





DRL and Blockchain Empowered Federated Learning Framework for Genetic Data Engineering

Yuxuan Zhong¹ (✉), Zihan Li¹, Siya Xu¹ , Shaoyong Guo¹ , Xingyu Chen¹,
Tao Shen², and Lin Huang³

¹ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

{xxuan, xusiyaxsy, syguo, chenxy}@bupt.edu.cn

² Kunming University of Science and Technology, Kunming, China
shentao@kust.edu.cn

³ Yunnan Provincial Academy of Science and Technology, Kunming, China

Abstract. With the development of genetic data engineering and edge intelligence, more and more intelligent applications and services are trained in the edge side. However, the centralized training mode has the problems of high transmission delay and user privacy disclosure, while federated learning (FL) can protect the privacy of users, and reduce data transmission costs by distributing the training work. Existing FL schemes often ignore the impact of low-quality training nodes and the security issues in the data transmission process. To improve the accuracy of the FL model, we design a node selection algorithm based on deep reinforcement learning (DRL). In addition, we use blockchain for model transmission to complete the global aggregation of FL to enhance the security and reliability of model parameters. We design a blockchain empowered FL framework and further propose a two-layer consensus algorithm based on PBFT to improve consensus efficiency, reduce consensus delay and reduce communication resource consumption. Simulation results show that the proposed node selection algorithm outperforms other compared algorithms, and can well improve the accuracy of the model and reduce the loss function. The proposed consensus algorithm can balance the consensus efficiency and communication resource consumption.

Keywords: Blockchain · Consensus algorithm · DRL · Genetic data engineering

1 Introduction

Mobile terminals constantly generate a large number of different types of data, including genetic data, material information and multimedia data. These data are often used for the training of services models. However, the traditional centralized training mode requires users to upload all data, which will lead to the problem of privacy disclosure. At the same time, the central server in the centralized training mode needs to obtain a large

amount of terminal data, which may lead to server overload or congestion due to too much network traffic [1].

Because of the huge scale of the genetic engineering, the information of terminals is easy to be tracked, which leads to the disclosure of users' security and privacy. FL can well solve the problem of privacy disclosure. It enables different data owners to collaborate without exchanging data by designing a virtual model. However, the security problem in the data transmission process have not been solved.

Blockchain has tamper-proof, anonymity, and traceability functions. It does not rely on additional third-party management agencies, and works with no central control. Each blockchain node realizes self-verification, transmission and management of information through distributed accounting and storage. Using blockchain for model transmission enables secure data sharing in the terminal edge network.

Introducing blockchain into the aggregation process can prevent the models from attacks and malicious tampering [2]. Several schemes [3–5] use blockchain for model aggregation and update tasks, avoiding the problem of a single point of failure caused by central server storage. [6] proposed a novel blockchain to solve the critical message propagation problem in VANET, improving node trustworthiness and message reliability. Lu [7] proposed a data sharing scheme based on blockchain and FL, using blockchain to ensure trusted sharing and FL to protect user privacy, which ensures the accuracy and security of the model. However, the above schemes ignore the delay incurred in the blockchain consensus process.

To solve the above problems, a DRL and blockchain empowered FL framework is designed for genetic data engineering. We first optimize the node selection, solving the problem of low model accuracy caused by malicious nodes. Then, we divide nodes into primary network layer nodes generating blocks and secondary network layer nodes uploading models, so the institution node with a higher trust value can obtain the billing right more easily, which improves the security and credibility of data in genetic engineering. Since the consensus processes of secondary network layer and primary network layer are asynchronous, the consensus efficiency is improved.

2 System Model

2.1 Network Architecture

As shown in the figure, we propose a blockchain empowered FL framework for genetic data engineering. Considering that the system includes multiple terminals and multiple institutions, the terminals are distributed in the mobile layer and train the local models with their own data, while institutions are distributed as edge nodes in the edge layer and aggregate models, with MEC servers deployed on them.

To prevent attacks and malicious tampering of models trained by a single institution, we store local models in blocks and make all institutions work together to aggregate global models by using blockchain technology. Multiple institutions collect and aggregate their models simultaneously, and decentralize model updates and storage through the blockchain's consensus mechanism and public storage. The blockchain records globally aggregated model parameters, model accuracy, sender identity information, data signatures, and data sharing events between institutions (Fig. 1).

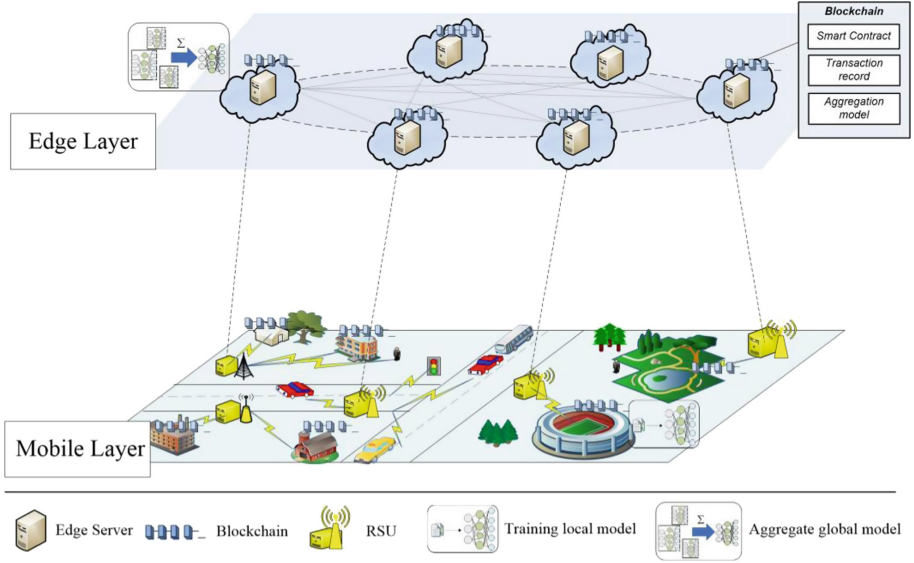


Fig. 1. The DRL and blockchain empowered FL framework for genetic data engineering

2.2 Consensus Delay

The block propagation delay of miner i is set to:

$$\xi_{lock}^p = \frac{H_{data}t_{block}}{R} + \frac{2d_n}{c} \tag{1}$$

where, t_{block} is the number of transactions within the block, H_{data} is the amount of one-transaction data, R is the data transmission rate, d_n is the distance between nodes and c is the speed of light [8].

Since transaction verification consumes a fixed amount of computing power, assume that authentication time and verification time has a linear relationship with the number of transactions in the block, denoted as:

$$\xi_{block}^v = l \cdot t_{block} + k \cdot t_{block} \tag{2}$$

where l is a parameter determined by the network size and the average verification speed of blockchain nodes. Therefore, the sum of propagation time, authentication time and validation time of a block with t_{block} transactions are:

$$\xi_{block} = \frac{H_{data}t_{block}}{R} + \frac{2d_n}{c} + (k + l) \cdot t_{block} \tag{3}$$

2.3 Communication Resource Model

The concept of communication resource consumption in the consensus process is modeled using the communication count F . Assuming that there are M nodes in the network

participating in the master node selection phase, the formula for calculating the number of node communications in the network is:

$$T_a = N - 1 \quad (4)$$

$$T_b = (N - 1)^2 \quad (5)$$

$$T_c = N(N - 1) \quad (6)$$

$$F = T_a + T_b + T_c \quad (7)$$

where T_a is the number of communications from the previous primary node sending the current state, T_b is the number of communications from each node initiating a vote, and T_c is the number of communications receiving confirmation from most candidate nodes in the current round.

3 FL Framework

3.1 Training Node Selection Algorithm

Terminal devices have different computing power and data sets, which will affect the accuracy of the local model and training delay. Therefore, we need to select a group of appropriate terminals as training nodes to make the training results optimal and the training delay controlled within a certain range.

We consider summing the loss functions to represent the training performance of training nodes in FL. We propose an optimization problem to maximize the accuracy of the global model by optimizing the vehicle selection decision. Then, the node selection problem can be expressed as:

$$\begin{aligned} \min \quad & L(x_{test}, y_{test} : \omega) \\ \text{s.t.} \quad & C1 : a_i = \{0, 1\}, \forall i \in \mathbf{n} \\ & C2 : t_{training} \leq \sigma \end{aligned} \quad (8)$$

where constraint C1 ensures that the terminal selection decision is valid, and constraint C2 ensures that the training delay is not higher than a threshold. We use DRL to solve the optimization problem. We consider the available resources of the terminals as the state space, the selection decision of the terminals as the action space, and the optimization objective as the reward function.

3.2 The Hierarchical Node Consensus Mechanism

In the hierarchical node consensus mechanism, blockchain nodes have three roles. Among them, one institution node acts as the master node to package the messages of the master network layer into blocks, and the other institution nodes act as the master network layer nodes to participate in the consensus process of the master network

layer with the master node. As the secondary network layer node, the terminal node is responsible for sending the local model parameters.

We structure the consensus mechanism into two layers. The secondary network layer transmits data to the master network layer nodes through an optimized PBFT consensus between the secondary and master network layer nodes, and the primary network layer selects the nodes that generate blocks through a reputation value-based master node selection mechanism. Since the two-layer consensus is asynchronous, the two-layer structure can improve the consensus efficiency.

3.3 The Global Model Update Process

The global model update process in this paper mainly consists of three steps.

Local model (secondary layer) submission stage: When terminals complete local model training, these terminal nodes submit model parameters and timestamps to the institution nodes in the area and send summary information to other terminal nodes in the secondary network layer. The institution node packages the model parameters submitted by terminal nodes received over a period of time into a collection. In order to achieve reliable model updates, institution is required to identify and delete low-quality local model updates using reliable shared data sets that are updated regularly to ensure that the accuracy of models participating in the aggregation reaches a certain threshold. After some time, the institution collects a set of qualified terminal local model updates and sends a response containing the collection information to each terminal node. After verification of terminal nodes, institution and each terminal node reached a consensus similar to PBFT.

Master network layer consensus stage: in the master node selection stage, the voting mechanism, heartbeat mechanism, and trust mechanism based on FL model contribution are combined. Each main network layer node broadcasts the model parameter set. The main node collects and sorts the message set received within a period of time, and reaches a consensus with each master network layer node. Finally, the master node saves all model parameters into the block, digitally signs the block, and broadcasts the block to other master network layer nodes for verification.

Global model training: Since each institution can directly retrieve the latest block data from its own blockchain ledger and calculate the weighted average of local model updates as the new global model, their final aggregation results are also the same. In FL, the latest global model is directly downloaded from the blockchain for all terminals participating in the next round of global aggregation as the initialized model for the next round of global model training iterations.

3.4 Primary Node Selection Mechanism Based on Trust Value

(1) Voting mechanism

All institution nodes record the term number of the leader node. When each node receives an election request, the term stored by the node is automatically incremented by one, and then a voting election is held. Within a term, each node can only cast one vote and is specified to vote for the first voting request node it receives, where a node can only vote for the node whose term number is greater than or equal

to its stored current term number. When a leader node finds that the term number of a new leader node is greater than its own term number, it will automatically become a child of the new leader node [9].

(2) **Heartbeat mechanism**

The leader node needs to periodically send a heartbeat signal to other nodes, which contains the current term number of the successfully elected leader node. Other nodes need to verify whether the current term number in the received heartbeat signal is greater than or equal to its own stored term number. If this condition is satisfied, for a period of time, then the node will identify this leader and submit a subsequent request message to the leader. In addition, when a node does not receive a heartbeat signal from the leader for a period of time or verifies that the corresponding information is incorrect, it will send an election request to other child nodes. Other child nodes hash this message and compare it with the last received summary information, and if it is verified to be correct, a new round of election is opened.

(3) **Trust mechanism**

An institution may obtain conflicting ratings for certain results, using weighted aggregation on these ratings to obtain an offset of the trust value. Offsets between -1 and $+1$ are positively correlated with the positive rating ratio of this message [10]. The trust offset of institution j in this region can be calculated as follows:

$$\tau_j^k = \frac{\kappa_1 \cdot m - \kappa_2 \cdot n}{m + n} \quad (9)$$

where m and n are the number of positive ratings and the number of negative ratings with weights κ_1 and κ_2 , respectively. κ_1 and κ_2 are calculated using the following equations.

$$\kappa_1 = \frac{F(m)}{F(m) + F(n)} \quad (10)$$

$$\kappa_2 = \frac{F(n)}{F(m) + F(n)} \quad (11)$$

where $F(\cdot)$ controls the sensitivity to the minority rating group. Finally, τ_j^k is placed into the offset set O_j of institution j and the sum of the absolute values of its trust bias is calculated as follows:

$$S_j = \min\left(\sum_{\tau_j^i \in O_j} |\tau_j^i|, S_{\max}\right) \quad (12)$$

where S_{\max} is the upper limit of S_j .

If a node is found to be malicious, or the model state provided by an institution node is found to be out of date, the node will be penalized by subtracting a warning value from the node's trust value and will be prevented from participating in the next round of voting.

Therefore, the trust value of a node for a limited time is defined as:

$$P = \min(\alpha \cdot \text{vote}_j + \beta \cdot S_j, P_{\max}) \quad (13)$$

where, $vote_j$ is the number of votes received by this node, S_j is the trust value of this node, and a and b are the relevant parameters set. Within a certain period of time, the node with the largest trust value is selected as the leader node among all nodes initiated by selection.

Algorithm 1: Transaction submission verification and Reputation mechanism.

Initialization:

A leader node and a set of child nodes

A series of transactions transList

Iteration:

1. Package Signature, timestamp, transList to node n_i
 2. Calculate Hash value h_i according to transaction t_i
 3. **For** each nineighborNode **do**
 4. Record h_i and corresponding nodes
 5. **End**
 6. **If** (Return value obtained $\neq h_i$) || (No heartbeat signal from the master node for a while) then
 7. **For** each ninneighborNode **do**
 8. reputationValue $P_{cur} \leftarrow \alpha \cdot vote_i + \beta \cdot S_i + \text{warningValue } \omega$
 9. **End**
 10. **If** reputationValue $P_{new} > P_{cur}$ **then**
 11. **For** each ninneighborNode **do**
 12. Select the new node
 13. $vote_i = \sum v_n$
 14. **End**
 15. **End**
 16. **End**
-

Output:

Reputation value of each node

leader node

(4) **Candidate block validation mechanism**

The process of candidate block validation includes the following steps:

Preparation stage: Secondary nodes generate verification results for block data. If they pass the verification, they send the verification results together with their own digital signature (summary information) to other secondary nodes for mutual confirmation. The preparation phase is completed when the miner receives at least 2f verification results as block data.

Submission phase: Each secondary node compares its own validation results with those received from other secondary nodes and replies with a confirmation message containing its digital signature to all other miners to indicate whether it agrees with the validation results. Over a period, if a secondary node receives

more than $2f + 1$ verified messages, secondary node sends the submission result and digital signature to the master node to prove that most miners agree on the authenticity of the block data.

Reply stage: When the master node receives $m + 1$ responses with the same results from different secondary nodes, that is, verify the message and check whether more than $2/3$ secondary nodes have reached the same conclusions about the block data. If so, the block data is recorded into the blockchain (Fig. 2).

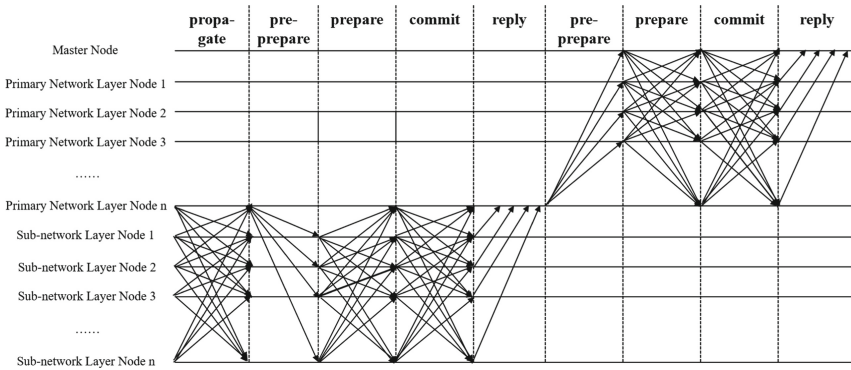


Fig. 2. Consensus process

4 Simulation Results and Analysis

We simulated the algorithm using TensorFlow 2.0 on a Python 3.8-based simulator. We conducted multiple experiments and averaged the experimental results. To verify the performance of the proposed node selection algorithm, the following algorithms are used for comparison.

- 1) Proposed: The algorithm selects a group of terminals as model training nodes and participates in global aggregation;
- 2) Local CNN: The FL mechanism is not used, and the model training is only performed on the local device;
- 3) FL-All: The algorithm selects all devices for global aggregation.

To verify the performance of the proposed consensus algorithm, the following algorithms are used for comparison.

- 1) Proposed: This consensus mechanism uses a two-layer structure to improve the efficiency of consensus. Combined with the trust degree, a voting mechanism is adopted to select the master node;
- 2) PBFT: This consensus mechanism can work in an asynchronous environment, but the communication complexity is too high and the scalability is relatively low;

- 3) Optimized PBFT [11]: Use the polling bookkeeping method of each node in the blockchain, and sets the number of consensus restarts to be no more than M rounds, which is used to improve the efficiency of consensus reaching among all nodes.

It can be seen from the figure that the proposed has the highest accuracy and converge faster. This is because the proposed algorithm aggregates high-quality models by selecting proper training nodes. FL-All is greatly affected by malicious nodes, so the global model has low accuracy. And, Local CNN has the lowest accuracy because it obtains the locally optimal solutions.

The following figure depicts that the consensus delay of Proposed is lower than the scheme of PBFT and Optimized PBFT. The reason for this is that the propagation delay of this scheme is lower than that of PBFT and Optimized PBFT because of the closer distance between the terminals and institution nodes in the same region. In addition, although this scheme improves the accuracy of the model by modifying the reputation value, the consensus latency of this scheme and the scheme of Optimized PBFT gradually approaches as the number of blockchain nodes increases. However, since Optimized PBFT uses the polling bookkeeping method, its communication resource consumption is smaller than the master node selection method used in this scheme; therefore, the accuracy rate is inversely proportional to the communication resource consumption, and more communication resources are required to achieve a higher accuracy rate. With the increase in the number of nodes running the blockchain, the communication load and computational overhead need to be increased (Fig. 3, Fig. 4 and Fig. 5).

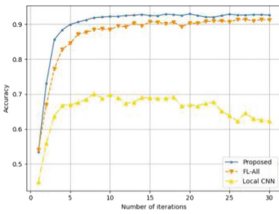


Fig. 3. Accuracy

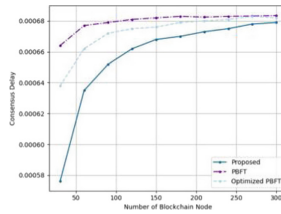


Fig. 4. Consensus delay

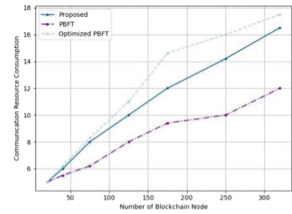


Fig. 5. Communication resource consumption

5 Conclusion

In order to solve the data security problem in the model training process of intelligent services, we propose a blockchain empowered FL framework in genetic data engineering. We introduce FL to protect data privacy and reduce communication costs, design node selection algorithm, and use blockchain to store model information to further ensure data security. In addition, we divide blockchain nodes into primary network layer nodes which are responsible for generating blocks and secondary network layer nodes which are responsible for uploading models. Then, we design a two-layer consensus algorithm based on PBFT to improve consensus efficiency. Simulation results show that the proposed algorithms significantly improve the accuracy of the global model, consensus efficiency, and communication resource consumption.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (62071070), and the Major Science and Technology Special Project of Science and Technology Department of Yunnan Province (202002AB080001-8). Yuxuan Zhong is the corresponding author with email: xxuan@bupt.edu.cn.

References

1. Wu, W., He, L., Lin, W., Mao, R.: Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems. *IEEE Trans. Parallel Distrib. Syst.* **32**(7), 1539–1551 (2021)
2. Liu, H., et al.: Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing. *IEEE Trans. Veh. Technol.* **70**(6), 6073–6084 (2021)
3. Korkmaz, C., et al.: Decentralized federated machine learning via blockchain. In: 2020 Second International Conference on Blockchain Computing and Applications (BCCA), pp. 140–146. Antalya, Turkey (2020)
4. Ramanan, P., Nakayama, K.: BAFFLE: Blockchain based aggregator free federated learning. In: 2020 IEEE International Conference on Blockchain (Blockchain), pp. 72–81. Rhodes (2020)
5. Majeed, U., Hong, C.S.: FLchain: Federated Learning via MEC-enabled Blockchain Network. In: 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 1–4. Matsue, Japan (2019)
6. Shrestha, R., Bajracharya, R., Shrestha, A., Nam, S.Y.: A new-type of blockchain for secure message exchange in VANET. *Digit. Commun. Netw.* **6**, 177–186 (2019)
7. Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y.: Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE Trans. Industr. Inf.* **16**(6), 4177–4186 (2020)
8. Guo, S., Dai, Y., Guo, S., Qiu, X., Qi, F.: Blockchain meets edge computing: stackelberg game and double auction based task offloading for mobile blockchain. *IEEE Trans. Veh. Technol.* **69**(5), 5549–5561 (2020)
9. Li, Y., Qiao, L., Lv, Z.: An optimized byzantine fault tolerance algorithm for consortium blockchain. *Peer-to-Peer Netw. Appl.* **14**(5), 2826–2839 (2021)
10. Yang, Z., Yang, K., Lei, L., Zheng, K., Leung, V.C.M.: Blockchain-based decentralized trust management in vehicular networks. *IEEE Internet Things J.* **6**(2), 1495–1505 (2019)
11. Guo, S., Xing, H., Zhou, Z., Wang, X., Qi, F., Gao, L.: Trust access authentication in vehicular network based on Blockchain. *China Commun.* **16**(6), 18–30 (2019). <https://doi.org/10.23919/JCC.2019.06.002>