Power Data Credible Decision-Making Mechanism Based on Federated Learning and Blockchain

Xin Li[1], Fangjian Shang[1], Yanli Yao[1], and Tianren Zheng[2](✉)

[1] State Grid Jibei Information and Telecommunication Company, Beijing 100053, People's Republic of China
{li.xin.q,shang.fangjian,yao.yanli}@jibei.sgcc.com.cn
[2] Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China
zhtr@bupt.edu.cn

Abstract. In modern power systems, it is an important issue to process and analyze power big data and perform reliable decision-making analysis. In response to this problem, this paper proposes a distributed computing architecture for power data based on a consortium chain, which realizes distributed and trusted shared training computing for power data while taking into account the privacy protection of the original data. To solve the problem of sample imbalance, this paper proposes a data balancing method combining SMOTE algorithm and the k-means algorithm. This paper also proposes an LSTM neural network load forecasting method based on federated learning and proves that it has higher accuracy and applicability than traditional methods through examples.

Keywords: Power data · Federated learning · Blockchain · LSTM

1 Introduction

Modern power systems have built a series of advanced intelligent monitoring infrastructures. These devices can generate a large amount of data. Processing and analyzing these data can get a lot of effective information, to understand the operating status of power equipment and the power consumption of users. Through the analysis and research of historical data, the future electricity consumption data can be predicted, to make a series of decisions on the change of the operating state of power equipment. Smart grids are designed to save energy, reduce losses, and enhance grid reliability. This puts forward higher requirements for credible decision-making in the power system. How to process and analyze power big data and perform reliable decision-making analysis has become important for research problem. In the field of power systems, making full use of the value of existing data research, fusing machine learning models and artificial neural networks can provide accurate forecasts for power loads, and provide reference

F. Fang and F. Shu (Eds.): GameNets 2022, LNICST 457, pp. 73–82, 2022.
https://doi.org/10.1007/978-3-031-23141-4_6

and decision-making guidance for power generation, power sales, and power consumption. Real-time high-accuracy load forecasting will promote the development of credible decision-making and further promote the development of smart grids.

Blockchain technology has great advantages in terms of security. The characteristics of high data redundancy and resistance to tampering are very consistent with the strict requirements for data security in the power system. Each node in the blockchain network stores complete data, and the consistency of the data is ensured through a consensus mechanism. Blockchain technology was first applied in the field of digital encryption currency, and its high security based on cryptographic principles has been widely recognized, and then it has been introduced into more and more new application scenarios. In the energy field, blockchain technology has been tried to be applied in fields such as electricity trading, carbon emission rights trading, and more and more other businesses.

In modern power systems, in many cases, the data set cannot be centralized or shared between the two parties' data, and cannot be used for data mining and analysis, so it faces the problem of data islands. If the power data is centralized for analysis and utilization, it may also involve laws and regulations, user privacy, and data security issues. To solve this problem, this article introduces a federated learning method. The federated learning can ensure that the data can be trained locally without the data. This can protect the data security and user privacy of the edge nodes to a certain extent, and reduce the security risk in the data transmission process.

2 Related Work

The combination of blockchain and energy Internet of Things can promote the marketization of energy and the intelligentization of the grid system, which has huge development potential. Literature [1] analyzed the application applicability of blockchain in multi-energy systems and the information interconnection problems brought about by heterogeneous blockchains and proposed the construction of a multi-energy system transaction system based on heterogeneous blockchain technology. Necessity and method. Literature [2] proposes a blockchain system suitable for distributed photovoltaic microgrids, with the help of digital currency, photovoltaic transactions can be carried out without being monitored.

Regarding federated learning, in 2019, Google used the federated learning platform to train LSTM neural networks to learn vocabulary outside the vocabulary [3]. Zhao Tao et al. [4] proposed a federal learning aviation travel prediction method oriented to data privacy protection, which greatly improved the accuracy and reliability of aviation travel prediction.

Regarding power load forecasting, there have been decades of research in academia, and many effective forecasting methods have been proposed. In the 1970s and 1980s, researchers often used regression analysis, time series, and other forecasting methods. These methods are generally called classic forecasting methods. Literature [5] aimed at regression-based prediction models, using modern computing power to customize models for two to three years of hourly data to maximize prediction accuracy. Literature [6] expresses load forecasting as a functional time series problem and uses the function wavelet kernel method to predict the load curve of the clusters divided into groups.

With the rapid development of the energy Internet of Things, the amount of load data is increasing, and the demand for accuracy of power load forecasting is also increasing. In recent years, deep learning has made great progress. Therefore, the current research on short-term load forecasting Mainly focus on neural network. Literature [7] trains a neural network on multiple data sets generated by random sampling and replacement and then averages the results to reduce load forecast errors. In [8], to overcome the short-term load forecasting (IGRA-BA-BP) based on improved gray correlation and bat optimized neural network, to overcome the disadvantages of back-propagation (BP) neural network with poor generalization ability and easy to fall into local optimum. Method, the forecasting effect has been improved. Literature [9] established a cloud power load forecasting platform, based on the Markov chain to reduce forecasting errors from the bottom up.

3 Power Data Distributed Computing Architecture Based on Alliance Chain

As shown in Fig. 1, based on the existing consortium chain architecture of the power grid system, a distributed computing model and framework based on the consortium chain can be constructed to realize distributed and trusted shared training and computing of power data while taking into account the protection of original data privacy. The architecture consists of the data collection layer, the national network chain layer, the federation learning layer, and the algorithm application layer. The bottom layer is the data collection layer, which is mainly responsible for the collection of raw power data in various power parks, and then the collected data is Uploaded to the edge servers of the power grid, these edge servers constitute the working nodes of the alliance chain network of the upper state grid chain layer. Finally, through the design of smart contracts, each working node is organized to complete the uppermost federal learning application based on power data. Decision-making and analysis tasks are made by the level. In the process of specific algorithm application, taking into account the characteristics of power data sharing, this project introduces a data balance method combining SMOTE algorithm and K-means clustering algorithm into the data layer of the joint decision-making system to balance positive and negative samples. Use LSTM neural network to predict and analyze power load data to complete related decision-making tasks.

4 Data Credible Decision-Making Mechanism

4.1 Date Balance Method Combining SMOTE Algorithm and K-means Algorithm

Due to the robustness of the power system in practice, the system can often recover to a steady state by itself after being disturbed, and the probability of instability is relatively low, which brings the problem of sample imbalance to the method based on deep learning.

A typical oversampling method to solve the problem of data imbalance is the SMOTE algorithm, which aims to make up for the shortcomings of a small number of random oversampling. Random oversampling of the minority samples make the minority samples

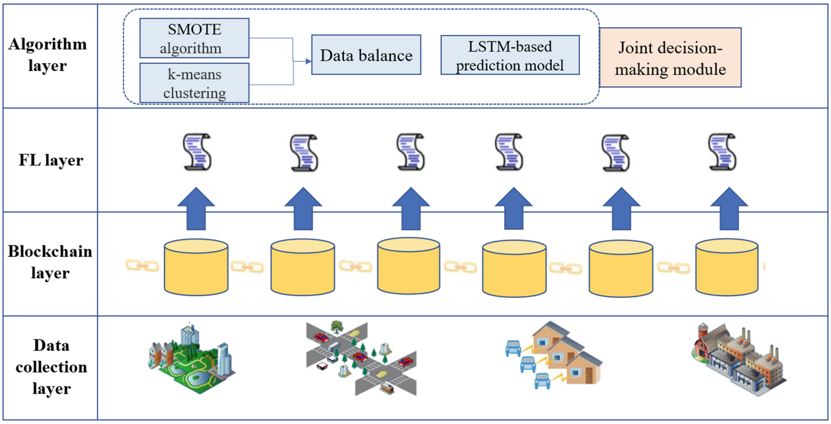

Fig. 1. Power data distributed computing architecture based on alliance chain

more recognizable, because the oversampling process is actually copying the samples. This copying will make the decision-making decisions oToes more and more rigorous and specific, leading to Classification overfitting. In order to solve this problem, SMOTE introduce synthetic data points and line segments adjacent to any 1 or all of k nearest neighbors of a minority class in the characteristic space. If (x_1, x_2) is an instance of a minority class, and if its nearest neighbor is selected as $\left(x_1{}', x_2'\right)$, then the data (X_1, X_2) is synthesized, namely:

$$(X_1, X_2) = (x_1, x_2) + \text{rand}(0, 1) \times \Delta \tag{4.1}$$

Among them, $\Delta = \left\{\left(x_1{}' - x_1\right), \left(x_2{}' - x_2\right)\right\}$, rand(0, 1) is a random number between 0 and 1. This technique broadens the decision-making area by generating artificial samples, because the samples added to the data set are located in the vicinity of the original samples, rather than the samples themselves. Compared with random oversampling with replacement, the decision area is more general.

The traditional SMOTE algorithm still has some problems. For example, the use of the SMOTE algorithm may blur the positive and negative class boundaries of the data set, which will increase the difficulty of training the classification model, and the SMOTE algorithm has a certain degree of blindness when processing data. In order to solve these problems. A clustering algorithm can be introduced to cluster the minority classes before oversampling, and then sample the clusters after clustering. This paper introduces the combination of K-means clustering algorithm and SMOTE algorithm. The K-means clustering algorithm takes the distance between the sample point and the cluster center as the optimization goal. According to the core idea of the clustering algorithm, the algorithm will maximize the similarity of the elements in each cluster as much as possible, and the difference between clusters The similarity is minimized. The K-means algorithm selects the desired clusters, through continuous iteration and recalculation of the cluster centers to minimize the variance within the entire cluster, and obtains relatively compact and independent clusters as the final goal of the algorithm.

Use the function method to obtain the extreme value, adjust the threshold of the number of iterations to obtain the best clustering effect.

The specific steps of the K-means algorithm are as follows:

STEP1 For data set D, randomly select k initial cluster centroid points as $\mu_1, \mu_2, \cdots, \mu_k \in D^n$.

STEP2 For the data except for the cluster center, calculate the Euclidean distance between them and $\mu_i(i = 1, 2, \ldots, k)$ one by one, and group the data closest to $\mu_i(i = 1, 2, \ldots, k)$ together so that all the data are divided into k categories. Can use formula (4.2) to classify

$$c^{(i)} = \arg \min_j \left\| x^{(i)} - \mu_j \right\|^2 \tag{4.2}$$

STEP3 Calculate the mean value of the data in various clusters and set the obtained mean value to the center of the new cluster. Then calculate the sum of the Euclidean distance from each data point in this cluster to the center of the cluster.

$$J(c, \mu) = \sum_{i=1}^{m} \left\| x^{(i)} - \mu_c(i) \right\|^2 \tag{4.3}$$

STEP4 Repeat the second and third steps. If the sum of Euclidean distance J does not change, output the clustering result.

Suppose a given data set D, where the majority class sample set is D_{max}, the minority class set is D_{min}, p is the number of minority class samples, and the sampling magnification is n. The specific description of the data balance method combining SMOTE algorithm and K-means algorithm is as follows:

STEP1 According to the specified parameter t of the given data set, randomly select t sample points in the data set (the selected sample points must belong to the minority class), and divide the minority class samples into t clusters $T_i(i = 1, 2, \ldots, t)$.

STEP2 Standardize the data in the data set and scale the data according to a certain ratio. After processing, the values of all attributes in the data set are in the interval (0,1).

STEP3 judges the number of most sample points in each cluster, that is $|T_i \cap D_{max}| = q$, if $q = |T_i|$, the cluster belongs to the noise cluster and the set is N'; if $q > |T_i|/2$, the cluster belongs to the boundary cluster and the set is B'; if $0 < q < |T_i|/2$, the cluster belongs to the safe cluster and the set is S'.

STEP4 removes the majority samples in the boundary clusters and only retains the minority samples. With each boundary cluster as the unit and the respective cluster center as the core, new sample points are generated according to formula (4.4).

$$\mathrm{Y_{new}} = \mathrm{c_i} + RAND(0, 1) \times \left(y_j - c_i\right) \tag{4.4}$$

STEP5 synthesizes the generated sample points into the minority samples to determine whether the overall data set is in balance. If it does not reach the balance, iterate 3 and 4 processes until the data set is close to balance, then the algorithm ends.

STEP6 reverse-standardizes the obtained data set, and converts the values in the interval (0,1) into the indicators of the original data according to the ratio of each attribute in step 2, and restores the attributes of the data in the original data set.

4.2 Load Forecasting Module Based on LSTM Neural Network

In order to deal with the problem of gradient disappearance that often occurs in RNN, Hochreiter et al. proposed a long short term memory (LSTM) neural network based on RNN. This is an effective non-linear cyclic neural network, which can take care of the time series and non-linear relationship of the data, and has emerged in load forecasting. This paper proposes a load forecasting method based on LSTM.

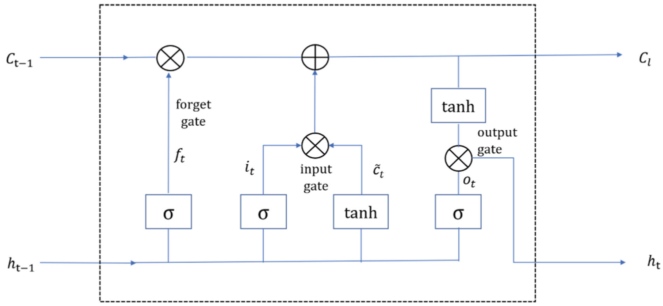

Fig. 2. Schematic diagram of LSTM structure

As shown in Fig. 2, in order to prevent the gradient extinction problem similar to RNN, LSTM adopts a gate structure to strengthen the information transmission between each neuron. It consists of three gate structures: input gate, output gate, and forget gate. Responsible for controlling the input, output and historical dependence of cells, and work together to realize load forecasting of power data. The specific operation process of LSTM is as follows

$$f_t = \sigma\left(W_{\mathrm{f}}\left[h_{t-1}, x_t\right] + b_{\mathrm{f}}\right) \tag{4.5}$$

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \tag{4.6}$$

$$\tilde{c}_t = \tanh\left(W_{\mathrm{C}}\left[h_{t-1}, x_t\right] + b_{\mathrm{C}}\right) \tag{4.7}$$

$$c_t = f_t c_{t-1} + i_t \tilde{c}_t \tag{4.8}$$

$$o_t = \sigma\left(W_{\mathrm{o}}\left[h_{t-1}, x_t\right] + b_{\mathrm{o}}\right) \tag{4.9}$$

$$h_t = \sigma_t \tanh c_t \tag{4.10}$$

Equation (4.5) controls the information that the current neuron forgets in the previous neuron, which is realized by the Sigmoid layer of the forget gate. Operate by reading the output h_{t-1} of the previous neuron and the input x_t of the current neuron to output a value in the interval [0,1], where 1 represents the complete memory of the previous neuron state, and 0 represents the previous neuron state. The neuron state is completely forgotten, and then multiplied by the previous neuron state c_{t-1}. In this way, the long-term memory of the neural network is guaranteed.

Equations (4.6) and (4.7) jointly control the input of neurons. Equation (4.6) is realized by the sigmoid layer of the input gate. The input gate reads the output h_{t-1} of the previous neuron and the input x_t of the neuron outputs a value i_t in the interval [0,1]. Equation (4.7) generates a candidate value vector $\tilde{c}_t$ through the tanh layer, and then obtains the neuron state c_t through (4.8).

Equation (4.10) gets the final output. After calculating the information retained by the previous neuron and the information of the neuron, the sigmoid layer of the output gate will get the final output. The neuron state c_t is processed through the tanh layer, and then multiplied by the output gate σ_t to obtain the final output h_t.

For the current predicted moment t, the power data from moment $t-n$ to moment $t-1$ is used as model input X, and the power data from moment t is used as output y, as shown below

$$X_t = \left[d_{t-n}, d_{t-n+1}, \ldots d_{t-1}\right] \tag{4.11}$$

$$y_t = d_t \tag{4.12}$$

where, d represents power data.

Federated learning solves the problem of data silos. The model training and sharing through federated learning mainly consists of two parts: organization model training and cloud model aggregation. The training objectives of cloud model and each node mechanism model can be expressed as:

$$\arg \min_{\omega,b} L = \sum_{i=1}^{n} l(y_i, \mathrm{f_S}(\boldsymbol{x}_\mathrm{i})) \tag{4.13}$$

$$\arg \min_{\omega^j,b^j} L_j = \sum_{i=1}^{n^j} l\left(y_i^j, f_j\left(\boldsymbol{x}_i^j\right)\right) \tag{4.14}$$

wherein, ω and b represent the training parameters: weight and deviation, L represents the loss function, $(\boldsymbol{x}_i, y_i)$ and $\left(\boldsymbol{x}_i^j, y_i^j\right)$ represent the global power data and the power data of the JTH node mechanism, and n represents the data set size.

The algorithm flow of federal learning and training LSTM neural network model is as follows:

Algorithm 1 Load prediction model training process

Input: common data set D_0, different mechanism data set $\{Q_1, Q_2, \ldots, Q_j\}$
Output: Final model F
1: LSTM model f_s is trained by using D_0 in cloud server
2: FOR round = 1, 2,... ,r DO
3: Send the model f_s to all mechanism nodes
4: Each node uses local data D_j to train its own local model f_j and upload it to the cloud server
5: The cloud server aggregates all model parameters and updates the global model
6: END FOR

5 Performance Evaluation

Footnotes should be avoided whenever possible. If required they should be used only for brief notes that do not fit conveniently into the text.

5.1 Data Set

The power load data of Slovenia from 2020-11-01 to 2021-11-01 was downloaded from the ENTSO website as an experimental data set. The data set consists of power load sampling every 1 h, totaling 8784 (24 × 366) pieces of power data (Fig. 3).

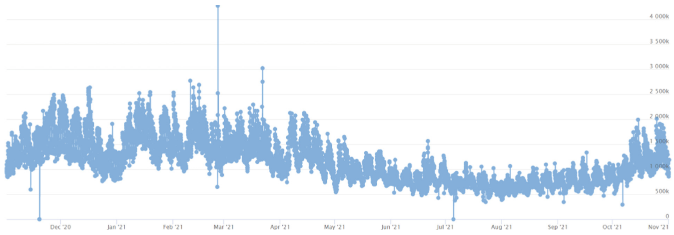

Fig. 3. Data set

5.2 Model Structure Design

The experimental data is divided into two parts. The first 51 weeks are used as the data training set, and the load data on October 30 and 31 are used as the test set to test the effect of the model in comparison with the predicted results. Input the training data into RNN and LSTM models for training, the number of hidden layer neurons (num_units) is designed to be 128, the training data batch (batch_size) is 16, the number of data in each batch (window_size) is 400, and the learning rate (AdamOptimizer) is 0.001, and the number of iterations (train_steps) is 3000.

5.3 Result Analysis

The average percentage error is used to compare the training effects of the two methods. The average percentage error is defined as (Figs. 4 and 5)

$$E_{\text{MAPE}} = 100\frac{\sum_{i=1}^{n}\left|\frac{L_i - L_i{}'}{L_i}\right|}{n} \tag{5-1}$$

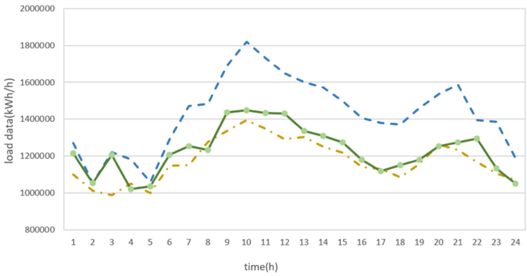

Fig. 4. Comparison of 2021.10.30 and real data

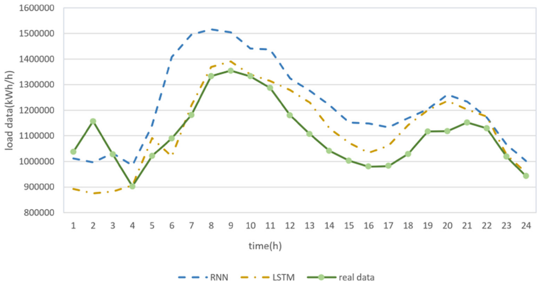

Fig. 5. Comparison of 2021.10.31 with real data

where L_i and $L_i{}'$ are the real value and the predicted value respectively.

Calculate E_{MAPE} for each method, as shown in Table 1.

Table 1. Comparison of E_{MAPE} between the two methods

Method\Data	10.30	10.31
RNN	15.88	11.92
LSTM	5.12	6.90

It can be clearly seen from the above table that the average percentage error of the prediction results of the LSTM neural network is much better than that of the RNN network. This is because the RNN model can only learn the diurnal variation of power data, and the LSTM model can learn better Daily and weekly changes. This proves that the LSTM method is better than the traditional method in the use effect.

Acknowledgments. This paper is supported by the science and technology project of State Grid Jibei Information & Telecommunication Company "Trusted Sharing Technology and Application of Winter Olympic Power Data Based on Blockchain" (52018E20008J).

References

1. Li, B., et al.: Transaction system and key technologies of multi-energy system based on heterogeneous blockchain. Autom. Electr. Power Syst. **42**(4), 183–193 (2018)
2. Johnson, L., Isam, A., Gogerty, N., Zitoli, J.: Connecting the Blockchain to the Sun to Save the Planet. Available at SSRN 2702639 (2015)
3. Beaufays, F.S., Chen, M., Mathews, R., Ouyang, T.: Federated learning of out-of-vocabulary words (2019)
4. Chen, T., Guo, R., Liu, Z.: Research on aviation application model of federated learning algorithm for big data privacy protection. Inf. Secur. Commun. Secrecy **09**, 75–84 (2020)
5. Wang, P., Liu, B., Hong, T.: Electric load forecasting with recency effect: a big data approach. Int. J. Forecast. **32**(3), 585–597 (2016)
6. Chaouch, M.: Clustering-based improvement of nonparametric functional time series forecasting: application to intra-day household-level load curves. IEEE Trans. Smart Grid **5**(1), 411–419 (2013)
7. Khwaja, A.S., Naeem, M., Anpalagan, A., Venetsanopoulos, A., Venkatesh, B.: Improved short-term load forecasting using bagged neural networks. Electr. Power Syst. Res. **125**, 109–115 (2015)
8. Wu, Y., Lei, J., Bao, L., Li, C.: Short-term load forecasting based on improved grey relational analysis and neural network optimized by Bat algorithm. Autom. Electr. Power Syst. **42**(20), 67–72 (2018)
9. Gong, X., Cardenas-Barrera, J.L., Castillo-Guerra, E., Cao, B., Saleh, S.A., Chang, L.: Bottom-up load forecasting with Markov-based error reduction method for aggregated domestic electric water heaters. IEEE Trans. Ind. Appl. **55**(6), 6401–6413 (2019)