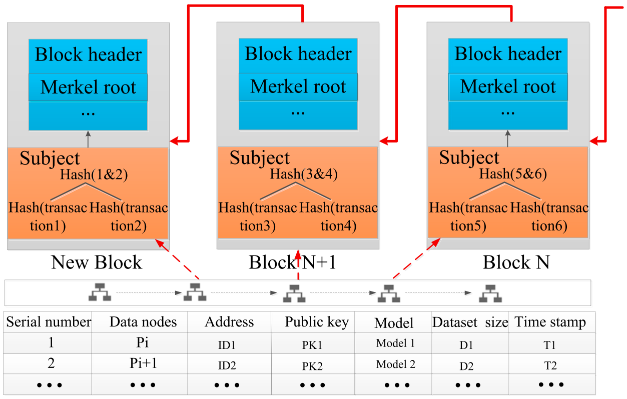
Reasonable rewards and punishments are an important guarantee for team stability. The "mortgage-punishment" mechanism plays an important role in maintaining team stability.

4.2. Data sharing process

Most of the existing data sharing methods realize the purpose of data security sharing by encrypting the data. However, in actual data sharing scenarios, encryption algorithms will reduce the data sharing efficiency. With the increasing demand for data sharing in a distributed environment, a safer and more efficient method is to share the data model instead of the original data, thereby protecting the data privacy of the data provider.

After the data requester publishes the data sharing task, the nodes that own the data will form a team to respond to the task. Specifically, after a number of nodes form a team, a member trains a data model locally selects another data node i from the team, and sends the model parameters to this node. The data node i then updates the model parameters according to the local data and again selects a data node j from the team and sends the model parameters that it has trained to the data node j. This process will be repeated until the K data nodes jointly verify that the model reaches the accuracy or maximum training time required by the requester. The specific steps for data sharing are as follows:

Step 1 Initiate a data sharing request task: The data requester initiates a data sharing request. The task contains the requester's ID,



Serial number	Data nodes	Address	Public key	Model	Dataset size	Time stamp
1	Pi	ID1	PK1	Model 1	D1	T1
2	Pi+1	ID2	PK2	Model 2	D2	T2
• • •	• • •	• • •	• • •	• • •	• • •	• • •

Fig. 3. Shared transaction storage model.

requested task category, timestamp, and task level and is signed by its private key.

Step 2 Team response task: After the data requester publishes the request task, the node connected to it will first verify its identity and then search the blockchain for whether or not the request has been processed before. The query result is directly returned if there is a cache record. If it is a new request, the task will be broadcast on the blockchain, and the data team that meets the credit requirements will respond to the task.

Step 3 Training data model: This is the data node in the above team that responds to the requested task. These nodes perform joint learning to train the global model *M*. An initial data model is generated first. A private key is then used to sign the model parameters. Signed model parameters are randomly sent to the next data node. The next data node updates the model parameters based on local data and randomly sends them to the next node. Repeat this process.

Step 4 Generate shared transaction records: All shared records between the data requester and the data node are regarded as shared transactions. These transactions will be packaged into blocks by the transaction record node. The process is shown in Fig. 3.

Step 5 Reaching consensus: The consensus process is executed by the data nodes that perform data sharing tasks. Each data node competes for the opportunity to write transaction records into the block through the work contribution mechanism. Nodes with accounting rights broadcast their blocks to other data nodes for verification. After the verification is passed, the block is added to the blockchain for further audit.

The combination of blockchain and federated learning not only solves the privacy and security issues of data sharing in distributed scenarios, but also improves the quality of shared data. The shared records of each participant can be tracked, making security audits possible. However, a consensus mechanism based on proof-of-work requires a large amount of resources. In this study, we propose a consensus algorithm based on the training model contribution to the improvement of the computational efficiency in the consensus protocol.

4.3. Consensus mechanism: proof of model contribution (POC)

We transformed the shared data problem into the shared model problem, which not only protects the data privacy of the data providers, but also solves the problem of new data requests. For the collaborative tasks, we propose a consensus algorithm based on the data node contribution. The POC can use the training results of the data nodes to reach a consensus without additional computing resources. Team members who meet the level requirements form a consensus node set responsible for promoting the consensus process and training the data model to meet the corresponding requirements of the task through cooperation. The purpose of federated learning is to train a global model as a response to the task requests. A completed model training means a completed request task.

4.3.1. Inference attack prevention

For each data node in the team, the tasks should be completed locally using the two following steps:

Step1 Use local data to update the received model parameters and broadcast the results to the other participants after the update.

Step2 If each data node has completed the iteration, enter the verification phase. Each data node will verify the received data (accuracy of the classification task and average absolute error of the regression task). If it does not meet the requirements of the requested task, it will continue training, and the verification phase will be recorded as a transaction.

Considering the inference attack launched by dishonest data requesters, the team leader should add interference to the model because doing so to the local training model of each team member will reduce the efficiency of the entire training process. In addition, because data providers have a certain degree of trust among team members, adding noise mainly prevents the inference attacks initiated by the data requesters. A model protection method based on differential privacy is designed. Given a random algorithm *G*, *O* is any subset of the set composed by all possible outputs of *G*. For two adjacent datasets *D* and D' with at most one different record, *G* satisfies:

$$\Pr\left[G(D)\in O\right] \leq \exp(\epsilon)\cdot \Pr\left[G(D^{'})\in O\right] \tag{4}$$

where, $\mathcal{E}$ represents the privacy budget, which is usually a small constant. This suggests that we can apply the Laplacian mechanism to the global model against the inference attack by

$$\widetilde{G} = G_m + Lap(\Delta f\,/\,\epsilon) \tag{5}$$

where, G_m is the global model of training, and Δf is the sensitivity, as shown in the formula:

$$\Delta f = \max_{D,D^{'}} \|G(D) - G(G^{'})\| \tag{6}$$

Algorithm 2 presents the federated learning algorithm with differential privacy. Fig. 4 illustrates the process of model training within the team.

4.3.2. Consensus based on node contribution

The consensus-reaching process is executed by the data nodes participating in the model training, which is related to the contribution of each node to the global model. The process of federated learning is that each data node is trained on a data node model; thus, the contribution of each node should be quantified to achieve fairness, that is, the greater the weight of the training result in the global model, the greater the contribution. In addition, after the local training, we obtained the local and global models of each data node. These shared transaction records are signed by the data nodes with their own private keys and broadcast to other data nodes. The node contribution serves as the proof of the node's training workload.

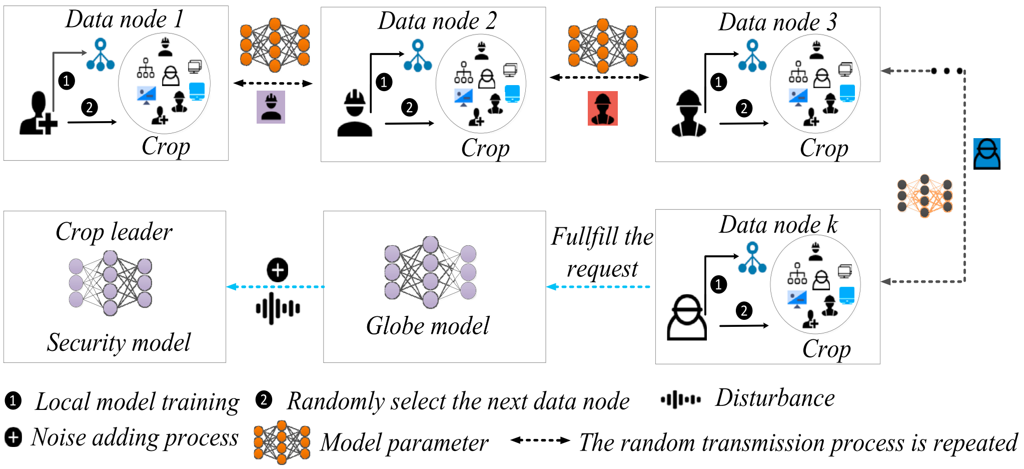


Fig. 4. Model training process.

```
Algorithm 2 DP-federated learning.
for each team member $D_t \in D_\tau$ do:
    while $G_m$ meet requirement accuracy or time $\leq$ train − time max do:
        Randomly select a participant i within the team to generate the initial model $m_i$;
        Participant $i$ randomly select $k$ from $\{1, \cdots, L\} \backslash \{k\}$ and send to $k$;
        $k$ update model $m_i$ to $m_k$ based on local data and randomly send to the next member;
    end while
end for
The leader of team disturbance the global model $M$, obtain $G_m$;
Return $G_m$ to the data requester
```

The correlation between the updated local gradient and the global model gradient is used to measure the contribution of each node. Specifically, the contribution of each data node is balanced based on the cosine similarity defined as follows:

$$\theta_k = \arccos \frac{\langle \nabla F_{k-c}(w), \nabla F(w) \rangle}{\|\nabla F_{k-c}(w)\| \cdot \|\nabla F(w)\|} \tag{7}$$

where $\nabla F_{k-c}(w) = \nabla F_k(w) - \nabla F_{k-1}(w)$, $\nabla F_{k-c}(w)$ represents the node, which is the actual update gradient of k; $\nabla F_k(w)$ is the local update gradient of the kth node; and $\nabla F_{k-1}(w)$ represents the data node k's model gradient before update; and $\nabla F(w)$ is the gradient of the global model. According to the formula, a small angle means that the actual update of the data node has a similar direction to the global model and has a positive impact on the global model, that is, the contribution to the global model is greater.

Considering that the actual contribution of a node can be measured by the above formula, to realize the reward fairness, the node that contributes more to the global model can get more rewards. Therefore, we propose a reward mechanism based on the contribution weight ratio. First, we give the mapping function, which uses a similarity-based perspective, to measure the actual contribution of the data nodes as follows:

$$f = 1 - e^{-e^{-\theta_k}} \tag{8}$$

After obtaining the contribution through the mapping function, we use the soft-max function as follows to calculate the weight ratio of the data node's contribution to the global model:

$$W_k = \frac{e^{f(\theta_k)}}{\sum_k e^{f(\theta_k)}} \tag{9}$$

At the beginning of the consensus process, the data node with the highest contribution percentage is selected as the node that records the transaction by voting. The accounting node is responsible for packaging all previously shared transactions and the global model into a block and broadcasting the block to all data nodes. The data node verifies the generated block. After the verification of each data node, the node responsible for generating the block will broadcast the block signed with its private key to all nodes and write the block to the blockchain. Another advantage of our proposed consensus mechanism is that it can prevent the lazy behavior of nodes. In the process of the multi-party cooperation training model, some lazy nodes may directly copy the previous model parameters to the next data node. We introduced a credit rating mechanism to reward or punish data nodes based on their contributions to promote honest and effective training of data nodes.

4.4. Credit management

The original credit of the team leader or each member is zero. They can get corresponding credit rewards after completing related tasks. Considering that the responsibilities of the team leader and the team member are different, their credit rewards should be different. Specifically, the team leader should be rewarded more than the team members. Similarly, if they are punished, the team leader will be punished more. A new contribution-based reward algorithm that achieves fair rewards and motivates participants to provide excellent training models is designed. In general, credit rewards should be given according to the task completion. That is, for the team leader, the reward is given by

$$C^{\text{leader}}{}_{\text{obtain}} = \frac{1}{N} C_{\text{redit}} + W_k \cdot C_{\text{redit}} \tag{10}$$

The team leader guarantees the quality of the entire data sharing process during the data sharing process and prevents data requesters from launching inference attacks; hence, they will obtain more than $\frac{1}{N} C_{\text{redit}}$ reward. C_{redit} is the credit reward provided by the task publisher, while W_k is the contribution of the weight ratio data node to the global model. The reward for each member of the team is given by:

$$C_{\text{obtain}} = \left(1 - \frac{1}{N}\right) W_k \cdot C_{\text{redit}} \tag{11}$$

The credit of each data node is updated by

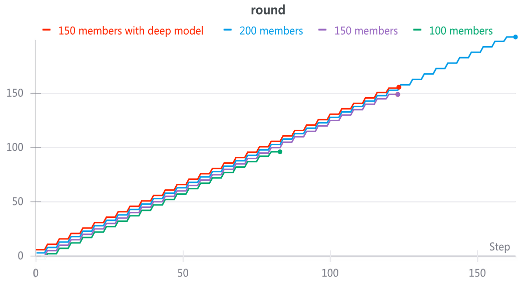


Fig. 5. Training rounds.

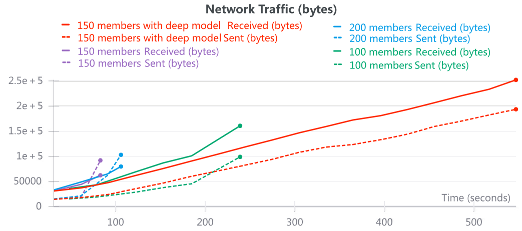


Fig. 6. Network traffic.

$$C = C_{\text{base}} + C_{\text{obtain}} \tag{12}$$

where C_{base} is the original credit accumulation of the data node.

5. Experiments

5.1. Experimental setup

The simulation was completed on a computer equipped with a Windows 7 system. The machine was equipped with an Intel Core i7 processor with 6.4 GHZ CPU frequency. The Python programming language was used to verify the effectiveness of the proposed scheme. In this section, we perform an experimental verification on the proposed scheme. First, we verified the effectiveness of the proposed data-based team to complete shared tasks and conducted experiments on the performance of running on the blockchain.

5.2. Experimental results

We conducted an evaluation on the mnist dataset containing 0–9 number categories. The size of each picture was 28*28, which is widely used for the evaluation of the classification tasks. We used this data set to simulate data fragments in the IoT. We randomly divided the mnist dataset into multiple partial datasets to simulate the situation that each data node has small-scale data in reality. Each team member randomly had $0-20$, 20–40, 40–60, 60–80, 80–100, 100–120, 120–140, 140–160 and 160–180 different data sizes, with the local batch size set to 128. We will evaluate the proposed scheme from two aspects: different numbers of team members and different local training models. The first is to experiment from the perspective of the number of team members. The experiment set up three teams of different sizes with different numbers of members, each with 100, 150, and 200 people. The size of the dataset and the quality of the team member dataset are randomly distributed, but the team leader will review and screen each team member before forming the team. We believe that the data quality can be guaranteed to a certain extent.

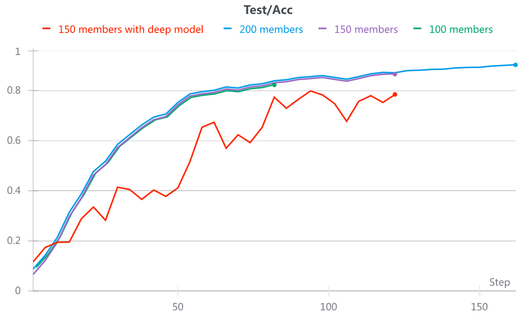


Fig. 7. Test accuracy.

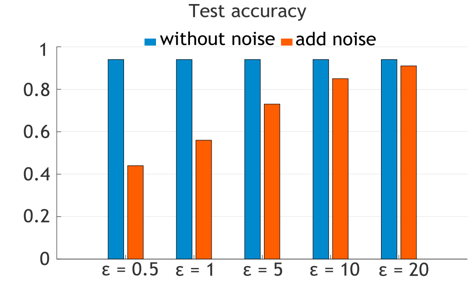


Fig. 8. Accuracy of adding disturbance.

Fig. 5 shows the communication rounds and steps performed within each team. There are 83 steps in a data group of 100 people, 123 steps in a data group of 150 people, and 163 steps in a data group of 200 people.

Fig. 6 illustrates the network traffic of each team during the model training period. The network load of a 200-person team is not the largest, indicating that the proposed solution is feasible, and completing the data sharing tasks in the form of a team will not affect the efficiency.

Fig. 7 shows the accuracy of the global model delivered by teams of different sizes. The classification accuracy increases as the data scale increases. The red curve in Fig. 7 depicts the training result of a 150-person team using a deeper neural network. At the end of each team member's iteration, its accuracy is lower than the training result of a smaller neural network layer. Therefore, in addition to considering the data size, the team must also evaluate the models used in training.

We took $\Delta f = 1$ in formula (4), transformed the value of ϵ, and added disturbance to the global model trained by a team of 200 people. The results are shown in Fig. 8. The larger the value of ϵ, the larger the privacy budget. The greater the usability, the higher the test accuracy rate. When $\epsilon = 20$, the test accuracy rate can still reach 90%. Even if there is a certain amount of interference, the model's usability can be guaranteed. The experiments proved that the solution to prevent the data requester from launching inference attacks is effective.

In addition, we selected eight data nodes and calculated the cosine similarity with the global model. As shown in Fig. 9, three of them had a negative impact on the overall model training. In actual scenarios,

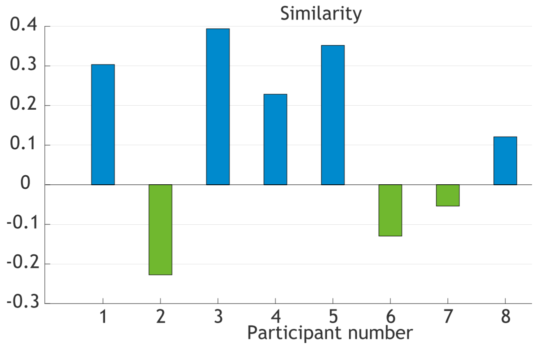
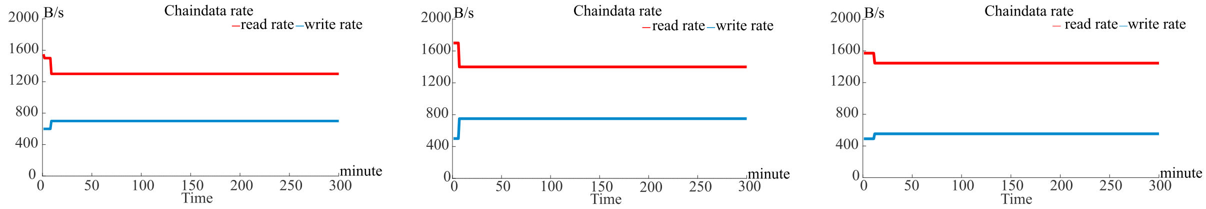


Fig. 9. Similarity test.

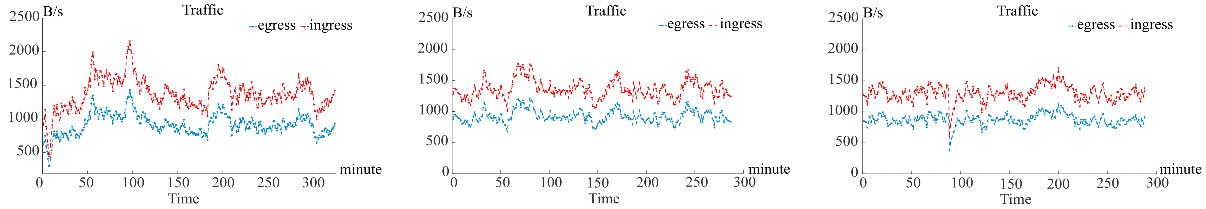


(a) Team of 100 members.

(b) Team of 150 members.

(c) Team of 200 members.

Fig. 10. Read write rate.



(a) Team of 100 members.

(b) Team of 150 members.

(c) Team of 200 members

Fig. 11. Network status.

different effects on the global model training will be observed because each person's data quality and computing ability are different.

We verified the proposed scheme on the Ethereum platform and set the transaction volume of the blockchain network to four transactions per second. According to the strategy mentioned herein, we design smart contracts such that the nodes that contribute the most packaged transactions will receive corresponding rewards. The block generation rate was one block every 2 s. The system and blockchain performance data were written into the influxdb database. As a time series database, the influxdb database can record the performance indicators that change over time. As shown in Figs. 10 and 11, we obtained the read and write rate of blockchain data in the database and system network status. We also verified that the team members who have contributed the most to the global model were responsible for packaging the transaction records of each shared process into blocks for teams of 100, 150, and 200 people.

6. Conclusions

The data sharing between IoT devices helps to improve the quality of service and user experience in various IoT applications; however, data sharing may lead to the privacy disclosure of data providers. To solve this problem, a secure data sharing mechanism, called BP2P-FL, was proposed herein using peer-to-peer federated learning, which can protect data providers' privacy by realizing the model sharing instead. Moreover, the blockchain was introduced within the data sharing, in which every training process is recorded to ensure that the data providers offer high-quality data. For further privacy protection, the differential privacy technology was applied to the global data sharing model. The experimental results showed that BP2P-FL has an excellent performance in accuracy and feasibility.

Declaration of competing interest

The authors declare that there is no conflict of interests.

Acknowledgment

This work is supported by National Natural Science Foundation of China under Grant No. U1905211 and 61 702 103, Natural Science Foundation of Fujian Province under Grant No. 2020J01167 and 2020J01169.

References

[1] K. Shafique, B.A. Khawaja, F. Sabir, S. Qazi, M. Mustaqim, Internet of Things (IoT) for next-generation smart systems: a review of current challenges, future trends and prospects for emerging 5G-IoT scenarios, IEEE Access 8 (2020) 23022–23040.

[2] T.P. Ezhilarasi, N. Sudheer Kumar, T.P. Latchoumi, N. Balayesu, A Secure Data Sharing Using IDSS CP-ABE in Cloud Storage, in Advances in Industrial Automation and Smart Manufacturing, 2021, pp. 1073–1085. Singapore.

[3] A. Bozorgchenani, F. Mashhadi, D. Tarchi and S. Salinas Monroy, Multi-objective computation sharing in energy and delay constrained mobile edge computing environments, IEEE Trans. Mobile Comput., DOI: 10.1109/TMC.2020.2994232.

[4] L. Liu, et al., Blockchain-enabled secure data sharing scheme in mobile-edge computing: an asynchronous advantage ActorCCritic learning approach, IEEE Internet Things J. 8 (4) (2021) 2342–2353.

[5] X. Zhou, X. Xu, W. Liang, Z. Zeng and Z. Yan, Deep learning enhanced multi-target detection for end-edge-cloud surveillance in smart IoT, IEEE Internet Things J.l, DOI: 10.1109/JIOT.2021.3077449.

[6] T. Li, A.K. Sahu, A. Talwalkar, V. Smith, Federated learning: challenges, methods, and future directions, IEEE Signal Process. Mag. 37 (3) (2020) 50–60.

[7] B. Cao, et al., Performance analysis and comparison of PoW, PoS and DAG based blockchains, Digit. Commun. Netw. 6 (4) (2020) 480–485.

[8] Q. Liu, Y. Xu, B. Cao, L. Zhang, and M. Peng, Unintentional forking analysis in wireless blockchain networks, Digit. Commun. Netw. 7 (3) (2021) 335-341, DOI: 10.1016/j.dcan.2020.12.005.

[9] Y. Gao, Y. Chen, H. Lin, J.J.P.C. Rodrigues, Blockchain Based Secure IoT Data Sharing Framework for SDN-Enabled Smart Communities, IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops, 2020, pp. 514–519, https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162725.

[10] H. Xu, Q. He, X. Li, B. Jiang, K. Qin, BDSS-FA: a blockchain-based data security sharing platform with fine-grained access control, IEEE Access 8 (2020) 87552–87561.

[11] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, PrivySharing: a blockchain-based framework for privacy-preserving and secure data sharing in smart cities, Comput. Secur.y, DOI: 10.1016/j.cose.2019.101653.

[12] K. -P. Yu, L. Tan, M. Aloqaily, H. Yang and Y. Jararweh, Blockchain-enhanced data sharing with traceable and direct revocation in IIoT, IEEE Trans. Ind. Inf.s, DOI: 10.1109/TII.2021.3049141.

[13] X. Zhou, W. Liang, J. She, Z. Yan, and K. Wang, Two-layer federated learning with heterogeneous model aggregation for 6G supported Internet of Vehicles, IEEE Trans. Veh. Technol.y, DOI: 10.1109/TVT.2021.3077893.

[14] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, IEEE Trans. Ind. Inf. 16 (10) (2020) 6532–6542.

[15] F. Sattler, S. Wiedemann, K.-R. Müller, W. Samek, Robust and communication-efficient federated learning from non-i.i.d. Data, IEEE Transact. Neural Networks Learn. Syst. 31 (9) (2020) 3400–3413.

[16] A. Imteaj, M.H. Amini, Distributed Sensing Using Smart End-User Devices: Pathway to Federated Learning for Autonomous IoT, 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 1156–1161, https://doi.org/10.1109/CSCI49370.2019.00218.

[17] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, IEEE Trans. Ind. Inf. 16 (6) (2020) 4177–4186.

[18] Y. Lu, X. Huang, K. Zhang, S. Maharjan, Y. Zhang, Blockchain empowered asynchronous federated learning for secure data sharing in Internet of Vehicles, IEEE Trans. Veh. Technol. 69 (4) (2020) 4298–4311.

[19] L. Yin, J. Feng, H. Xun, Z. Sun and X. Cheng, A privacy-preserving federated learning for multiparty data sharing in social IoTs, IEEE Trans. Netw. Sci. Eng.g, DOI: 10.1109/TNSE.2021.3074185.

[20] H. Chai, S. Leng, Y. Chen and K. Zhang, A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in Internet of Vehicles, IEEE Trans. Intell. Transport. Syst.s, DOI: 10.1109/TITS.2020.3002712.