Hashgraph Based Federated Learning for Secure Data Sharing

Xiuxian Zhang[1,2], Lingyu Zhao[1], Jinfeng Li[1], and Xiaorong Zhu[1(✉)]

[1] Nanjing University of Posts and Telecommunications, Nanjing 210003, China
xrzhu@njupt.edu.cn
[2] Nanjing Xiaozhuang University, Nanjing 211171, China

Abstract. As the key technology of connected intelligence, the importance of artificial intelligence has increased rapidly. It is worth to note that the most critical challenge is the secure data sharing which is stored in different area and belonged to different organization. With this in mind, a hashgraph based federated learning for secure data sharing model is proposed to protect user privacy and detect the dishonest model provider. In terms of technologies, detection of the local model is added to the hashgraph consensus processing, and only if the supermajority (more than 2/3) of the participants agree, the local model could be adopted. Therefore, the accuracy and convergence rate of the federated learning both increased largely. On the other hand, the asynchronous working mode of hashgraph can greatly reduce network overload. Simulation results show that the hashgraph based federated learning enables the data sharing more secure and reliable.

Keywords: Hashgraph · Federated learning · Blockchain · Gossip · Virtual voting

1 Introduction

With the advent of the 6G, connected things have gradually transformed into connected intelligence. Artificial intelligence, as the key technology of connected intelligence, has attracted more and more attention of the researchers. However, up to now, there are two big challenges of artificial intelligence. The first one is data sharing, which is stored in different areas and belonged to different organizations. The second one is to guarantee the privacy and security of the shared data during model training. The federated learning (FL), which obtains a central model on the server by aggregating models trained locally on clients [1], can solve the problems.

In FL, the distributed local devices compute their local model based on local data samples and send them to a central server. The central server trains a shared model by aggregating the local models received from different devices [2]. Therefore, the raw data stays in the local devices all the time during the training. Notably, not only data shared but also privacy protection both realized in FL. Nonetheless, there are several limitations in FL. For the first, the reliable privacy of the learning model from each device cannot be

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 358, pp. 556–565, 2021.
https://doi.org/10.1007/978-3-030-69072-4_45

guaranteed. Secondly, dishonest users can have an adverse impact on the global model by offering the low-grade local model. Besides, users also lack the motivation to participate in the FL using their own computing resources and data. The last one is the problem of network overload. Massive amounts of models are transmitted at the same time during FL, which will cause network overload under the constraint of bandwidth.

In recent years, lots of researchers have been engaged in the research of FL combined with blockchain to solve the above problems. The blockchain was used to store the retrieval data and access rights in paper [3], which could prevent malicious users to temper the models. And, the differential private algorithm was used to protect personal privacy data. However, the use of differential private algorithm leads to a sharp decline of the data availability due to the random noise interference. FL was investigated based on the fabric channel in paper [4]. The decentralized FL request was proposed in one channel with the same type of user data to guarantee the personal privacy of users in different channels, but it does not involve the personal privacy protection among users in the same channel. The blockchain and FL are coupled to ensure the privacy of the user data. The trained learning model parameters can be stored on the blockchain securely in an immutable manner against unauthorized access and malicious actions [5]. Y. J. Kim et al. [4] proposed a blockchain-based FL to provide an incentive mechanism and prevent malicious users from changing the models according to the natural trading attributes and immutable ledger of blockchain. Besides, a fast and stable target accuracy convergence joint learning model was proposed to reduce the network overload.

To summarize, although there exist a number of studies on blockchain based on FL, the results in those studies fail to consider the dishonest model provider. To fill this gap, a hashgraph based FL is proposed to detect dishonest model provider. The main contributions of this paper are mainly given as following:

- In order to detect the dishonest model provider, detection of the local model is added to the hashgraph consensus processing. Theoretically, the local model will be adopted only if the supermajority (more than 2/3) of the participants agree.
- The hashgraph with shorter consensus time is adopted to reduce the training time. Typically, the asynchronous working mode of hashgraph can greatly reduce network overload.

The remainder of this paper is organized as following. In Sect. 2, the system model is established. And, the hashgraph for data sharing is proposed in Sect. 3. Then, FL is described in detail in Sect. 4, Numerical results are presented in Sect. 5, followed by a conclusion in Sect. 6.

2 System Model

In this paper, a common distributed data sharing scenario with multiple parties is considered. The system can be divided into two parts shown as Fig. 1: the blockchain platform and communication network. The blockchain platform adopts hashgraph consensus to reduce the consensus time and network overload. Specifically, the blockchain platform is used to record the local model retrieval (the raw model parameters are stored in local

devices), the availability of local models, and all the sharing data events which can trace the use of data for further audit. The communication network takes charge of data communication.

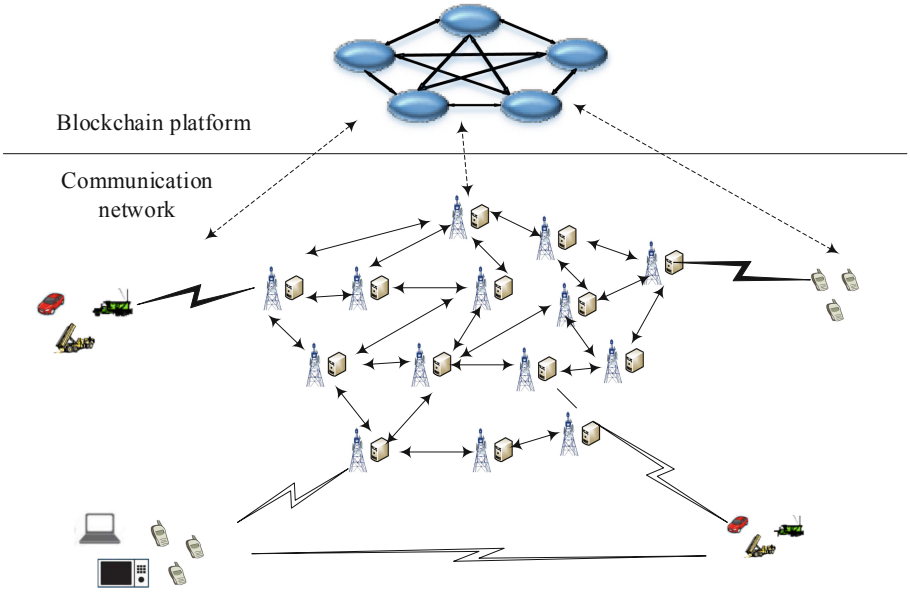

Fig. 1. System model

All users who want to provide data sharing service can apply to join the blockchain platform. A data sharing requester lunches a request to the blockchain platform, the blockchain will check whether the request has been processed before. If there is a hit, the request will be forwarded to the node that has cached the results and the cached results are then sent to the requester as a reply. In contrast, a new FL request with data categories and incentive mechanism will be published in the platform, then, the data provider participated in the blockchain will choice whether join the FL or not according to the matched-degree of the data they have with the data requested and incentive mechanism. All nodes participated in the FL are regarded as committee nodes which are responsible for driving the consensus in the blockchain.

3 The Hashgraph for Secure Data Sharing

In this paper, we consider the problem of privacy-preserving and dishonest model provider checking in the data sharing process with decentralized multiple parties. In order to protect the privacy of local models, homomorphic encryption is used to encrypt the local models and then transmit the encrypted models to the next node. The local models are transmitted among the committee nodes. All the committee nodes check the availability of the local models and vote on them. Only more than 2/3 of the committee nodes agree. The local models are legitimate.

Hashgraph [6] is a relatively novel DAG (Directed Acyclic Graph) technology, which is a consensus of blockchain 3.0. Theoretically, the hashgraph is an aBFT (Asynchronous Byzantine Fault Tolerance) system, with no node that can prevent the network from reaching consensus or modify data after consensus has been reached, and it can achieve bank-level security. Lots of problems of the traditional blockchain can be solved in hashgraph, such as long consensus time, the lack of concurrent processing mechanism to meet the large scale application scenario, and too high transaction costs in small transactions. To this end, the hashgraph is adopted in this paper. In terms of technologies, the hashgraph is different from the previous blockchain, the nodes of the hashgraph package the transactions into events, which are the smallest data units, communicate with other nodes through gossip protocol, and reach the consensus relying on the virtual voting protocol.

3.1 Event Structure

Events as the smallest unit of the hashgraph mainly include a timestamp, transaction information, self-parent hash, other-parent hash, and a digital signature, as shown in Fig. 2. Where the self-parent hash is the hash of the last event on this node and other-parent hash is the hash of the last event on the other nodes. Transaction information is the main content of the event mainly includes local model retrieval, sharing data information, the support number of the local model. In more detail, the local model retrieval is to record the retrieval information of the local model, the sharing data information contains data categories and data quantity. The support number of the local model is the number of the node who vote "yes" to the local model.

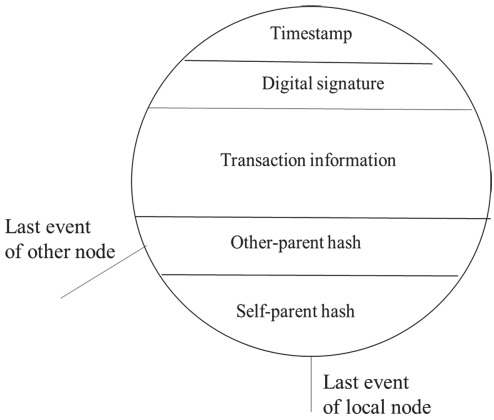

Fig. 2. The event structure

3.2 Gossip Protocol

The communication mechanism uses gossip protocol. In the gossip protocol, the node randomly selects another node and sends data known in this node but not known in

the selected node to the selected node. When the node receives the data containing new transactions, it first executes all the new transactions and checks the availability of the local model, and then repeats the same process by selecting another random node. Therefore, the protocol of gossip spreads exponentially until all the nodes receive the event.

For the sake of simplicity, assume that there are four nodes A, B, C, D in hashgraph, as depicted in Fig. 3. And, define the event generated by the node A as A1, A2, A3…in order, define the node B, C, D in the same way. At the beginning of the gossip protocol, node A randomly selects a node from B, C, D, here select B for example, to send transactions that are known in A but not known in B and package the transactions into events A1. After node B receiving event A1, it will deal with the new transactions, which is local model checking in this paper, contained in event A1. Additionally, Node B repeats the same process of node A that select node D to transmit event B2 which includes the information of event A1 if node D hasn't received the transactions before. It is obvious to note that, the transactions of event A1 will be known by all nodes A, B, C, D ultimately.

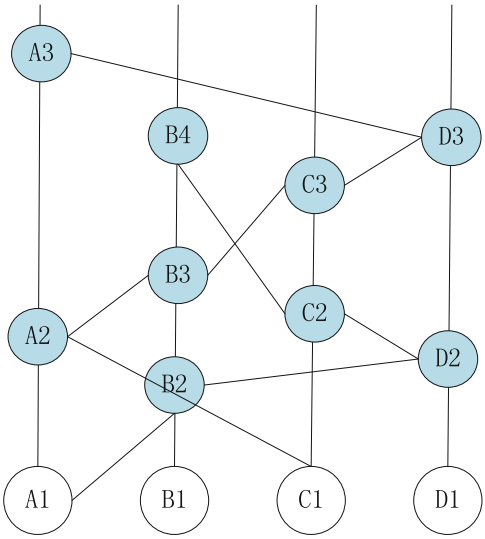

Fig. 3. Gossip protocol

3.3 Virtual Voting Protocol

As evident from above, the communication of each node has been completed. Ultimately, all nodes participated in the FL have stored the complete history of the transaction. But, this is just a communication step, and a virtual voting protocol is needed to reach a consensus among the nodes. When a consensus is proposed, there is no need for large-scale message communication due to the gossip protocol, each node performs the

voting algorithm independently, and all nodes will reach the same consensus result [7]. Therefore, zero bandwidth is used, beyond simply gossiping in the hashgraph. Therefore, it can sharply reduce the network overload. Virtual voting processing is divided into 3 steps: divide rounds, decide famous witnesses, determine round received and consensus time.

To understand the process of virtual voting, it's worth describing some terminology about hashgraph, such as witnesses, round, seeing, strongly seeing, and famous witness, before discussing the specific voting steps. Conventionally, the first event created by the node is called witness and the witness is the beginning of the round (r) for the node. As such, an event Y could see event X only if event Y can find X by the pointer of hash. Typically, when all paths of event Y could find Event X go through supermajority (more than 2/3) of nodes called Y strongly sees X. Correspondingly, if a witness of round r can be seen by supermajority of witnesses of round $r+1$, then the witness of round r is a famous witness.

Divide Rounds: A witness event is the beginning of a round (r) known from the definition of round. Assume that node B receives the event X send by node A and node B will create event Y to send to node C. Before creating event Y, node B should check whether need to start a new round, if event X can see supermajority of witnesses of r round, the event Y is the beginning of $r+1$ round. Meanwhile, Y is the witness of $r+1$ round. Otherwise, event Y is still in r round.

Decide Famous Witness: A witness in round r is called famous if and only if it can be seen by supermajority witnesses in round $r+1$. If witness Y of node B in round $r+1$ could see witness X of node A in round r, then witness Y will vote yes to witness X. Witness Z in round $r+2$ will collect the votes about the event X from the witnesses that are seen strongly by Z in round $r+1$. If the amount of votes is not less than the number of the supermajority nodes, then the event X is a famous witness. Hashgraph has proved mathematically that if any of the witnesses in $r+2$ round make a decision on the result of the vote, the result will be the conclusion of the network, and if the witnesses in $r+2$ round can not make a decision, the next round of witnesses will collect the votes until a firm conclusion is reached [7].

Determine Round Received and Consensus Time: Assuming that all witnesses in round r have been determined whether they are famous witnesses or not, the receiving round of event X is r when all known witnesses of round r can see the event. Find all first events $Y = \{Y_1, Y_2, Y_3, \ldots\ldots Y_m\}$ that can see X on the paths from X to all famous witnesses in round r. Then, the median of the timestamp of event $\mathrm{Y}_i i = [\frac{1+m}{2}]$ is the consensus time.

4 The FL for Data Sharing

In this section, the FL model based on hashgraph is described in detail showed in Fig. 4. The process of the FL can be divided into two parts: the global model and the local model. At global iteration t, the local model $w_i(\mathrm{t})$ is trained based on its own data set D_i

on each device. Then, the encrypted model parameter $w_i^{'}(t)$ is transmitted to the other participants of the blockchain platform to check the availability of the local model. When the consensus of the event include the local model is complete, the global model will be concluded based on the local model and the vote result of the committee nodes.

Local Model: In this paper, the local machine learning adopts linear regression. As known to all, the learning objective of node i is to minimize the loss function $l(w_i)$ over all the data samples $D_i = \{x_{ij}, y_{ij}\}$ where $x_{ij} \in \Re^d$, $y_{ij} \in \Re$. Then, the local model$w_i(t)$, local loss function (w_i), and global loss function $L(w)$ is described as:

$$l(w_i(t)) \triangleq l\left(w_i(t), x_{ij}, y_{ij}\right) = \frac{1}{2} \parallel y_{ij} - w_i^T(t) x_{ij} \parallel^2 \tag{1}$$

$$w_i(t) = arg\ min\ l(w_i(t)) \quad w_i(t) \epsilon \Re^d \tag{2}$$

$$L(w(t)) = \frac{1}{N_i} \sum_{i=1}^{i=I} m_i * l(w_i(\mathrm{t})) \tag{3}$$

Where N_i is the number of the committee nodes who support $w_t'(t)$ and $m_i = 0$, if the node i is a dishonest model provider, otherwise, $m_i = 1$.

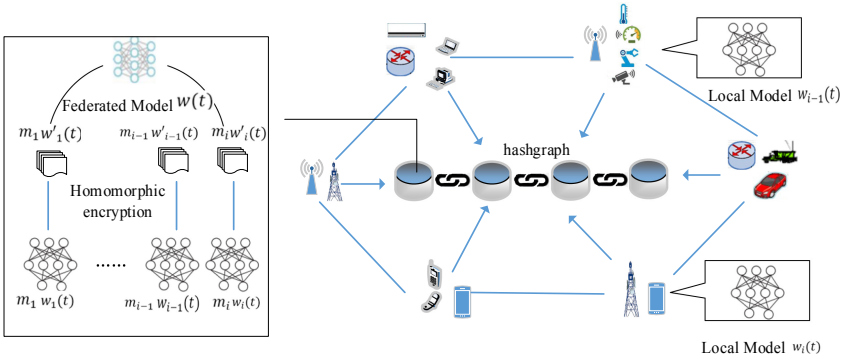

Fig. 4. FL model

For each global model iteration t, assume that the local model of the device i is updated for R epochs. At epoch r, the local model for device i is updated by stochastic variance reduced gradient (SVRG) as [8]:

$$w_i^r(t) = w_i^{r-1}(t) - \lambda \nabla \varphi \tag{4}$$

$$\nabla \varphi = \left[\nabla l\left(w_i^{r-1}(t)\right) - \nabla l(w_i(t)) + \nabla L(w(t))\right] \tag{5}$$

Where $\lambda > 0$ is step-size and after R local epochs $w_i(t) = w_i^R(t)$. $\nabla L(w) = \frac{1}{N_i} \sum_{i=1}^{i=I} m_i * \nabla l(w_i)$. Furthermore, after the local model $w_i(t)$ is trained, $w_i(t)$ and local loss function $\nabla l(w_i(t))$ are encrypted to $w_i^{'}(t)$and$\nabla l^{'}(w_i(t))$. Correspondingly, they are transmitted to the other participants of the blockchain platform to check the availability of the local model.

Global Model: The preliminary weight parameters at global iteration $t = 0$ are randomly chosen from pre-selected range $w_i(0), w(0) \in [0, w_{max}]$, $\nabla l(w(0)) = (0, 1]$. When the consensus of the event including the local model $w_i^{'}(t)$and$\nabla l^{'}(w_i(t))$ is complete, read the number of the committee nodes N_i who support $w_i^{'}(t)$ from blockchain. And, if it is less than 2/3 amount of committee nodes, the node i is a dishonest model provider and $m_i = 0$, otherwise, $m_i = 1$. When all local models of the committee nodes are confirmed, the global model can be concluded as:

$$w(t) = \sum_{i=1}^{i=I} m_i * \mu_i * w_i'(t) \tag{6}$$

Where $0 \leq \mu_i \leq 1, \mu_i = \frac{|D_i|}{|D|} * \frac{N_i}{I}$ is the weight coefficient of $w'_i(t)$ with $D = \sum_{i=1}^{i=I} D_i$ is the data set of all devices. Then all the nodes update $w(t)$ from blockchain and start a new round of local and global training.

5 Performance Evaluation and Analysis

For the sake of presentation, the performance of the proposed method with the detection of dishonest providers is compared with the conventional FL. Where the total number of the local model provider I is 10, and assume that there is one dishonest provider. Then, divide the dataset into 9 fragments. Simultaneously, randomly generate a dataset as the dishonest provider's dataset and assume that the proportion of the data provided by the dishonest provider is 0.2. It is worth to notice from Fig. 5 that the performance of the FL based on hashgraph is better than the conventional FL in the aspects of precision, recall, f-measure and accuracy. As shown in Fig. 6, it is obvious that the more proportion of the data provided by the dishonest provider, the lower accuracy of the FL model will be in the conventional FL. But, it makes no difference in the proposed FL.

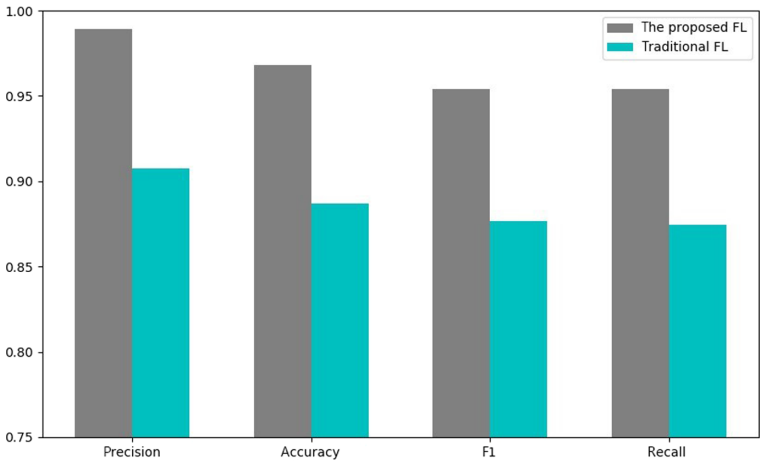

Fig. 5. Performance of the FL proposed comparing with the conventional

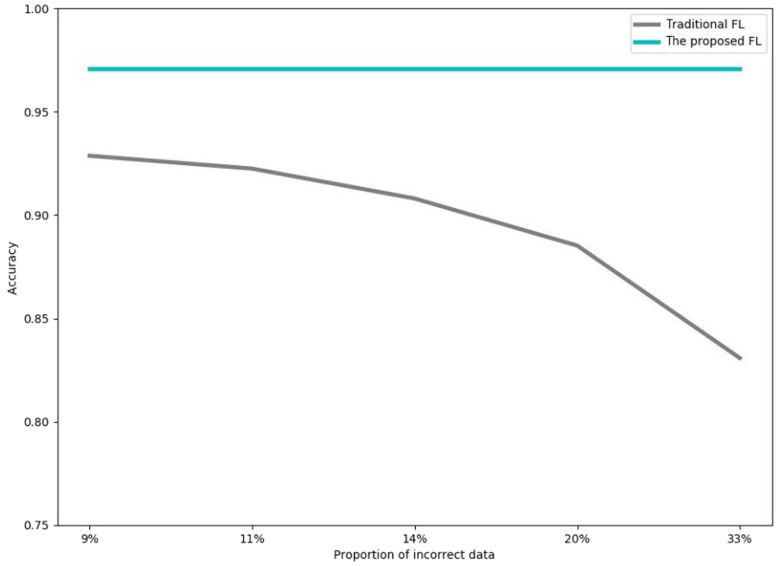

Fig. 6. The accuracy with different proportion of incorrect data provided by dishonest providers

Through the above evaluation, it is important to note that the performance of the hashgraph based FL is better than the conventional FL due to the detection of the dishonest provider. As pointed by the previous work, the hashgraph based FL enables the data sharing more secure and reliable.

6 Conclusion

To summarize, a hashgraph based FL for secure data sharing model is proposed to protect user privacy and detect the dishonest model provider. According to the analysis, the system can effectively check out the dishonest model provider and protect the privacy of users. On the other hand, the hashgraph with shorter consensus time can reduce the training time largely. Typically, the asynchronous working mode of hashgraph can greatly reduce network overload. Ultimately, simulation results show that the hashgraph based FL enables the data sharing more secure and reliable.

References

1. Chen, Y., Sun, X., Jin, Y.: Communication-efficient federated deep learning with asynchronous model update and temporally weighted aggregation. IEEE Trans. Neural Netw. Learn. Syst. **31**, 4229–4238 (2019)
2. Zappone, A., Di Renzo, M., Debbah, M.: Wireless networks design in the era of deep learning: model-based, AI-based, or both? IEEE Trans. Commun. **67**, 7331–7336 (2019)
3. Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y.: Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. IEEE Trans. Ind. Inf. **16**(6), 4177–4186 (2020)

4. Majeed, U., Hong, C.S.: FLchain: federated learning via MEC-enabled blockchain network. In: 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, pp. 1–4 (2019)
5. Salah, K., Rehman, M.H.U., Nizamuddin, N., Al-Fuqaha, A.: Blockchain for AI: review and open research challenges. IEEE Access **7**, 10127–10149 (2019)
6. Baird, L., Harmon, M., Madsen, P.: Hedera: A Public Hashgraph Network & Governing Council. Hedera Hashgraph, LLC., Whitepaper V.2.0, August 2019. https://www.hedera.com/hh-whitepaper-v2.0-17Sep19.pdf.
7. Baird, L.: The swirlds hashgraph consensus algorithm: fair, fast, byzantine fault tolerance, May 2016. https://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf
8. Konen, J., Mcmahan, H.B., Ramage, D., et al.: Federated optimization: distributed machine learning for on-device intelligence. Edinburgh Research Explorer - University of Edinburgh (2016)