

## Research Article

# A Blockchain-Enabled Trading Framework for Distributed Photovoltaic Power Using Federated Learning

Xuefeng Piao <sup>1</sup>, Hao Ding <sup>1</sup>, Huihui Song <sup>2</sup>, Meng Liu,<sup>3</sup> and Song Gao<sup>3</sup>

<sup>1</sup>School of Computer Science and Engineering, Harbin Institute of Technology, Weihai 264209, Shandong, China

<sup>2</sup>School of New Energy, Harbin Institute of Technology, Weihai 264209, Shandong, China

<sup>3</sup>Electric Power Research Institute, State Grid Shandong Electric Power Company, Jinan 250000, Shandong, China

Correspondence should be addressed to Huihui Song; songhh@hitwh.edu.cn

Received 30 November 2023; Revised 23 December 2023; Accepted 8 January 2024; Published 17 January 2024

Academic Editor: R. Palanisamy

Copyright © 2024 Xuefeng Piao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

As the integration of distributed energy into the power grid continues to rise, the significance of electricity transactions in promoting renewable energy consumption grows substantially. This necessitates the establishment of a robust electricity trading framework to facilitate reliable and trustworthy transactions for prosumers. This paper introduces an innovative blockchain-based electricity trading framework. Within this framework, we present a decentralized collaborative model training approach aimed at predicting the power generation of distributed photovoltaic power stations. Simultaneously, we propose a graph-based algorithm for efficiently matching producers and consumers. Our experiments, conducted on the Hyperledger Fabric blockchain platform using real-world datasets, demonstrate enhanced prediction accuracy compared to existing models. The framework effectively handles transactions within one second, sustaining a sending rate of up to 200 transactions per second. The outcomes of this study not only surpass the performance of existing prediction models but also provide valuable insights for the development of trading frameworks in actual production environments. The proposed framework stands as a reference point for constructing reliable trading infrastructures, thereby contributing to the acceleration of green electricity trading.

## 1. Introduction

Escalating global environmental problems and energy demand, coupled with the steady development of new energy technologies, have been opening up new opportunities for the promotion of new energy sources. In the context of global advocacy of energy conservation and emission reduction, photovoltaic (PV) power, as an important part of new energy, has been vigorously utilized in recent years, and the demand for PV power is increasing yearly [1, 2]. In 2019, more than 200 GW of renewable energy capacity was added globally, with PV power accounting for about 2.8% of global power generation [3]. An increase in PV power generation capacity typically requires a stronger consumption capability.

However, considering the instability of renewable energy sources, insufficient energy storage technology, inaccurate

load forecasting, and many other reasons, the self-regulation ability of power grid is limited. In addition, there are still some inadequacies in the existing electricity trading mechanism that need to be improved [4, 5]. For example, the traditional electricity trading mechanisms may not be able to respond effectively and accurately to customers' demand. As a result, the problem of how to improve the consumption capacity of distributed PV power generation has become increasingly prominent.

Therefore, a more effective and efficient electricity trading framework is urgently needed to satisfy the requirements of both energy producers and consumers to the greatest extent. And then, the new energy consumption can be facilitated while the needs of producers and consumers can be met [4, 5]. This requires cooperation between both power producers and consumers: the producers are supposed to accurately forecast their own power generation to avoid

oversupply or undersupply and ensure a stable trading environment, while the consumers have to trade with the most suitable power producers to secure their own interests.

This paper explores the application of federated learning (FL) and blockchain technology in peer-to-peer (P2P) electricity trading and proposes a novel blockchain-enabled trading framework for distributed photovoltaic power using FL. The main contributions of this paper are as follows:

- (i) We design a decentralized collaborative model training approach using blockchain-based federated learning (BFL) with the consideration of data privacy preservation in distributed PV power stations
- (ii) We develop a graph-based matching algorithm by abstracting the producers and consumers of electricity trading into an equivalent graph model and utilize a smart contract to execute the proposed matching algorithm automatically
- (iii) We implement the proposed framework, called blockchain-based federated learning with smart contract (BFLSC) framework, on the Hyperledger Fabric platform and validate its effectiveness using real-world datasets, which provides a better reference for the developers of blockchain

The rest of our paper is organized as follows. The related work is surveyed in Section 2, followed by the description of the decentralized collaborative power prediction model designed by BFL in Section 3. Then, Section 4 gives the design of transaction matching algorithm and the implementation of smart contract. Section 5 introduces the proposed BFLSC framework. The experiment and analysis are provided in Section 6. Eventually, we conclude our work with a summary in Section 7.

## 2. Related Work

Recently, there have been many researches on prediction of power generation and on blockchain technologies in electricity trading. For power predicting, Dutta et al. [6], in the context of short-term PV power forecasting, shift from meteorological-based prediction to an approach grounded in historical electricity data. This modification has led to heightened accuracy. However, the method exhibits limitations in predicting large-scale variations, necessitating additional recent historical data as supplementary support. Zhou et al. [7] propose a multivariate hybrid prediction system that integrates signal decomposition and artificial intelligence techniques. This system leverages independent variable features to enhance the accuracy of solar power generation predictions. But the incorporation of multiple variables implies a greater demand for data. Almonacid et al. [8] try to employ a dynamic artificial neural network for PV power prediction and achieve a satisfactory margin of error. Their method is regarded as a new prediction strategy with better generalization capability. Hossain and Mahmood [9] select features according to the correlation between different weather variables and PV power and

introduce long short-term memory for prediction. Different from the previous centralized processing, Yoo et al. [10] utilize FL to build a solar power generation prediction model based on data collected from local generators.

On the aspect of applying blockchain in electricity trading, Oprea et al. [11] embed two novel settlement mechanisms for P2P electricity exchange into smart contracts as stored procedures, improving the classic pairwise settlement. Alao and Cuffe [12] devise the trading strategy based on smart contract arrangements, thus reducing the volatility risk for power producers. In addition to combining smart contracts with settlement mechanisms and trading strategies, Liu et al. [13] execute smart contracts to implement the proposed proof-of-benefit consensus protocol and protect the system from potential risks. Vieira and Zhang [14] rely on blockchain technology and auction mechanism to develop two market frameworks, facilitating autonomous P2P energy transactions in microgrids. Han et al. [15] make use of the fact that smart contracts can strictly implement the designed trading and payment rules to encourage direct energy trading between producers and consumers. Furthermore, Ping et al. [16] propose a joint energy-reserve prosumer-centric market based on blockchain, contributing to an autonomous and trustworthy prosumer-centric market.

As summarized in Table 1, previous studies have extensively explored various strategies for the smart sustainable grid. In the realm of power prediction research, a predominant approach involves relying on large-scale data aggregated from individual power stations to enhance accuracy. However, the growing emphasis on data security has made the aggregation of large-scale data increasingly challenging. FL, as a distributed machine learning approach, presents itself as a novel method for collaborative multiparty data analysis.

While there have been efforts to address the issue of insufficient local data through FL, the traditional FL model faces challenges such as single-point failure and potential malicious data, stemming from its centralized architecture and dependence on a central server [17]. Furthermore, in the domain of electricity trading, the application of smart contracts [18] has predominantly focused on transaction pricing and arrangements. However, achieving accurate matching between producers and consumers is equally vital for establishing a reliable and trusted trading environment.

## 3. Decentralized FL Training Model for Prediction

FL is a distributed machine learning approach which enables model training on a large corpus of decentralized data [19, 20]. It uses local data of all parties involved in training to update the global model by exchanging information of model rather than data, so as to break data islands and protect data privacy. In this paper, we achieve the decentralized architecture of FL by using blockchain technology and merge convolutional neural network and long short-term memory (CNN-LSTM) as a hybrid training model with

TABLE 1: An overview of related work.

Type	Source	Contribution	Limitation
Power predicting	[6]	An approach grounded in historical electricity data	Additional recent historical data
	[7]	The combination of multiple variables	Greater demand for data
	[8]	Dynamic artificial neural network	Greater demand for data
	[9]	The correlation between different weather variables and PV power	Greater demand for data
	[10]	FL	Insecurity and single-point failure
Blockchain in electricity trading	[11]	The introduction of settlement mechanism	Neglect of transaction subjects
	[12]	Blockchain for risk management	Neglect of transaction subjects
	[13]	New consensus protocol	Neglect of blockchain performance
	[14]	Auction mechanism for microgrids	Incapability of automated transactions
	[15]	Smart contract for implementation of rules	Without considering future generation capacity
	[16]	Proposal of a joint energy-reserve prosumer-centric market	Without considering future generation capacity

attention mechanism for predicting the power generation of each PV power station.

**3.1. The Proposed Hybrid Model of BFL.** The proposed hybrid model with attention mechanism is composed of input layer, CNN layer, LSTM layer, attention layer, full connection layer, and output layer, as shown in Figure 1. The model input is the time-series data composed of power generation and related environmental factors considering the influence of external environment, such as illumination and temperature. The time-series data is a one-dimensional grid obtained by regular sampling on the time axis. Therefore, CNN with strong capabilities in grid-like structured data processing is used to extract features of the power generation data. The CNN layer consists of two 1D convolution layers and two pooling layers; this is simple but effective. The first convolution constitutes the feature map of input data, and the second convolution is conducted on the feature map created in the first convolution with an attempt to further amplify the significant features. The pooling layer uses max-pooling which selects the maximum value on feature area for down-sampling to reduce the number of features and simplify the feature map. The activation function in the CNN layer is the ReLU function, and its formula is  $\text{ReLU}(x) = \text{def} \max(0, x)$ .

The LSTM layer is used to learn the variation law of power generation because it is apt at processing longer sequences of time data [21]. The LSTM employs gating mechanism to selectively retain or forget historical information with a forget gate  $f_t$ , input gate  $i_t$ , and output gate  $o_t$ . The  $f_t$  selectively retains the hidden state information of the previous cell to the current cell, the  $i_t$  selectively retains the information of input to the cell unit, and the  $o_t$  selectively retains the current internal state information and outputs it to the next cell unit. The basic cell structure of the LSTM network model is shown in Figure 2.

In the figure,  $x_t$  and  $g_t$  are the input and candidate state of the current cell, respectively,  $c_{t-1}$  and  $c_t$  are the internal state of the previous cell and the current cell, respectively,

$h_{t-1}$  and  $h_t$  are the hidden state of the previous cell and current cell, respectively, and  $\sigma$  and  $\tanh$  are the sigmoid function and hyperbolic tangent function, respectively. The calculation formulas are shown in the following.

$$\begin{aligned}
 f_t &= \sigma(w_f[h_{t-1}, x_t] + b_f), \\
 i_t &= \sigma(w_i[h_{t-1}, x_t] + b_i), \\
 o_t &= \sigma(w_o[h_{t-1}, x_t] + b_o), \\
 g_t &= \tanh(w_g[h_{t-1}, x_t] + b_g), \\
 c_t &= f_t \circ c_{t-1} + i_t \circ g_t, \\
 h_t &= o_t \circ \tanh(c_t),
 \end{aligned} \tag{1}$$

where  $w_f, w_i, w_o, w_g$  and  $b_f, b_i, b_o, b_g$  represent the weight matrix and bias matrix of  $f_t, i_t, o_t, g_t$ , respectively, and  $\circ$  represents the Hadamard product. The process of calculation is as follows: (1)  $i_t, o_t$ , and  $g_t$  are calculated separately. (2)  $c_t$  is obtained by combining  $f_t$  and  $i_t$ . (3)  $h_t$  of the current network is acquired using  $o_t$ .

Additionally, to improve the efficiency of model and saving computing resources, we also introduce attention mechanism into the model, which is a processing method simulating the cognitive attention of the human brain [22]. The attention mechanism distributes feature weights according to the feature importance; then, the neural network will only select the key information for processing when a large amount of input information from the previous layer needs to be processed. The parameters of attention mechanism are calculated by the following.

$$\begin{aligned}
 \alpha_i &= \frac{\exp(\sigma(wh_i + b))}{\sum_{j=1}^n \exp(\sigma(wh_j + b))}, \\
 y &= \sum_{i=1}^n \alpha_i x_i,
 \end{aligned} \tag{2}$$

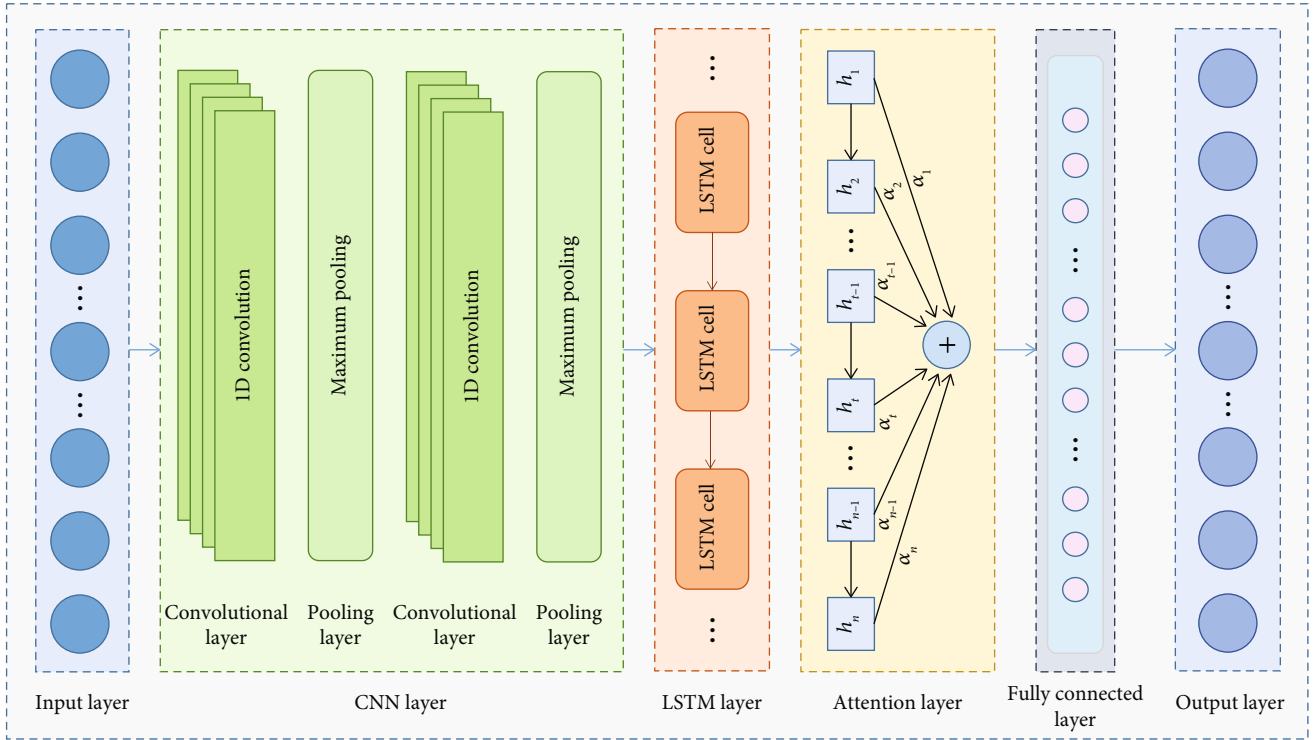


FIGURE 1: The structure of the hybrid model with attention mechanism.

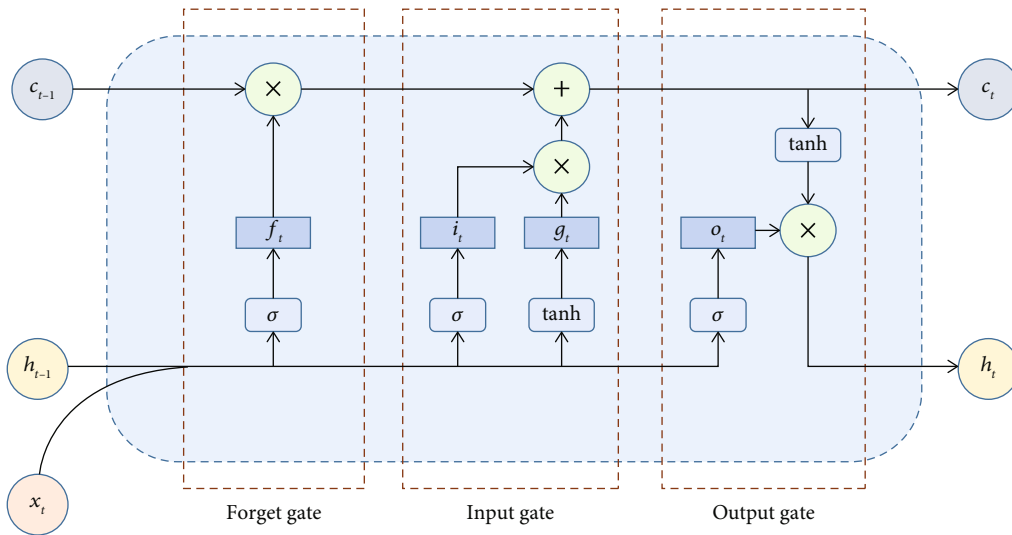


FIGURE 2: Basic cell structure of the LSTM network.

where  $\sigma$  represents attention scoring function,  $w$  represents weight matrix,  $b$  represents bias coefficient,  $x_i$  represents the input of the previous layer,  $h_i$  represents the output of the previous layer and the input of the current attention mechanism layer,  $\alpha_i$  represents the value of the attention probability distribution, and  $y$  represents the output of attention mechanism layer.

Eventually, the future power generation is acquired through the full connected layer and output layer.

3.2. *The Execution Process of BFL.* Traditional FL counts on a central server to aggregate parameters and update the global model [23] as shown in Figure 3(a). Despite the data-preserving capability of traditional FL, its centralized architecture is vulnerable to the malfunction of the server. Fortunately, the advent of blockchain provides more opportunities for the development of FL, by which the aggregation and update can be realized in a distributed and secure manner as shown in Figure 3(b). In our BFL, the

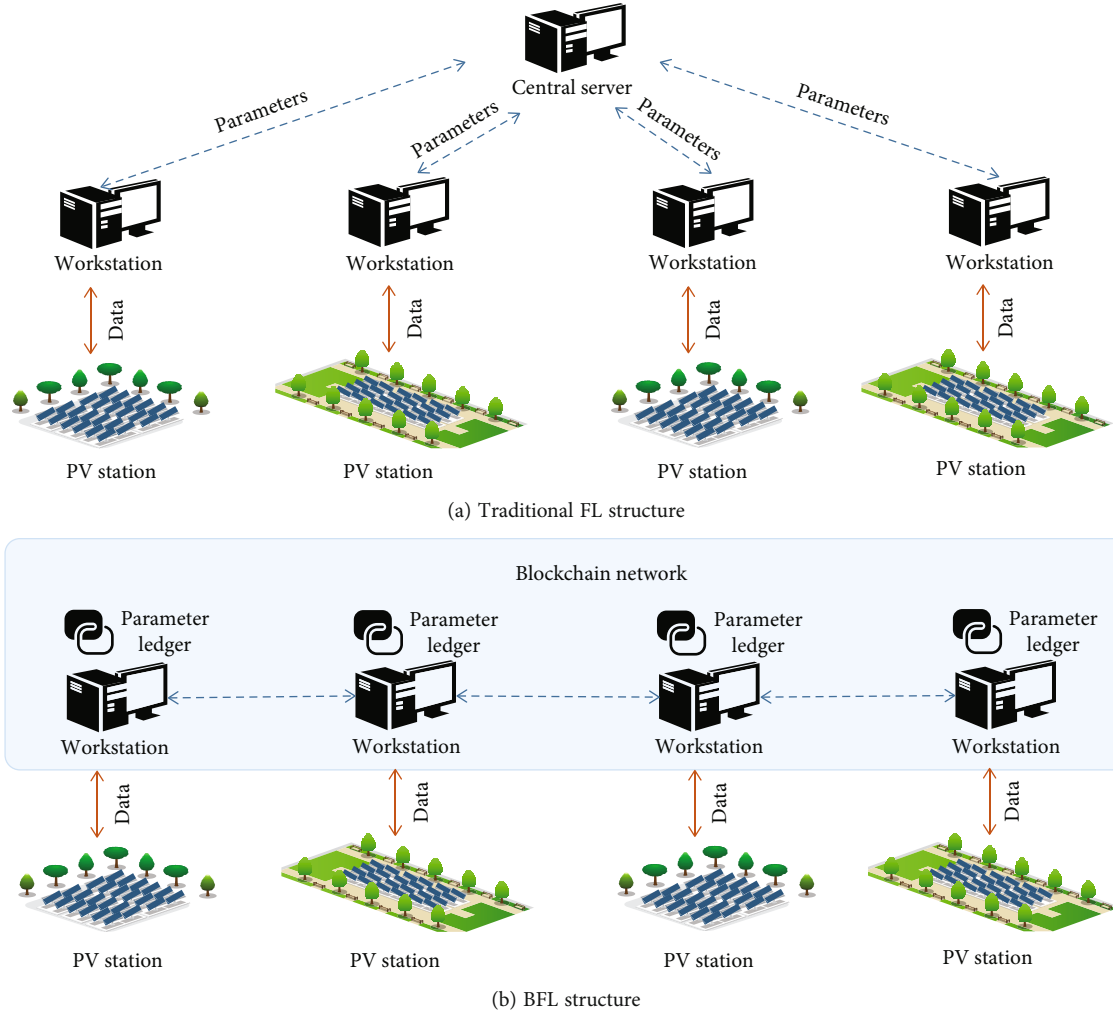


FIGURE 3: The comparison of traditional FL and BFL.

blockchain serves as an immutable distributed ledger to record and store model updates; thus, a decentralized FL structure can be achieved while preventing attacks from possible malicious node.

We assume that there are  $K$  number of PV power stations, and each PV power station has a corresponding computer workstation which is regarded as a node in the blockchain network. The local dataset of a node  $i$  ( $1 \leq i \leq K$ ) is denoted as  $D_i$ , the total number of data in  $D_i$  is denoted as  $n_i$ , and  $\sum_{i=1}^K |D_i|$  is denoted as  $N$ . Algorithm 1 depicts an overview on the BFL execution, and the execution process will be described in detail below.

The initial parameters of the prediction model mentioned in Section 3.1 are manually set to genesis block which is the first block of the chain at the beginning. Then, all nodes use these parameters for initialization at the beginning of training separately. We define the structure of data stored in the ledger with the node ID, denoted by `nodeID`; local model parameter, denoted by `localParm`; and the corresponding training epoch, denoted by `epochNum`, i.e., each data item in the ledger represents the `localParm` of `nodeID` in `epochNum`.

Let us denote the local model parameter of node  $i$  and the global model parameter in epoch  $t$  by  $w_i^t$  and  $w_0^t$ , respectively. And the local model parameter of node  $i$  in the next epoch of  $t$  is denoted as  $w_i^{t+1}$ . Then,  $w_i^{t+1}$  minimizes  $f(w_i)$ , namely,  $\arg \min f(w_i)$  in the epoch, during the local model training using  $D_i$ . The following Equation (3) defines the  $f(w_i)$ , and then,  $w_i^{t+1}$  can be obtained by Equation (4) according to the gradient descent algorithm.

$$f(w_i) \stackrel{\text{def}}{=} \frac{1}{n_i} \sum_{j=1}^{n_i} \sigma(x_{ij}, w_i), \quad (3)$$

$$w_i^{t+1} \leftarrow w_i^t - \eta \nabla f(w_i^t), \quad (4)$$

where  $x_{ij}$  represents the sample indexed by  $j$  in  $D_i$ ,  $\sigma$  represents the loss function,  $\eta$  represents the learning rate of gradient descent, and  $\nabla f(w_i)$  represents the calculated gradient of  $f(w_i^t)$ .

During training, the local model parameters of each node will be updated in every epoch. For example, the current  $w_i^t$  will be updated to  $w_i^{t+1}$  for a node  $i$  after local model

```

Input:  $w_0^1, t_{wait}, D_i, K$ 
Output: The global model  $\mathcal{M}$ 
1 begin
2    $w_i^t \leftarrow w_0^1; t \leftarrow 0; flag \leftarrow true;$ 
3   while global model training has not yet been completed do
4     use  $w_i^t$  and  $D_i$  for local model training;
5      $w_i^{t+1}$  obtained by Equation (4);
6     nodeID  $\leftarrow i$ ; epochNum  $\leftarrow t + 1$ ; localParm  $\leftarrow w_i^{t+1}$ ;
7     store {nodeID, epochNum, localParm} to the ledger;
8     while  $flag = true$  do
9       sleep ( $t_{wait}$ );
10      totalNum  $\leftarrow$  amount of records where  $epochNum = t + 1$  in the ledger;
11      if totalNum =  $K$  then
12         $flag \leftarrow false$ ;
13      end
14    end
15    for  $i \in K$  do
16       $w_i^{t+1} \leftarrow$  localParm of node  $i$  where epochNum =  $t + 1$  in the ledger;
17    end
18     $w_0^{t+1}$  obtained by Equation (5);
19    update  $\mathcal{M}$  with  $w_0^{t+1}$ ;
20     $t \leftarrow t + 1$ ;
21     $w_i^t \leftarrow w_0^t$ ;
22  end
23  return  $\mathcal{M}$ 
24 end

```

ALGORITHM 1: BFL execution.

training of epoch  $t$ . Whenever the local model parameter of a node is updated, the node stores its own node ID, the number of next epoch, and the obtained new local model parameter to the distributed ledger. After that, the node waits for a time interval, denoted by  $t_{wait}$ , in order to allow each node to determine whether the global model update can be carried out in current epoch. Once the interval reaches  $t_{wait}$ , the node queries the amount of records where epochNum =  $t + 1$  from the ledger. If the amount equals to  $K$ , it indicates that local model training and parameter recording have been completed on all nodes. Then, the node can gather the local model parameters of all nodes from the ledger and locally update a new global model parameter by Equation (5) which aggregates the gathered parameters based on a weighted average. Otherwise, the node has to wait for  $t_{wait}$  time interval again until the work is done on all nodes. Overall, the whole process mentioned above including local model training, local parameter recording, and global model update is repeated until the global model training is completed.

$$w_0^{t+1} \leftarrow \sum_{i=1}^K \frac{n_i}{N} w_i^{t+1}. \quad (5)$$

#### 4. Real-Time Electricity Trading Strategy Based on Graph Theory and Smart Contract

4.1. *Graph-Based Algorithm for Matching.* Accurate matching of producers and consumers can better maintain the bal-

ance between supply and demand in the power system while accurate matching requires appropriate matching algorithms to achieve. In the following, we propose an efficient matching algorithm based on graph theory.

The processing of semistructured data usually relies on graph theory which is a mathematical model describing the relationships between objects [24, 25]. As a model of graph theory, the bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets. The properties of electricity trading can be conveniently described by means of a diagram consisting of two sets of vertices with a set of edges joining certain pairs of these vertices. Therefore, an undirected graph  $G = (U, V, E)$  is used to represent an electricity trading system. Where the vertex sets  $U$  and  $V$  represent the producers and consumers, respectively, the edge set  $E$  represents the transaction links between them. In the graph  $G$ , an element edge  $e_i \in E$  has a weight  $\tilde{w}_i$  ( $\tilde{w}_i \geq 0$ ) which represents the power quantity traded by a matched pair of producer and consumer. The elements  $u_i \in U$  and  $v_i \in V$  represent a producer and a consumer, respectively. For convenience of description, the predicted power of a producer  $u_i$  is denoted as  $S_i$ , the demanded power of a consumer  $v_i$  is denoted as  $R_i$ , and the weight between the  $u_i$  and  $v_i$  is denoted as  $\tilde{w}(u_i, v_i)$ .

According to the given bipartite graph  $G$ , the matching problem between the two sides of electricity trading can be modeled as Equation (6) with the purpose of promoting power consumption. The objective function  $f$  is to maximize the quantity of electricity traded with three constraint

conditions: (1) transactions made are as few as possible; (2) for all producers, the amount of electricity traded should not exceed the predicted power generation; (3) for all consumers, no more power is traded than they need.

$$\begin{aligned}
\max f &= \sum_{i=1}^{|E|} \tilde{w}_i \\
\text{s.t. min } &|E| \\
\forall u_j \in U, &\sum_{k=1}^{|V|} \tilde{w}(u_j, v_k) \leq S_j \\
\forall v_j \in V, &\sum_{k=1}^{|U|} \tilde{w}(u_k, v_j) \leq R_j.
\end{aligned} \tag{6}$$

The greedy strategy is a common strategy for solving optimal problems. This strategy greedily chooses the best available option at each step without considering any potential consequences, but it is very convenient and effective. To solve the above mathematical model, we apply the greedy strategy which makes the locally optimal choice at each stage, i.e., meeting the maximum demand with as few producers as possible at each stage.

Let us assume that the sets of  $U$  and  $V$  are sorted in descending order by the power generation and power demand, respectively. Then,  $S_i \geq S_{i+1}$  and  $R_i \geq R_{i+1}$  for all  $i$  (1, 2, 3, ...). Additionally, the maximum tradable quantity of electricity must be less than or equal to  $\min(\sum_{i=1}^{|S|} S_i, \sum_{i=1}^{|R|} R_i)$ . In this context, there are two cases for the  $S_1$  and  $R_1$ : one case is that  $S_1 \geq R_1$ , another case is that  $S_1 < R_1$ . For the first case, a new edge would be built in the  $G$ , and the weight of the edge would be assigned to  $R_1$ . For another case, it should be searched backwards to  $k$  that satisfies the condition  $\sum_{i=1}^k S_i \geq R_1$ . If there is a such  $k$ , the  $k$  number of new edges would be built. The weights of the first  $k-1$  edges would be assigned to the corresponding  $S_i$ , and the weight of  $k$ -th edge would be assigned to  $R_1 - \sum_{i=1}^{k-1} S_i$ . Otherwise, a new edge would be built for each supplier, and the weight of the edge would be assigned to the corresponding  $S_i$ . The following Algorithm 2 shows more details of our matching procedure. The time complexity of the matching algorithm that we designed is  $O(n \log n + m \log m)$ , where  $n$  is the data scale of the producers, and  $m$  is the data scale of the consumers.

Here, we give a proof of the correctness for our proposed greedy strategy. Without loss of generality, let us assume that the set of vertices currently connected to  $R_1$  to build edges in  $G$  is  $P\{S_1, S_2, \dots, S_k\}$  ( $k \geq 1$ ). Instead of selecting, choose  $S_1$  to build the edge with  $R_1$ , and now, we replace  $S_1$  with other elements in  $S$ . Due to the descending order of  $S$ , all elements in  $S$  are less than or equal to  $S_1$ . If  $k=1$  and there is  $S_i \notin P(S_i \geq R_1)$ , replacing  $S_1$  with  $S_i$  will not increase the number of edges in  $G$ , that is, the number of transactions. Otherwise, at least two elements  $S_i \notin P, S_j \notin P(S_i + S_j \geq S_1)$  are needed to substitute for the amount of electricity that  $S_1$  trades to  $R_1$ . In other words, there will be one new edge or more in

$G$ , which violates the condition of minimizing the number of edges in the graph. To sum up, this substitution method is incorrect. By analogy, replacing any element in  $P$  will not make the result better. That means the greedy strategy applied by us can get the optimal solution.

**4.2. Smart Contract in Trading Strategy.** Relying on the consensus mechanism and immutable data of blockchain, smart contract has gradually developed from an agreement where participants jointly fulfill digital commitments to an agreement enforced automatically by computer programs without the intervention of a trusted third party [18, 26]. In electricity trading systems, smart contracts can be utilized to specify rules concerning how and when to conduct electricity trading [27]. In our work, the smart contract is used to make the matching procedures progressed automatically in order to provide a reliable and credible environment.

The proposed BFLSC framework provides market matching services and transaction recording services for producers and consumers using smart contract as shown in Figure 4. Energy producers send predicted tradable power to the smart contract in hopes of achieving consumption of PV while energy consumers submit their energy demand which meet their production and living needs to the smart contract.

Our smart contract is built upon blockchain to ensure the correct execution. Once the smart contract is invoked, the contract clauses will be executed automatically as the business logic of electricity trading as shown in Figure 5.

Above all, producers provide the real-time-predicted power generation, and the consumers provide the power demand. These supply and demand serve as the basis for subsequent transactions between two parties. In the second place, the smart contract executes the proposed matching procedure to promote deals between them. Later, producers and consumers reach a consensus on electricity trading. Whenever new transactions are created, the transaction information will be recorded in the distributed ledger of blockchain for storage to ensure the security. Notice that the whole procedure which forms a real-time trading contract is completed between producers and consumers in a P2P manner without the intervention of a third party. As a result, the efficiency and security of trading can be greatly enhanced.

## 5. The BFLSC Framework for P2P Electricity Trading

The proposed BFLSC framework consists of two modules to summarize: the collaborative prediction method for power generation using BFL and a real-time trading strategy based on smart contract. Our approach involves using BFL to forecast the future power generation of each station and subsequently implementing real-time peer-to-peer (P2P) trading with smart contracts, as illustrated in Figure 6. In this trading framework, distinct PV stations within the same region are considered clusters. This clustering is based on the significant similarity observed in power generation among different PV stations within the same region and time frame.

```

Input:  $G(U, V, E)$ 
Output:  $totalQ, G$ 
1 begin
2   the total amount of electricity traded  $totalQ \leftarrow 0$ ;
3   a sequence  $\bar{a}(a_1, a_2, a_3, \dots) \leftarrow U$  sorted by descending order of  $S$ ;
4   a sequence  $\bar{b}(b_1, b_2, b_3, \dots) \leftarrow V$  sorted by descending order of  $R$ ;
5   maximum tradable quantity  $maxQ \leftarrow \min(\sum_{i=1}^{|\bar{a}|} S_{a_i}, \sum_{i=1}^{|\bar{b}|} R_{b_i})$ ;
6   while  $totalQ \neq maxQ$  do
7     sort  $\bar{a}$  and  $\bar{b}$  in descending order;
8     if  $S_{a_1} \geq R_{b_1}$  then
9       create edge  $(a_1, b_1)$  in  $G$ ;
10       $\tilde{w}(a_1, b_1) \leftarrow R_{b_1}$ ;
11       $S_{a_1} \leftarrow S_{a_1} - \tilde{w}(a_1, b_1)$ ;
12      if  $S_{a_1} = 0$  then
13        remove  $a_1$  from  $\bar{a}$ ;
14      end
15      remove  $b_1$  from  $\bar{b}$ ;
16       $totalQ \leftarrow totalQ + \tilde{w}(a_1, b_1)$ ;
17    else
18      if there is a  $k$  satisfying  $\sum_{i=1}^k S_{a_i} \geq R_{b_1}$  then
19        for  $i \leftarrow 1$  to  $k$  do
20          create edge  $(a_i, b_1)$  in  $G$ ;
21          if  $i \neq k$  then
22             $\tilde{w}(a_i, b_1) \leftarrow S_{a_i}$ ;
23            remove  $a_i$  from  $\bar{a}$ ;
24          else
25             $\tilde{w}(a_i, b_1) \leftarrow R_{b_1} - \sum_{j=1}^{k-1} S_{a_j}$ ;
26             $S_{a_i} \leftarrow S_{a_i} - \tilde{w}(a_i, b_1)$ ;
27          end
28           $totalQ \leftarrow totalQ + \tilde{w}(a_i, b_1)$ ;
29        end
30        if  $S_{a_1} = 0$  then
31          remove  $a_1$  from  $\bar{a}$ ;
32        end
33        remove  $b_1$  from  $\bar{b}$ ;
34      else
35        for  $i \leftarrow 1$  to  $|\bar{a}|$  do
36           $\tilde{w}(a_i, b_1) \leftarrow S_{a_i}$ ;
37          remove  $a_i$  from  $\bar{a}$ ;
38           $R_{b_1} \leftarrow R_{b_1} - \tilde{w}(a_i, b_1)$ ;
39           $totalQ \leftarrow totalQ + \tilde{w}(a_i, b_1)$ ;
40        end
41      end
42    end
43  end
44  return  $totalQ, G$ 
45 end

```

ALGORITHM 2: The matching procedure.

Moreover, the PV power generation sources and consumers should be connected within a network to facilitate communication between consumers and producers. Simultaneously, a blockchain network is established among PV stations for implementing BFL and executing smart contracts.

We employ BFL to train our innovative hybrid CNN-LSTM model, which incorporates an attention mechanism for predicting power generation in each PV power station.

The process begins with stations within a regional cluster gathering local small-scale historical power generation data and conducting data preprocessing, including normalization to eliminate noise data effects. Before BFL execution, the global model is initialized by manually setting its initial parameters to the genesis block.

Subsequently, each node utilizes its dataset for local model training, and the results are recorded in the blockchain's



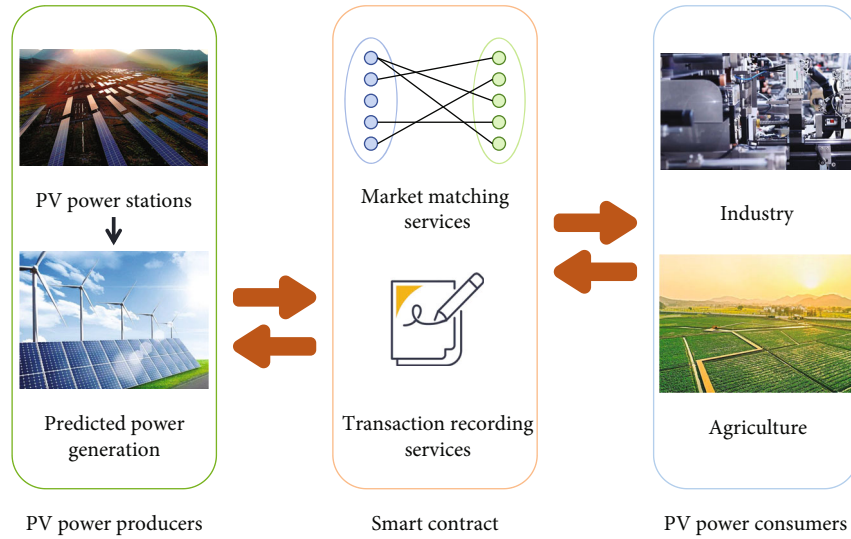


FIGURE 4: Smart contract in the framework.

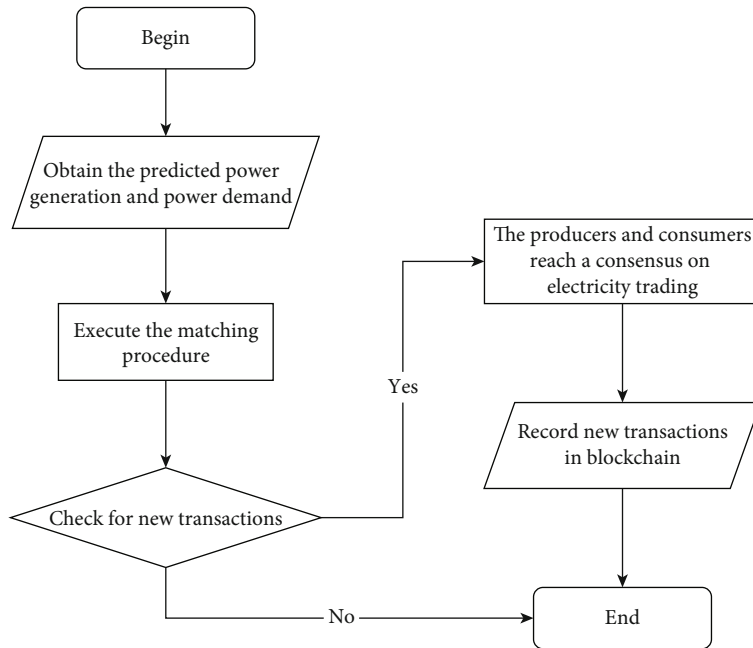


FIGURE 5: Smart contract execution.

distributed ledger. After completing local training tasks, nodes collect and update a new global model parameter using the FedAvg algorithm. This parameter is then used to replace local models iteratively until global model training is complete. Once BFL is done, nodes can employ the global model to predict future power generation.

In real-time, new actual power generation data is dynamically added to the local dataset, ensuring the dataset's continuous expansion. This BFL-based approach enhances the accuracy and real-time performance of power prediction for PV power stations, showcasing excellent generalization applicable to various time-series prediction problems while maintaining data security.

For the real-time trading strategy based on smart contracts, the system relies on the blockchain's distributed ledger, with producers and consumers acting as network nodes facilitating smart contracts. The trading system incorporates a time trigger, automatically invoking the smart contract interface and executing clauses based on preset business logic at the specified interval.

The smart contract acquires predicted future power generation results from each distributed PV power station for accurate power bidding. Simultaneously, it obtains electricity demand from consumers and matches power producers with consumers using an algorithm, promoting efficient electricity consumption. All transaction records are securely

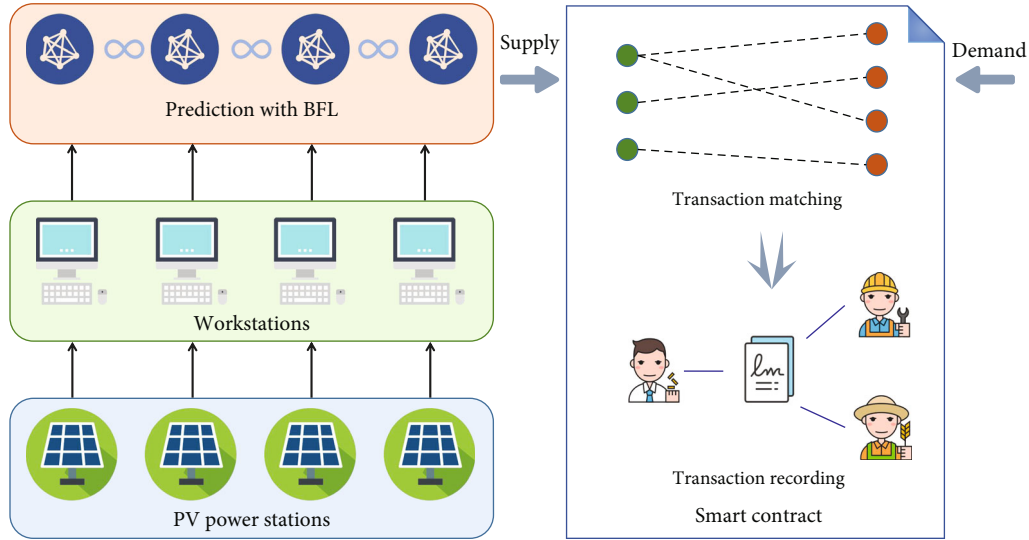


FIGURE 6: The BFLSC framework.

written into the blockchain, ensuring the reliability and credibility of transactions between the two parties.

Within the same cluster, PV station nodes preprocess small-scale historical local power generation data using normalization to eliminate noise data effects. All nodes then utilize BFL to train the hybrid CNN-LSTM model with an attention mechanism. Upon completing training task orchestration, nodes use the global model to predict future power generation. The system operates with automatic matching of producers and consumers using a bipartite graph mode to maximize quantity and minimize transactions. Transaction information is recorded in the blockchain's distributed ledger to ensure genuine, trustworthy, and immutable transactions. Both transaction matching and recording services are implemented as smart contract interfaces.

## 6. Experiments and Analysis

Prediction accuracy is an important indicator to measure the fit of model. It describes the degree to which the predicted value produced by the model fits the historical actual value. Transaction latency is a significant metric in blockchain systems. The latency performance of transaction is crucial to the blockchain in assessing its effectiveness. In this section, we evaluate the proposed BFLSC framework in terms of prediction accuracy and transaction latency.

P2P electricity trading has certain requirements for the identity of participants, the transaction information between them cannot be disclosed, and real-time transactions require high performance on transaction latency and throughput. In our experiments, according to the above reasons with the comparison of different blockchain types in Table 2, we built a blockchain network using the Hyperledger Fabric platform which is the most widely used consortium blockchain. The Hyperledger Fabric differs from other blockchains in that it introduces the concept of channels. The participants in the same channel have a separate ledger and maintain the ledger

collaboratively. Furthermore, a fabric network can have multiple channels, and each channel is isolated from each other [28].

In our experiments, we regard each PV power station as a Hyperledger Fabric organization with two peer nodes and an orderer node. Four organizations, R1, R2, R3, and R4, establish a network with configuration, CC1 and CC2. Organization R1 has peer nodes, P1 and P2, and an orderer node, O1 and P1, hosts a copy of ledger L1, and smart contract, S1 and P2, hosts a copy of ledger L2 and smart contract S2, respectively. Organizations R2, R3, and R4 also have their own peer nodes, orderer node, distributed ledgers, and smart contracts as shown in Figure 7. There are two channels, C1 and C2. The channel C1 is used to share the local model parameters during the BFL process, and the channel C2 is used to record the transaction information. Note that all organizations are joint to the both channels. The L1 and L2 store the model parameters and transaction information, respectively. A1 is an application of predicting future power generation by BFL for the channel C1. The smart contract S1 is the operation interface of model parameters in the ledger L1. A2 is an application of real-time electricity trading for the channel C2. The smart contract S2 is the interface to provide matching services and recording services.

Our experiments are conducted on the Hyperledger Fabric v2.0 system. We deploy a two-channel fabric network on 4 physical servers in an isolated LAN with 1,000 MB/s network bandwidth. The hardware configuration of each server is a 4-Core CPU, 16 GB main memory, and 1 TB magnetic disk, and each server is installed with Ubuntu Server 20.04 LTS 64bit operating system.

*6.1. Experiments for Prediction Performance.* To verify the generality of our BFLSC framework in real-world power scenarios, we use PV power data from the Desert Knowledge Australia Solar Centre (DKASC) [29] as the dataset. More than 30 PV power station systems are installed in the center, and the power generation data with corresponding environmental parameters of the PV power station system is

TABLE 2: A comparison between different types of blockchain.

	Public blockchain	Private blockchain	Consortium blockchain
Degree of decentralization	High	Low	Middle
Consensus mechanism	PoW, PoS	PBFT, RAFT	PBFT, RAFT
Oriented group	Public	Individual or organization	Multiple organizations
Application scenario	Virtual currency	Internal audit	Transactions between organizations
Examples	Bitcoin, Ethereum	Multichain	Hyperledger Fabric
Performance of transaction	Low	High	Middle

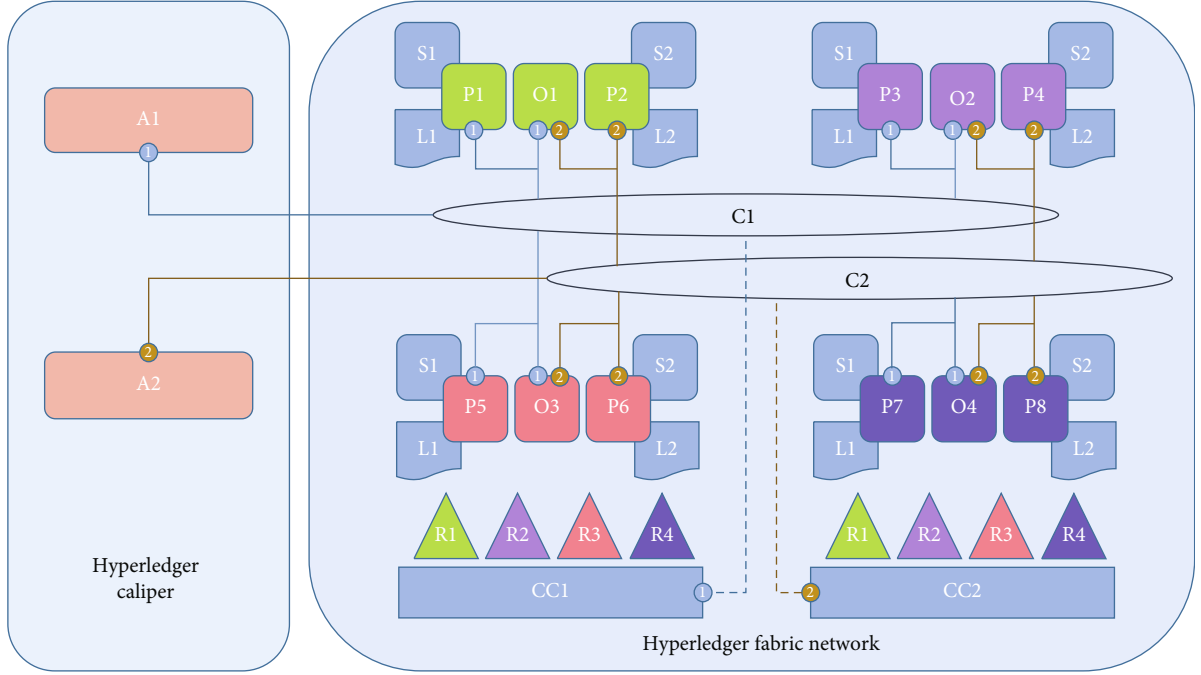


FIGURE 7: Network topology diagram of blockchain.

publicly accessible. Short-term prediction timescales typically range from 30 minutes to 72 hours [30]. Therefore, four distributed PV power stations located in DKASC, Alice Springs, are selected. The station codes are BP Solar, 2.0 kW, poly-Si, Fixed, 2008, North/Flat/East/West, and the data resolution is 5 minutes; the data ranges from November 6, 2022, to November 9, 2022. We label these four power stations as station A/B/C/D, respectively, to simplify the expression. Considering the possible influence of dimension attached to data, we scale the data by min-max normalization and then control the data range within the given interval. The following Equation (7) shows how to perform the min-max normalization. Once the data preprocessing is done, the overall statistical distribution of data will turn consistent with the elimination of noise data. Then, the dataset is divided into a training set and a test set at a ratio of 8:2. The training set is used to train the model while the test set is used to compare with the prediction results of the model to calculate the prediction accuracy.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad (7)$$

where  $x$  represents sample data,  $x'$  represents normalized sample data, and  $\max(x)$  and  $\min(x)$  represent the maximum and minimum values of sample data, respectively.

We adopt mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), and coefficient of determination ( $R^2$ ) to indicate the prediction accuracy more directly as the following Equations (8)–(11) [31]. For MAE, MSE, and RMSE, their values measure the deviation between the predicted and actual values. So the smaller their values, the more accurate the prediction results. While for  $R^2$ , it portrays the proximity of predicted and actual values. Therefore, the larger their values, the more accurate the prediction results. In addition, the  $R^2$  is more informative and truthful and does not have the interpretability limitations of MAE, MSE, and RMSE [32].

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (8)$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (9)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \quad (10)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}, \quad (11)$$

where  $y_i$  and  $\hat{y}_i$  represent the actual values and predicted values of PV power generation at a certain time, respectively,  $\bar{y}$  represents the average of  $y$ , and  $N$  represents the total number of samples involved in the prediction.

We compare the prediction accuracy of the proposed BFL model with that of the individual model and centralized model in an identical simulation configuration. The individual model means that each of the PV power stations uses individual data for model training separately, and the centralized model means that there is a central server that centralizes data from each power station for model training in the cluster. They train in different ways and with different datasets even though the goal of them is to get a trained hybrid model designed in Section 3.1.

Tables 3–6 provide a quantitative analysis of the prediction results by error indicators. The centralized model is expected to perform better than the individual model because it has access to the data of all stations while the individual model only uses its own data. From the viewpoint of  $R^2$ , the prediction results of the centralized model for all power stations are higher than those of the individual model, and the maximum increase of  $R^2$  is 2.8%. Meanwhile, from the perspective of the other three indicators, the accuracy of the centralized model is similar to that of the individual model. This demonstrates that despite the pain of gathering data and the risk of data leakage, the augmentation of the dataset empowers the centralized model to predict more accurately. However, the proposed BFL achieves the better performance on the MSE and RMSE indicators than the other models without a trade-off in privacy. Although the BFL model on Station D is larger than the individual model by 18.8% and is larger than the centralized model by 18.5% on the MAE indicator, the RMSE decreases by 13.6% and 14.1%, respectively, and the  $R^2$  especially increases by 4.0% and 2.4%, respectively. Despite certain shortcomings in performance as indicated by the MAE and  $R^2$  metrics, the BFL model has demonstrated a significant improvement in performance according to the MSE and RMSE metrics. The reason for the comparatively lower performance on Station D compared to Station A/B/C may be attributed to the presence of more noise in the data from Station D. But from a comprehensive perspective where all metrics are taken into account, the BFL model could achieve similar or even higher accuracy than the other models.

Figure 8 shows the comparison of prediction results and actual power generation. The attention paid to the actual data quality of Station D is different from that of the other stations, evident from the observed curve. It can be found that there is a discernible gap between the prediction results of the individual model and the centralized model compared with the actual power generation. Moreover, the obvious

TABLE 3: Comparison of MAE.

Station	Individual model	Centralized model	BFL model
Station A	0.223579	0.216478	0.203644
Station B	0.224377	0.214547	0.216227
Station C	0.201429	0.199328	0.193171
Station D	0.273766	0.274456	0.325277

TABLE 4: Comparison of MSE.

Station	Individual model	Centralized model	BFL model
Station A	0.125278	0.117591	0.115155
Station B	0.124583	0.123211	0.119748
Station C	0.113743	0.110812	0.107770
Station D	0.234406	0.237148	0.174886

TABLE 5: Comparison of RMSE.

Station	Individual model	Centralized model	BFL model
Station A	0.353947	0.342916	0.339346
Station B	0.352963	0.351014	0.346047
Station C	0.337259	0.332885	0.328283
Station D	0.484155	0.486978	0.418194

TABLE 6: Comparison of  $R^2$ .

Station	Individual model	Centralized model	BFL model
Station A	0.449324	0.462304	0.459666
Station B	0.440533	0.452443	0.431904
Station C	0.545689	0.547228	0.537538
Station D	0.303390	0.308074	0.315653

anomalies emerge in the prediction for the Station D due to the difference of data quality. On the contrary, the prediction results of our BFL model are not only close to the actual power generation but also perform well under abnormal conditions.

In contrast, the prediction results of our BFL model not only closely align with the actual power generation but also demonstrate robust performance under abnormal conditions.

**6.2. Experiments for Trading Performance.** In the Hyperledger Fabric, the transaction latency is the amount of time from the time point that transaction is submitted to the time point that the result is widely available in the network. The average latency refers to the average time taken in seconds between issuing a transaction and receiving the corresponding response [33]. The sending rate is a measure of how many transactions are sent to the network in a given time period, expressed as transactions per second (TPS). It is mentioned in [34] that the average latency under different sending rates of transactions is an important metric to evaluate the performance of a blockchain system. Therefore, we decide to use the metric to evaluate

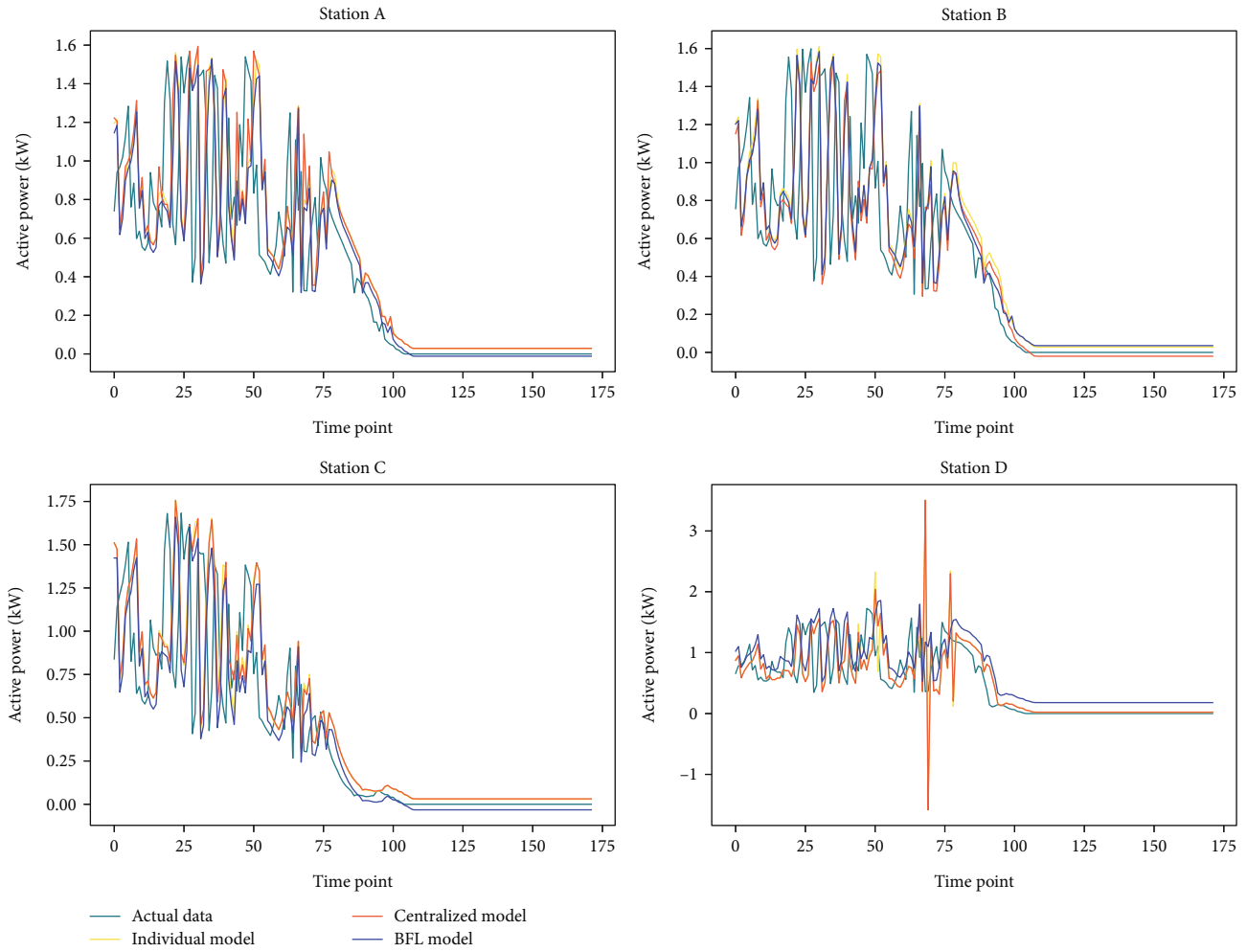


FIGURE 8: Prediction results of PV power stations.

TABLE 7: Experimental settings.

Parameter	Value	Unit
Number of organizations	4	/
Number of peer nodes	2	Per organization
Number of orderer nodes	1	Per organization
BatchSize	100/200/300	/
BatchTimeout	1	Second
Sending rate	20/40/60/80/100/120/140/160/180/200/220/240	TPS

the real-time performance of our trading platform and use hyperledger caliper which is an open-source benchmarking tool to get the average latency of transactions. Furthermore, generating a block of transactions depends on channel configuration parameters BatchSize and BatchTimeout, which directly affects the transaction latency. The BatchSize is the desired size that is allowed in each block, and the BatchTimeout is the maximum elapsed duration for creating a block. In other words, if the size of collected transactions is less than the BatchSize, the creation of the block has to wait for the BatchTimeout. Otherwise, the block will be created directly.

The experiments are performed for the different BatchSize with increasing the sending rate of transaction request from a client node in the same local network. The BatchSize is set to 100, 200, and 300, respectively, and the sending rate ranges from 20 TPS to 240 TPS with a 20 TPS interval. For the BatchTimeout, we set it to 1 second (minimum value) since a longer BatchTimeout will take longer to create blocks under the given sending rate. Table 7 provides a detailed reference of our experimental settings. Figure 9 shows the experimental results of the average transaction latency. The proposed BFLSC framework is able to cover the transactions well within one second, and the standard deviations are also

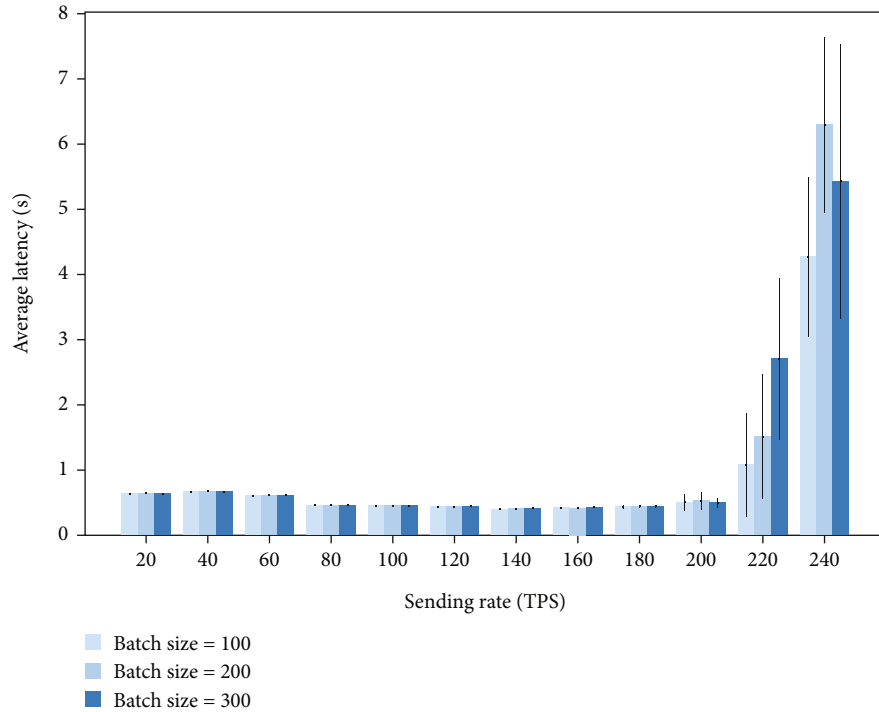


FIGURE 9: The average transaction latency.

low until the sending rate reaches 200 TPS. However, not only the average latency but also the standard deviation increase significantly due to bottlenecks in server performance when the sending rate is above 200 TPS.

It is worth noting that the average latency increases during the sending rate ranges from 20 TPS and 40 TPS, and it decreases during the sending rate ranges from 40 TPS to 100 TPS. The reason is that it should be waited to create a block up to the BatchTimeout reached when the sending rate is low, but the block can be created without waiting when the sending rate is high and the higher the sending rate has the faster block creation. Additionally, the latency fluctuation is relatively stable during the sending rate ranges from 100 TPS to 200 TPS. This is because the time used to create a block within the BatchTimeout is relatively stable.

## 7. Conclusion

In response to the initiative of distributed generation participating in market transactions locally, this paper takes distributed PV power as the research object and proposes the BFLSC framework from the perspective of privacy protection and data security. The framework includes collaborative prediction of power generation by the BFL approach and real-time trading strategy based on smart contract. Compared to the traditional FL method, the BFL method can predict future generation more accurately while ensuring data security. Additionally, the real-time trading strategy is executed automatically by smart contract. Once the real-time power generation predicted by the BFL is obtained,

the smart contract will invoke the designed graph-based algorithm to match the producers and consumers of electricity, prompting both parties to enter into power transactions. And then, the transaction information will be deposited by the blockchain. Simultaneously, the experiments implemented on the Hyperledger Fabric v2.0 network with real-world data corroborate the validity of our framework. This work has a guiding role in engineering practice for promoting the construction of blockchain-based electricity trading platform and accelerating green electricity trading.

In the current work, there are still certain limitations to the proposed approach. Optimization strategies for the performance of the blockchain network are in the exploratory stage, lacking targeted optimization measures. The latency of the blockchain network can be further reduced, and throughput can be increased. This can be achieved by designing more efficient consensus algorithms to shorten the time intervals for block generation. Additionally, optimizing the transaction processing flow and reducing the confirmation time for transactions can also enhance the scalability of the blockchain.

Faced with the shortcomings of traditional blockchain architectures in terms of scalability, throughput, latency, and other performance aspects, researchers in recent years have been exploring improvements by addressing the data structure and storage structure of blockchain. Solutions such as DAG-based blockchain and sharded blockchain have been designed. In the next phase of this research, we will continue to investigate how to achieve efficient electricity exchange on DAG-based blockchain and sharded blockchain with FL.

## Data Availability

The datasets analyzed during this study can be obtained from the corresponding author on reasonable requests.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

## Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant number 61773137) and the Natural Science Foundation of Shandong Province (grant number ZR2023ME162).

## References

- [1] B. Parida, S. Iniyar, and R. Goic, "A review of solar photovoltaic technologies," *Renewable & Sustainable Energy Reviews*, vol. 15, no. 3, pp. 1625–1636, 2011.
- [2] M. Hosenuzzaman, N. A. Rahim, J. Selvaraj, M. Hasanuzzaman, A. B. M. A. Malek, and A. Nahar, "Global prospects, progress, policies, and environmental impact of solar photovoltaic power generation," *Renewable & Sustainable Energy Reviews*, vol. 41, pp. 284–297, 2015.
- [3] J. Li, S. Chen, Y. Wu et al., "How to make better use of intermittent and variable energy? a review of wind and photovoltaic power consumption in China," *Renewable & Sustainable Energy Reviews*, vol. 137, article 110626, 2021.
- [4] R. Ambrosio, "Transactive energy systems," *Ieee Electrification Magazine*, vol. 4, no. 4, pp. 4–7, 2016.
- [5] S. Suthar, S. H. C. Cherukuri, and N. M. Pindoriya, "Peer-to-peer energy trading in smart grid: frameworks, implementation methodologies, and demonstration projects," *Electric Power Systems Research*, vol. 214, article 108907, 2023.
- [6] S. Dutta, Y. Li, A. Venkataraman et al., "Load and renewable energy forecasting for a microgrid using persistence technique," *Energy Procedia*, vol. 143, pp. 617–622, 2017.
- [7] Y. Zhou, J. Wang, Z. Li, and H. Lu, "Short-term photovoltaic power forecasting based on signal decomposition and machine learning optimization," *Energy Conversion and Management*, vol. 267, p. 115944, 2022.
- [8] F. Almonacid, P. J. Perez-Higueras, E. F. Fernandez, and L. Hontoria, "A methodology based on dynamic artificial neural network for short-term forecasting of the power output of a PV generator," *Energy Conversion and Management*, vol. 85, pp. 389–398, 2014.
- [9] M. S. Hossain and H. Mahmood, "Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast," *IEEE Access*, vol. 8, pp. 172524–172533, 2020.
- [10] E. Yoo, H. Ko, and S. Pack, "Fuzzy clustered federated learning algorithm for solar power generation forecasting," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 4, pp. 2092–2098, 2022.
- [11] S.-V. Oprea, A. Bara, and A. I. Andreescu, "Two novel blockchain-based market settlement mechanisms embedded into smart contracts for securely trading renewable energy," *IEEE Access*, vol. 8, pp. 212548–212556, 2020.
- [12] O. Alao and P. Cuffe, "Hedging volumetric risks of solar power producers using weather derivative smart contracts on a blockchain marketplace," *IEEE Transactions on Smart Grid*, vol. 13, no. 6, pp. 4730–4746, 2022.
- [13] C. Liu, K. K. Chai, X. Zhang, and Y. Chen, "Peer-to-peer electricity trading system: smart contracts based proof-of-benefit consensus protocol," *Wireless Networks*, vol. 27, pp. 4217–4228, 2021.
- [14] G. Vieira and J. Zhang, "Peer-to-peer energy trading in a microgrid leveraged by smart contracts," *Renewable & Sustainable Energy Reviews*, vol. 143, article 110900, 2021.
- [15] D. Han, C. Zhang, J. Ping, and Z. Yan, "Smart contract architecture for decentralized energy trading and management based on blockchains," *Energy*, vol. 199, p. 117417, 2020.
- [16] J. Ping, D. Li, Z. Yan, X. Wu, and S. Chen, "A trusted peer-to-peer market of joint energy and reserve based on blockchain," *Electric Power Systems Research*, vol. 214, article 108802, 2023.
- [17] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [18] Z. Zheng, S. Xie, H.-N. Dai et al., "An overview on smart contracts: challenges, advances and platforms," *Future Generation Computer Systems-The International Journal of Escience*, vol. 105, pp. 475–491, 2020.
- [19] K. Bonawitz, H. Eichner, W. Grieskamp et al., "Towards federated learning at scale: system design," *Proceedings of Machine Learning and Systems*, vol. 1, pp. 374–388, 2019.
- [20] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, article 106854, 2020.
- [21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [22] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.
- [23] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2020.
- [24] J. L. Gross, J. Yellen, and M. Anderson, *Graph Theory and Its Applications*, Chapman and Hall/CRC, 2018.
- [25] D. B. West, *Introduction to Graph Theory*, vol. 2, Prentice hall Upper Saddle River, 2001.
- [26] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, 1997.
- [27] D. Kirli, B. Couraud, V. Robu et al., "Smart contracts in energy systems: a systematic review of fundamental approaches and implementations," *Renewable & Sustainable Energy Reviews*, vol. 158, article 112013, 2022.
- [28] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *EUROSYS '18: Proceedings of the Thirteenth EUROSYS Conference*, New York, 2018.
- [29] "Dka solar centre," <http://dksolarcentre.com.au/>.
- [30] Y. Goude, R. Nedellec, and N. Kong, "Local short and middle term electricity load forecasting with semi-parametric additive models," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 440–446, 2014.
- [31] R. Ahmed, V. Sreeram, Y. Mishra, and M. D. Arif, "A review and evaluation of the state-of-the-art in PV solar power forecasting: techniques and optimization," *Renewable & Sustainable Energy Reviews*, vol. 124, article 109792, 2020.

- [32] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *Peerj Computer Science*, vol. 7, 2021.
- [33] "Hyperledger caliper," <http://github.com/hyperledger/caliper>.
- [34] X. Xu, G. Sun, L. Luo, H. Cao, H. Yu, and A. V. Vasilakos, "Latency performance modeling and analysis for hyperledger fabric blockchain network," *Information Processing & Management*, vol. 58, no. 1, 2021.