Federated Parking Flow Prediction Method Based on Blockchain and IPFS

Xuesen Zong[1(✉)], Zhiqiang Hu[1], Xiaoyun Xiong[1], Peng Li[2], and Jinlong Wang[1]

[1] Qingdao University of Science and Technology, Qingdao, China
1119698476@qq.com
[2] Qingdao Yilian Information Technology Co. LTD., Qingdao, China

Abstract. Aiming at the problem of privacy security of parking data and low generalization performance of parking flow prediction model, a federated parking flow prediction method based on blockchain and IPFS is proposed. In this method, blockchain and IPFS are applied to the federated learning frame-work. Under the condition of ensuring the privacy and security of parking data, blockchain is used to replace the central server of federated learning to aggregate multi-party local models. Through blockchain and IPFS, the model data in the training stage of the parking flow prediction model are stored and synchronized quickly, which improves the generalization performance of the model and further improves the training efficiency of the model. In addition, in order to improve the participation enthusiasm of all participants, an incentive mechanism based on data volume contribution and model performance improvement contribution is designed. The experimental results show that the method can improve the generalization performance of the model and improve the training efficiency of the parking flow prediction model, and provide a reasonable reward allocation.

Keywords: LSTM · Federated learning · Blockchain · IPFS · Parking flow prediction · Incentive mechanism

1 Introduction

Parking flow prediction plays an important role in intelligent parking management. For office parking lot, it needs accurate and real-time parking flow prediction to analyze parking planning and formulate reasonable parking resource allocation strategy [1, 2]. However, for different office parking lots, due to the difference of parking flow in the same period, the generalization performance of the parking flow prediction model in some office parking lots is low. How to scientifically improve the generalization performance of the parking flow prediction model to provide timely and effective reference is of great significance.

Statistical learning method is a traditional solution [3, 4], which is mainly divided into Auto-regressive Integrated Moving Average (ARIMA) [3] model and Kalman filter model [4]. However, due to the change of parking flow shows some nonlinear characteristics, and changes rapidly with time, the fitting degree of this method for parking

Z. Lv and H. Song (Eds.): INTETAIN 2021, LNICST 429, pp. 257–275, 2022.
https://doi.org/10.1007/978-3-030-99188-3_16

flow data is low. Then, the prediction method of parking flow based on machine learning models such as K-Nearest Neighbors (KNN) [5], Support Vector Regression (SVR) [6] and Long Short-Term Memory (LSTM) [7] appeared. By training the model with parking data, the parking flow data were better fitted. However, with the emphasis on user privacy and data security, it was difficult to achieve the above method to train the parking data set model. In order to solve this problem, Google proposed a new privacy protection technology—Federated Learning in 2016 [8]. It was a collaborative method of distributed machine learning, which made the original data remain in local devices and could maintain the integrity of users' data privacy. Therefore, it was widely used in many fields. For example, federated learning had been applied to the financial sector, with examples of microcredit risk management and anti-money laundering [9]. Nevertheless, federated learning relied on a central server to aggregate the local model. Once the central server fails, it interrupted the training of the model and cause a single point of failure. In response to this problem, some scholars proposed to replace the central server with blockchain [10–13], stored and updated the federated model by using blockchain, and avoided the problem of single point failure by decentralizing. However, the above literature did not consider the low training efficiency of the federated model. When the parameters of the parking flow prediction model were large, there would be excessive demand for blockchain storage space in the direct chain storage of model data, and the storage and query of model data were slow, resulting in a long training time of the model.

In addition, federated learning methods lack incentives to improve the enthusiasm for participants. In the process of model training, not all participants contribute data actively. If there was no feasible incentive mechanism, participants with high-quality data might not participate actively, and ultimately affect the performance of the federated model. Therefore, it was necessary to add incentives to federated learning methods to encourage participants to actively participate in training. Kim et al. [11] proposed a reward based on data volume of participants, the greater data volume of participants, the more reward. Whereas, the greater data volume did not mean the greater the improvement in model performance, and for participants with the same amount of data but different data quality, it was impossible to distinguish the contribution of the two to the federated model. Therefore, this method had certain one-sidedness. Other researchers [14, 15] proposed to evaluate the performance improvement on the federated model before and after the participants' data training by setting a public validation set. However, for the data with strong privacy, it could not provide a public validation set for this method. In addition, there was also an one-sidedness to evaluate the contribution of the participants only from the model performance and ignore the amount of data.

In view of the low efficiency of model training and one-sidedness of incentive methods in current methods, this paper designs a federated parking flow prediction meth-od based on blockchain and IPFS. This method constructs a parking flow prediction model based on LSTM network, trains the parking flow prediction model under the federated learning framework to ensure the privacy of parking data, and uses blockchain to replace the central server to aggregate the local model to realize decentralization under the federated learning framework to prevent single-point faults. At the same time, based on the tamper resistance of blockchain, it further ensures the safety of model data.

The contributions of this paper are itemized as follows.

(1) Improving the efficiency of model training: By combining blockchain and IPFS, it realizes the on-chain and off-chain storage of model data, improves the storage and query efficiency of model data, so as to improve the training efficiency of the federated parking flow prediction model.
(2) Providing a contribution incentive mechanism: A contribution incentive mechanism is designed to comprehensively evaluate the data volume of participants and the performance improvement on the federal model without setting a public validation set, so as to provide reasonable incentive allocation and improve the enthusiasm of participants.

2 Correlative Knowledge

2.1 LSTM Neural Network

Recurrent Neural Network (RNN) abandons the full connection mode of the general fully connected neural network of the hidden layer, introduces the concept of time series and adopts the 'recursive connection' mechanism to retain the time series characteristics of the sequence, and retains the previous input information on the network [14]. Because there are enough hidden layer neurons in RNN, it can accurately fit the predicted time series. However, the traditional RNN has the problems of gradient disappearance and gradient explosion during training. In this regard, Hochreiter and Schmidhuber proposed the LSTM model [17], which is a storage unit composed of several gates. The gate can control the transmission of information about the sequence, obtain the long-term dependence on time series, effectively avoid the problem of gradient disappearance, and achieve good results in the prediction of time series data.

A typical LSTM model is controlled and protected by forgetting gate f, input gate i and output gate o (see Fig. 1). The forgetting gate f is used to discard unimportant information in the previous time step, the input gate i selects useful information from the input, and the output gate o controls the output of the current LSTM model. The state update of each gate is as follows:

$$f_t = \sigma\left(w_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(w_i \cdot \left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$o_t = \sigma\left(w_o \cdot \left[h_{t-1}, x_t\right] + b_o\right) \tag{3}$$

where h_t is the hidden state of time step t, x_t is the input of time step t, w_f, w_i and w_o are the weight values for each gate, b_f, b_i and b_o are the offset of each gate, σ is sigmoid activation function. The cell state C_t is constantly updated over time. Firstly, $\tilde{C}_t$ is obtained from Eq. (4), and C_t is updated according to Eq. (5). w_c and b_c represent the weight value and offset respectively, tanh is activation function. The output value h_t is controlled by the output gate unit and the cell state activated with tanh, as shown in Eq. (6). In short, LSTM uses gate units to enhance the memory of the recurrent neural

network, reduce the amount of data carried by the intermediate process, and has better prediction results than RNN [18].

$$\tilde{C}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

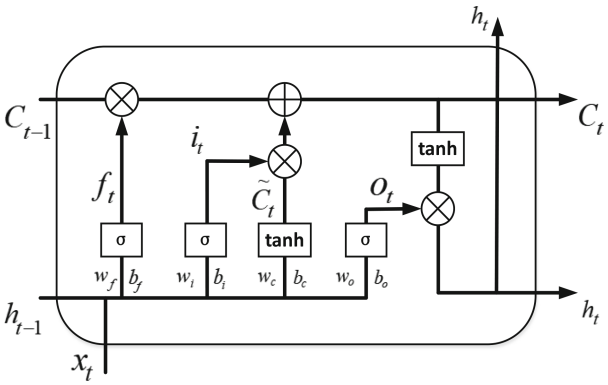


Fig. 1. LSTM network structure.

2.2 Federated Learning

Federated learning is essentially a distributed machine learning technology that can effectively reduce the risk of data privacy caused by source data in traditional ma-chine learning sets [9]. Its working principle mainly includes multiple participants and central servers (see Fig. 2). Participants (mobile phones, tablets, IoT devices, etc.) jointly train the model under the coordination of the central server. Participants use local data to train the local model. The central server uses the model average to aggregate the local model to obtain the global model, and finally obtains a model that is close to the centralized machine learning effect after multiple iterations.

An iterative process of federated learning is as follows:

① Each participant downloads global model parameters from the central server.
② Participants use local data to train local models.
③ Each participant uploads local model parameters to the central server.
④ The central server receives the model data of each participant and obtains the global model parameters by weighted aggregation.

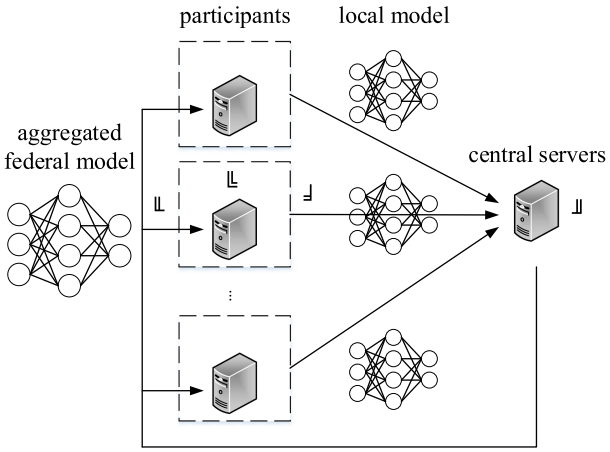


Fig. 2. Federated learning framework.

2.3 Blockchain

As the core technology of the digital cryptocurrency system represented by bitcoin, blockchain has the characteristics of traceability, transparency, tamper-proof and decentralization [19]. It uses the hash algorithm to calculate the head of each data block to obtain a hash value, connects blocks to form a data chain. In the data block, the transaction information in the current time is recorded and saved by Merk-le tree information, each data block is composed of block head and block (see Fig. 3).

The core technology of blockchain is distributed ledger, asymmetric encryption, consensus mechanism and smart contract. Among them, distributed accounts can be completed by multiple nodes in different places, and each node records a complete account. Asymmetric encryption makes data accessible only after authorized, which ensures the security and privacy of data. Consensus mechanism is to solve all ac-counting nodes how to reach a consensus, identify the validity of a record, but also a means to prevent tampering. smart contract is based on trusted and non-tampering data, which can automatically execute some predefined rules and clauses. Blockchain can be divided into three types: public chain, private chain and alliance chain. Among them, public chain is completely decentralized, and anyone can add data to public chain. Private chain is commonly used within the company or organization, and all operating permissions are controlled by the company or organization, with a high degree of centralization. The alliance chain is managed by several organizations or institutions. Each organization or institution controls a limited number of nodes, and the membership of members needs permission from the organization or institution.

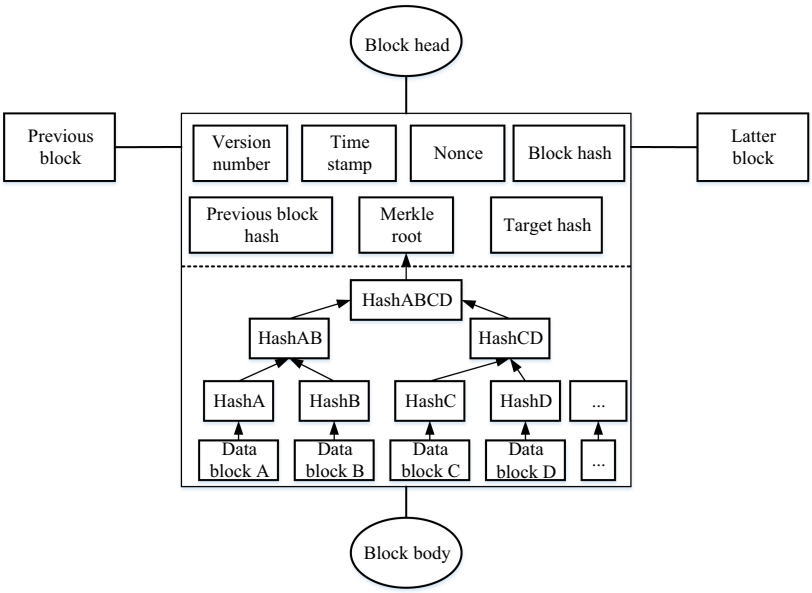


Fig. 3. Data block structure.

2.4 IPFS

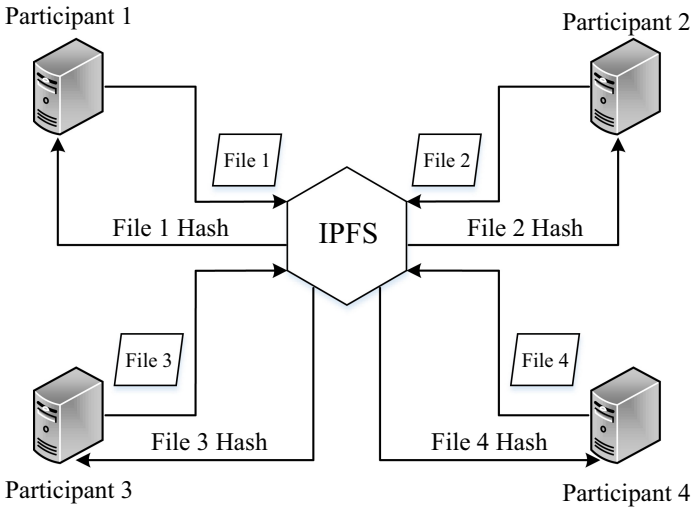


Fig. 4. Participants interact with IPFS.

IPFS provides distributed file storage system and content addressable technology to access stored files, which is convenient to connect with P2P network [20]. All peer points in IPFS computing network can access the unique Hash of files. Hash is modified every time the file is updated. IPFS is also known as version control system, which ensures the security, reliability and scalability of existing file storage and sharing systems [21]. The same transaction has the same hash in IPFS, which ensures the originality of the content. IPFS hash is spread to all peers, which also ensures the consistency between peers. Star file system provides high throughput content addressing block storage model to ensure transaction security. When multiple participants interact with the IPFS network, each participant uploads files to the IPFS network for storage, and then obtains the returned file Hash (see Fig. 4).

3 Federated Parking Flow Prediction Method Based on Blockchain and IPFS

The method in this paper is established under the framework of federated learning combined with blockchain and IPFS network. It is considered that multiple participants are required to train the federated parking flow prediction model at the same time. In order to prevent the leakage of parking data and the abuse of information, strict licensing management is needed. Therefore, the blockchain type in this article uses the alliance chain. In addition, the participants involved in the model training process need to register in the alliance chain, and the information of the registered participants will be sent to the participants nodes assigned to them. Through the participant registration contract, the elliptic curve encryption algorithm is used to generate the unique public-private key pair for the participants, and the participants registration information is bound to the public-private key pair. Finally, the participants node returns the public-private key pair to the corresponding participant.

3.1 Training of Federated Parking Flow Prediction Model

The overall framework of this method is mainly composed of local participants, blockchain networks and IPFS networks (see Fig. 5). Local participants are responsible for training local models. The trained local models are uploaded and stored in the IPFS network, and the returned local model Hash is stored in the blockchain network through the participant node calling smart contracts. In the blockchain network, the local model parameters are aggregated through smart contracts, and the aggregated global model parameters are stored in the IPFS network. The Hash of the returned global model parameters is stored in the blockchain network through smart contracts, and sent to the local participants through smart contracts. Then, the global model is obtained from the IPFS network. Finally, the local model is updated to the global model for the next round of model training. Next, the key components and processes of this method are described in detail.

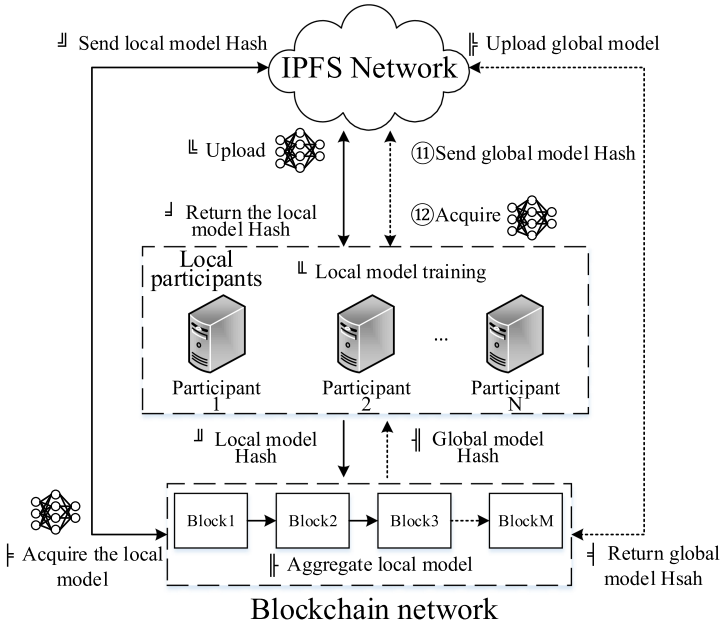


Fig. 5. Federated parking flow prediction method framework based on blockchain and IPFS.

1) Parking flow prediction model
In this paper, the prediction model of parking flow is constructed based on LSTM network, which is divided into two LSTM layers and a full connection layer. The cur-rent parking flow is predicted by combining the previous parking flow data. Due to the working mode of the office building personnel takes a week as a cycle, and its parking behavior also exists in a week as a cycle, with a total of 168 periods. There-fore, the input layer dimension of LSTM is set to 168, the hidden layer unit is

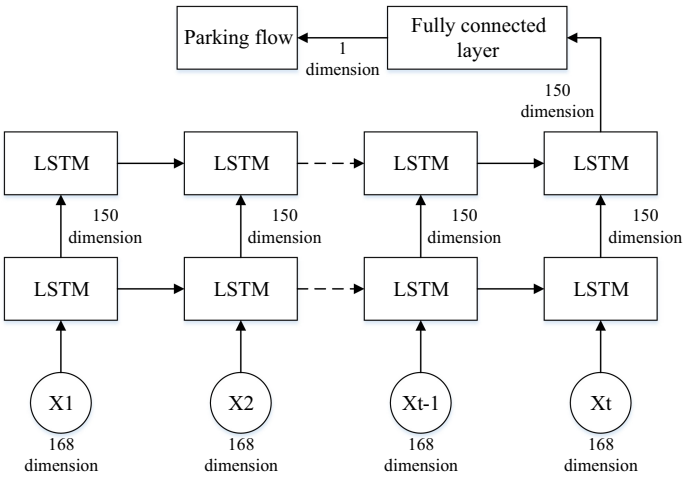


Fig. 6. Structure of parking flow prediction model.

set to 150, and the full connection layer is calculated to generate a one-dimensional parking flow data output with 150-dimensional data as input, where Xt is the model input value at t time (see Fig. 6).

2) Smart contract design
A smart contract is a self-executing contract that defines rules for negotiation, verifies the execution of rules and executes protocols using formal code. The method of this paper sets up the registration contracts of participants, upload contracts, download contracts, send contracts and aggregation contracts. The participant registration contract receives the participant's user name, user password and encrypted identity card number, generates a unique public-private key pair for it using the elliptic curve encryption algorithm, and binds the participant registration information to the public-private key. The encryption of ID card number is to prevent the information leakage of participants caused by hacker intrusion. The upload contract receives the user's name of the participants and the Hash information of the local model, and stores it on the chain in the form of key value pairs after encryption using the elliptic curve encryption algorithm. The download contract to receive the participant user name, using elliptic curve encryption algorithm to decrypt the Hash model information. The send contract is responsible for sending Hash for global model parameters to all participants. The aggregation contract sets the weighted average rule of model parameters, and automatically monitors the upload state of the Hash information of the local model of each participant. When each participant uploads, the local model aggregation is automatically performed, and the aggregated global model is stored in IPFS, and the Hash information of the global model is stored in the blockchain through the upload contract. The weighted average formula for model parameters is shown in formula (7), where w is a global model parameter, k is the number of local participants, n is the total amount of data for all local participants, n_i is the amount of data for local participant i, and w_i is the local model parameter for local participant i.

$$w = \sum_{i=1}^{k} \frac{n_i}{n} w_i \tag{7}$$

3) Model training process
The local model and global model training process of this method are described below (see Fig. 7). Firstly, the upload state of each local model of the local participant is defined as *model_state*, the state of the local participant obtaining the global model is *model_get*, and the update state of the global model is *model_aggregation*, and each state is initialized as 0. Then, each participant uses local data to train the local model, uploads the trained local model to IPFS, and uploads the hash value of the returned local model to the blockchain network with the upload state *model_state* of the local model as 1 through the participant node. The aggregation contract in the blockchain network monitors and queries the upload status of the local model of each participant. When the upload status query of the local model of all participants is 1, the Hash value of the local model of each participant is obtained, and further the local model of each participant is obtained by using the Hash value through the IPFS network, and the upload status *model_state* of the local model of each participant is

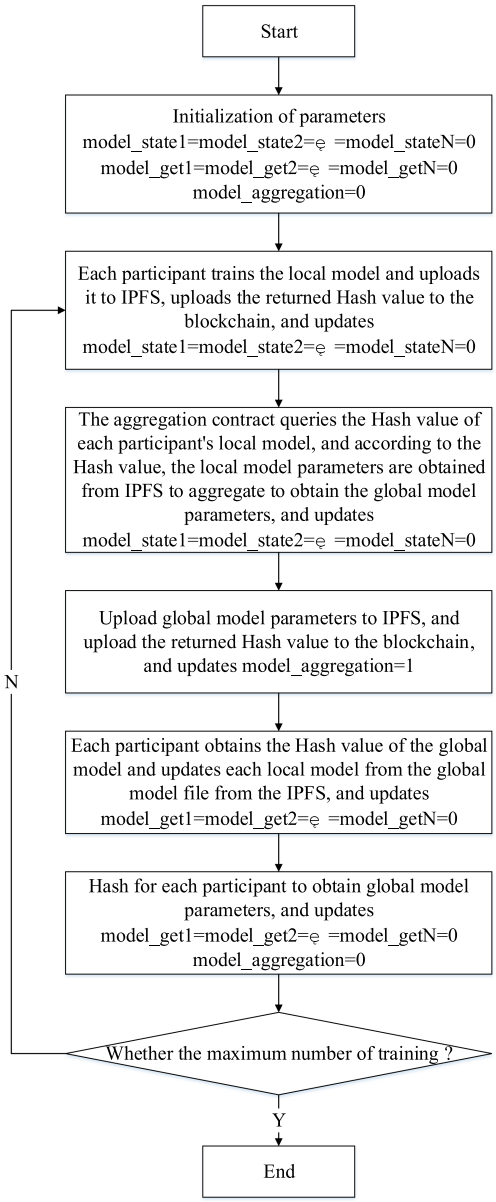


Fig. 7. Model training process.

modified to be 0. Then, the local model of all participants is aggregated according to the weighted aggregation rule set in the aggregation contract, so as to obtain the global model and upload it to the IPFS network. The Hash value of the returned global model is uploaded to the blockchain network, and then the global model is modified to update the state *model_aggregation* as 1. Each participant uploads the

Hash value of the local model, and then queries the update state *model_aggregation* of the global model through each participant node. When the *model_aggregation* is 1, each participant obtains the Hash of the global model from the blockchain network through the participant node, and further obtains the global model from the IPFS network. The state *model_get* of the global model obtained by the local participant is modified to 1. When all local participants obtain the global model, the update state *model_aggregation* of the global model is modified to 0, and the state *model_get* of the global model obtained by the local participant is 0. Then, each local participant updates the global model to the local model for the next round of the local model. When the maximum number of training rounds is reached, the model training is completed.

3.2 Incentive Mechanism

The incentive mechanism designed in this paper is divided into two parts (see Fig. 8). One part is to evaluate the contribution of each participant's data volume, and the other part is to evaluate the contribution of each participant's model performance improvement. Finally, the two parts are integrated to obtain the total contribution of each participant, calculate the allocation proportion of each participant and store it in the blockchain. The following two parts are introduced.

Firstly, the calculation method of data volume contribution is introduced. Let D_i^n denote the number of local training sets of participant i, then the data contribution D_i of each participant can be calculated by formula (8), where N is the number of participants.

$$D_i = \frac{D_i^n}{\sum_{j=1}^{N} D_j^n} \tag{8}$$

Next, the calculation method of model performance improvement contribution is introduced. Firstly, $M_{\{i\}}$ is defined as the sub-federated model trained by the data of other participants except participants, and M is the main-federated model trained by all participants. Each participant uses its local validation set to evaluate the accuracy of the main-federated model and the sub-federated model, respectively. The model accuracy calculation formula is as follows:

$$R = 1 - \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)} \tag{9}$$

where R is the accuracy of the model, m is the number of validation sets, y is the true value of the validation set, and $\hat{y}$ is the predicted value of the model. The model accuracy of each federated model $M_{\{i\}}$ and the main-federated model M in the local validation set of each participant j is denoted as $R_j^{M_{\{i\}}}$ and R_j^M, respectively. Let S_i denote the average model accuracy of sub-federated model $M_{\{i\}}$ and main-federated model M in

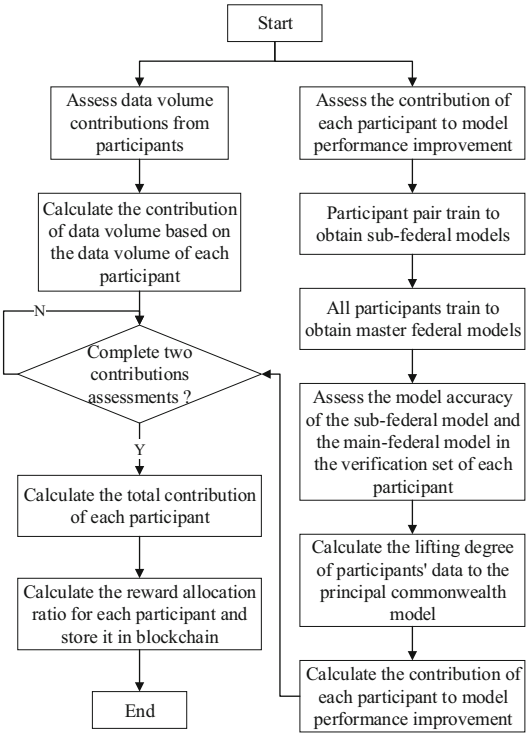


Fig. 8. Incentive Mechanism Process.

the verification set of each participant, as shown in Eq. (10).

$$S_i = \sum_{j=1}^{N} \left(R_j^M - R_j^{M_{\{i\}}} \right) \tag{10}$$

We believe that S_i can reflect the degree of performance improvement of the federated model. The larger S_i is, the higher the improvement of the data of the participant i on the federated model is, and the greater the contribution of the participant to the performance improvement of the model is. On the contrary, if S_i is smaller and close to 0, it is believed that the federated model has a good prediction performance for the data of the participant, then the contribution of the participant to the federated model is small. If S_i is negative, this shows that the data of the participant will reduce the performance of the federated model, then consider not providing incentives or rejecting its participation. In order to intuitively reflect the improvement of the performance of the federated model by the participant data, the contribution of each participant to the model performance improvement can be defined by the following calculation:

$$C_i = \frac{S_i}{\sum_{j=1}^{N} S_j} \tag{11}$$

The total contribution η_i of the participant i is then obtained by formula (12), where λ and $\mu \in [0, 1]$ is the weight of the data volume contribution and the model performance enhancement contribution, respectively, and $\lambda + \mu = 1$.

$$\eta_i = \lambda D_i + \mu C_i \tag{12}$$

According to the contribution value of each participant, the distribution proportion of the reward value G_i of each participant in the federated model training can be calculated as:

$$G_i = \frac{\eta_i}{\sum_{j=1}^{N} \eta_j} \tag{13}$$

4 Experimental Results and Analysis

4.1 Experimental Parameter Settings

In the method of this paper, the local model training is realized based on Python 3.8 and Pytorch deep learning framework. The blockchain adopts the Hyperledge Fabric 2.0 framework, and relies on the mature security mechanism of Fabric to ensure the security mechanism of the method. The consensus mechanism adopts the Raft of Fabric, and the smart contract is realized by Golang programming. The experimental device is a notebook with 16 GB memory, and 6-core Intel Core i7 processor. It runs three virtual machines with 4 cores and 4 GB of memory, each virtual machine with Ubuntu 18.04 system. The experimental data are selected from the parking data collected from three real parking lots in Qingdao, Shandong Province, China from April to September 2020, and the specific information is shown in Table 1. For example, parking lot 2 has 400 parking spaces, with full-day traffic between 650 and 800 on working days and between 100 and 250 on rest days.

4.2 Experimental Data Processing

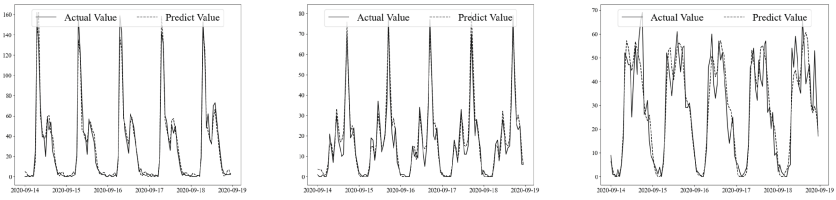
Check and clear abnormal data for three parking data to ensure the effectiveness of all parking data. Subsequently, the parking flow in each period is further counted for each parking data, and the 0–1 standardization is used to normalize the parking flow data. In addition, according to the time sequence, the parking data from April to August are used as the historical dataset, and the parking data in September are used as the test dataset.

Table 1. Experimental data information.

Parking lot	Number of parking spaces	Parking flow on working days	Parking flow on rest day
1	450	800–900	100–300
2	400	650–800	100–250
3	300	400–550	50–150

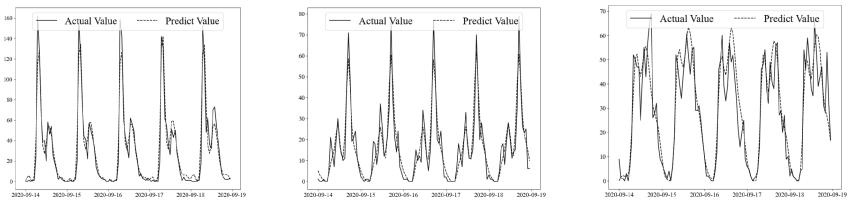
4.3 Experimental Result Analysis

This experiment is divided into two parts: model training and model contribution evaluation. The model training part uses the historical dataset of each participant to train the parking flow prediction model, and obtains the global parking flow prediction model. According to the trained global parking flow prediction model and the test dataset of each participant, the root mean square error of the prediction results is calculated respectively, and used as the evaluation standard of the model performance. The model contribution evaluation part evaluates the model contribution of each participant, calculates the reward allocation value of each participant according to the incentive mechanism, and stores it in the blockchain for participants to query.



(a) Participant 1 (b) Participant 2 (c) Participant 3

Fig. 9. Model performance after centralized data training based on LSTM model.



(a) Participant 1 (b) Participant 2 (c) Participant 3

Fig. 10. Performance of federated parking flow prediction model based on blockchain and IPFS.

In order to compare the performance of the federated parking flow prediction model, the method proposed in this paper is compared with the method based on LSTM centralized data training [8]. The performance of the model trained based on LSTM centralized data on the three participants' parking data validation sets (see Fig. 9). The performance of the federated learning model trained by the method proposed in this paper on the three participants' parking data validation sets (see Fig. 10). Table 2 shows the root mean square errors of the above two methods on the three participants' parking test datasets. Through observation and comparison, the performance of the federated model is slightly lower than that of the model based on LSTM centralized data training, but it meets the security of data privacy and meets the actual needs.

In order to measure the model training efficiency of the method in this paper, a total of 15 tests were conducted in the same environment with the federated learning method

Table 2. Model root mean square error statistics.

participants	Centralized data training method based on LSTM model	Federated parking flow prediction method based on blockchain and IPFS
Participant 1	5.442978220296002	6.621649297039326
Participant 2	3.85194816705429	4.176416201840635
Participant 3	2.052104307244879	2.391470359073033

based on blockchain. In each test, a total of 10 rounds of federated learning model training were conducted, and 50 rounds of local model training were conducted by each participant in each round. Statistics single model training time, single model upload time, single model query time, federated model training time as shown in Table 3.

Table 3. Average training time of model.

Test variables	Federated learning method based on blockchain and IPFS	Federated learning method based on blockchain
Single local model training time	10.95 s	10.6 s
Single local model upload time	0.97 s	2.71 s
Single local model query time	2.04 s	2.45 s
Global model aggregation time	12.04 s	14.82 s
Federated model training time	7 m 14 s	8 m 33 s

In addition, in order to reflect the advantages of using IPFS, the model data size is expanded by 2 to 4 times, and the model data upload time based on blockchain federated learning method is compared, as shown in Table 4. Among them, the model upload time of the federated learning method based on blockchain and IPFS is stable at 0.97 s, which does not change with the increase of model data size. The model upload time of the federated learning method based on blockchain increases with the increase of model size, which further reflects the effectiveness of this method to improve the efficiency of model training.

Table 4. Performance comparison based on IPFS.

Model data size	Federated learning method based on blockchain and IPFS	Federated learning method based on blockchain
5.05M	0.97 s	2.71 s
10.10M	0.97 s	4.04 s
15.15M	0.97 s	5.77 s
20.20M	0.97 s	9.08 s

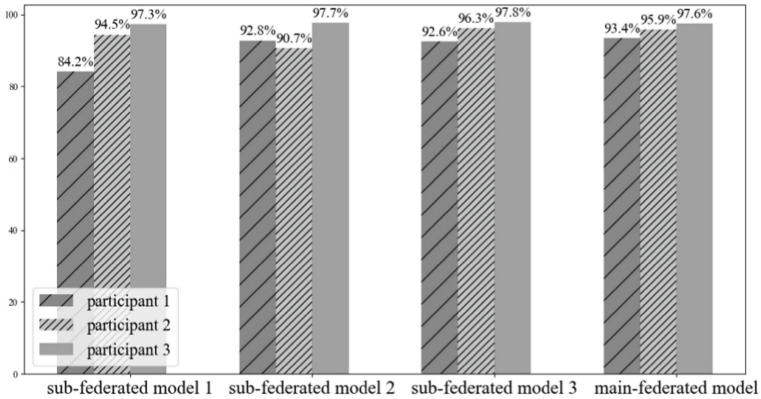


Fig. 11. Accuracy of sub-federated model and main-federated model in validation sets of participants.

Calculating the model accuracy of sub-federated models $M_{\{1\}}, M_{\{2\}}, M_{\{3\}}$ and main-federated model M on each participant validation set based on incentive mechanism for three participants (see Fig. 11). It can be seen from the graph that for the participant 1, the model accuracy of the sub-federated model $M_{\{1\}}$ and the main-federated mode M on its validation set is quite different, which shows that the participant 1 should make a greater contribution to the model performance of the main federated model M, and for the participant 2, the model accuracy of the sub-federated model $M_{\{2\}}$ and the main federated model M on its validation set is also quite different, which also shows that the participant 2 has a greater contribution to the model performance of the main federated model M, and for the participant 3, the model accuracy of the sub-federated model $M_{\{3\}}$ and the main federated model M on its validation set is basically the same, which shows that the participant 3 has little improvement in the model performance of the main federated model, so the model performance contribution should be small.

In the experiment, when the data volume of each participant is the same, by setting the contribution weight of data volume and the contribution weight of model performance improvement, the reward allocation proportion of each participant is obtained according to the formula (12) and formula (13) (see Fig. 12). Among them, the reward allocation proportion of participant 1 is the largest, accounting for 47.55%, which is because the accuracy of the federated model is the largest, followed by the reward allocation proportion of participant 2, accounting for 36.79%, and the reward allocation proportion of participant 3 is the smallest, which is because the accuracy of the federated model is the smallest.

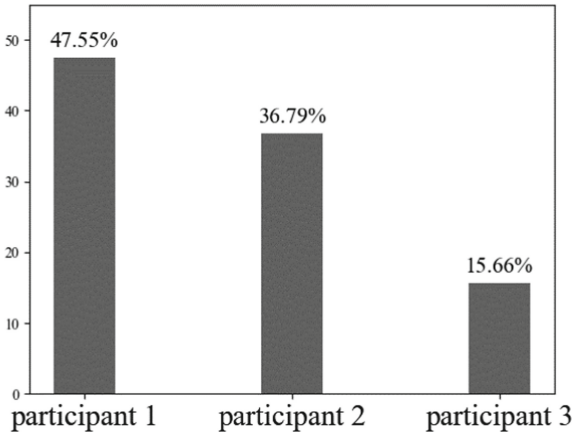


Fig. 12. Proportion of reward value allocation among participants.

Finally, considering the scenario that blockchain will generate concurrent transactions among multi-participant nodes during model training, a generation time of block test is performed to determine whether it can meet participants' daily applications. The test environment is as follows: Each participant node continuously sends 1000 model data to test whether the average generation time of block meets the requirements, that is, less than the set maximum generation block waiting time. In the experiment, the maximum generation block waiting time is set to be 1 s according to the generation block routine of Fabric alliance chain, and three participants are set, so the maximum block transaction number is set to be 3. A total of 15 tests were conducted, and the average generation block time was calculated after each test (see Fig. 13). It can be seen from the figure that the generation block time showed a gentle upward trend at the beginning, reaching the maximum generation block time in the 10th test, and then fluctuated around the average generation block time. The reason for this phenomenon is that with the increase in the number of tests, more memory space, and the rate of data processing is reduced. After 15 tests, the average generation block time of processing model parameters is 0.541 s, and the maximum generation block time is 0.643 s, which is far less than the set maximum generation block waiting time of 1s. Therefore, the Fabric alliance chain selected in this paper can effectively deal with 3000 transaction requests submitted by 3 participating nodes, which can meet the normal application requirements.

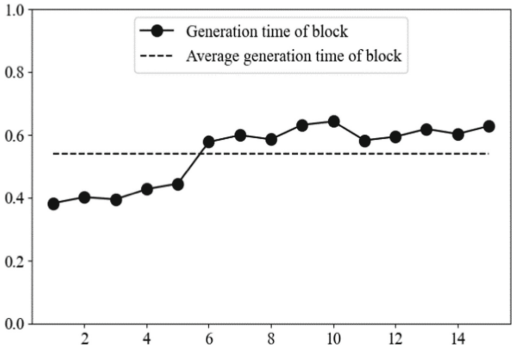


Fig. 13. Generation time of block.

5 Conclusion

This paper proposes a federated parking flow prediction method based on blockchain and IPFS, and uses federated learning to solve the privacy and security of parking data. Blockchain and IPFS are used to store and aggregate multi-party local parking flow prediction model data to further improve data security and training efficiency. At the same time, an incentive mechanism based on the contribution of each participant to the federated model is designed to promote the participation enthusiasm of the participants. Finally, relevant experiments are designed to verify the proposed method. The experimental results show that the proposed method improves the training efficiency of the federated parking flow prediction model and can reasonably calculate the contribution rewards of each participant. Although the performance of the federated parking flow prediction model is slightly lower than that of the traditional method, the data security is improved, which provides some reference for the training of the parking flow prediction model.

References

1. Wang, F.Y.: Parallel control and management for intelligent transportation systems: concepts, architectures, and applications. IEEE Trans. Intell. Transp. Syst. **11**(3), 630–638 (2010)
2. Wang, Y., Guo, L.Y., Cheng, X.: Short-term traffic flow forecasting optimization method based on deep learning. Comput. Eng. Appl. **56**(16), 211–217 (2020)
3. Williams, B.M., Durvasula, P.K., Brown, D.E.: Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. Transp. Res. Rec. **1644**(1), 132–141 (1998)
4. Okutani, I., Stephanedes, Y.J.: Dynamic prediction of traffic volume through kalman filer theory. Transp. Res. Part B Methodol. **18**(1), 1–11 (1984)
5. Zhang, T., Chen, X., Xie, M. P.: Short term traffic prediction method based on K nearest neighbor nonparametric regression. Syst. Eng. Theory Pract. **1**(2), 376–384 (2010)
6. Castroneto, M., Jeong, Y.S., Jeong, M.K.: Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst. Appl. **36**(3), 6164–6173 (2009)
7. Li, X.D., Cen, Y.F., Cen, G.: Prediction of short-term available parking space using LSTM model. In: 2019 14th International Conference on Computer Science & Education (ICCSE), Toronto, Ontario, Canada, vol. 20, pp. 631–635 (2019)

8. Mcmahan, H.B., Moore, E., Ramage, D.: Communication-efficient learning of deep networks from decentralized data. Artif. Intell. Stat. **1**(28), 1273–1282 (2017)
9. Yang, Q., Liu, Y., Chen, T.: Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) **10**(2), 1–19 (2019)
10. Ma, C., Li, J., Ding, M.: When federated learning meets blockchain: a new distributed learning paradigm. IEEE Internet Things J. **20**(9), 1–7 (2020)
11. Kim, H., Park, J., Bennis, M.: Blockchained on-device federated learning. IEEE Commun. Lett. **24**(6), 1279–1283 (2020)
12. Qu, Y., Pokhrel, S.R., Garg, S.: A blockchained federated learning framework for cognitive computing in industry 4.0 networks. IEEE Trans. Ind. Inform. **17**(4), 2964–2973 (2021)
13. Luo, C.Y., Chen, X.B., Ma, C.D.: Online federal incremental learning algorithm for blockchain. J. Comput. Appl. **41**(2), 363–371 (2021)
14. Kang, J.W., Xiong, Z.H., Niyato, D.: Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory. IEEE Internet Things J. **6**(6), 10700–10714 (2019)
15. Li, Y., Chen, C., Liu, N.: A blockchain-based decentralized federated learning framework with committee consensus. IEEE Netw. **35**(1), 234–241 (2021)
16. Phan, M.C., Hagan, M.T.: Error surface of recurrent neural networks. IEEE Trans. Neural Netw. Learn. Syst. **24**(11), 1709–1721 (2013)
17. Hochreiter, S., Schmidhuber, H.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
18. Zhao, R., Wang, D., Yan, R.: Machine health monitoring using local feature-based gated recurrent unit networks. IEEE Trans. Ind. Electron. **65**(2), 1539–1548 (2018)
19. Ma, L., Pei, Q., Qu, Y., Fan, K., Lai, X.: Decentralized privacy-preserving reputation management for mobile crowdsensing. In: Chen, S., Choo, K.K., Fu, X., Lou, W., Mohaisen, A. (eds.) SecureComm 2019. LNICST, vol. 304. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37228-6_26
20. Alessi, M., Camillo, A., Giangreco, E.: Make users own their data: a decentralized personal data store prototype based on ethereum and IPFS. In: 2018 3rd International Conference on Smart and Sustainable Technologies, Split, vol. 7, pp. 1–7. (2018)
21. Randhir, K., Rakesh, T.: Implementation of distributed file storage and access framework using IPFS and blockchain. In: 2019 Fifth International Conference on Image Information Processing (ICIIP), Shimla, India, vol. 11, pp. 246–251 (2019)