Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 2, pp. 630–640, DOI 10.12694/scpe.v26i2.3893

POWER DATA ANALYSIS AND PRIVACY PROTECTION BASED ON FEDERATED LEARNING

YAJIE LI,[*] SHUTING CHEN,[†] XINMIAO HU [‡], SEN XU [§], AND MAO FAN[¶]

Abstract. In order for active distribution network operators to carry out power business such as load forecasting without meter reading rights, the author proposes a research on power data analysis and privacy protection based on federated learning. The author proposes a federated learning load forecasting framework for industry user data protection by selecting weather and time factors as the correlation factors of load. On this basis, the author constructed an industry user dataset and established a load forecasting model based on Long Short Term Time Series Network (LSTNet). At the same time, the FedML framework was used to establish a sub industry load forecasting framework based on federated learning. The results indicate that: The accuracy of the industry specific load forecasting framework based on federated learning proposed by the author is less than 9 p.u., and the theoretical maximum value of SMAPE is 210 p.u., indicating that this method has universality and universality and can be applied in different industries. The training method of this scheme is parallel, although it increases the interaction time by 1 minute, the interaction time accounts for a smaller proportion compared to the training time, and the time consumption is interaction time (1 minute)+single training time (94 minutes). Conclusion: The method can enable users in the same industry to conduct federated training without sharing load data, and support active distribution network operators in related business operations while protecting user electricity privacy. It has better predictive performance, fewer model numbers, and shorter time consumption.

Key words: Long short-term time series network, Load forecasting, Federated learning, FedML framework, Privacy protection

1. Introduction.

With the increasing number of distributed power generation resources such as renewable energy and the extensive integration of various intelligent terminal devices such as smart homes, a large amount of data flow will be generated between power grid enterprises and power users, as well as between electrical equipment and control centers. The smart grid has generated an unprecedented amount of raw information that can accurately assess situational awareness, improve the intelligence, efficiency, and sustainability of multiple industrial systems [1]. Researchers generally believe that the true value of smart grids lies not in the physical interconnected devices themselves, but in the vast amount of crude and unrefined information they contain, as well as how to efficiently, quickly, and meaningfully process this information. Therefore, in recent years, the analysis and processing of data in smart grids have received widespread attention, among which data privacy protection and anomaly detection have always been hot and difficult research topics [2]. On the one hand, due to the collection, transmission, and processing of massive data, frequent communication among various participants in the smart grid has led to increasingly serious data privacy issues, such as sensitive data directly exposing user privacy information [3]. Therefore, privacy protection technology is needed to process smart grid data to ensure the privacy and security of data in the smart grid and promote the practical development of smart grid applications.

On the other hand, the problem of false data injection, which has not been effectively solved in traditional power grids, has become more serious in the information-based and digital smart grid environment. More smart meters and other devices provide malicious users with more opportunities to modify power data. Due to

[*] State Grid Xinjiang Electric Power Co., Ltd. Information and Communication Company, Urumqi, Xinjiang, 832000, China. Xinjiang Energy Internet Big Data Laboratory, Urumqi, Xinjiang, 832000, China (Corresponding author, YajieLi9@163.com)

[†] State Grid Xinjiang Electric Power Co., Ltd. Information and Communication Company, Urumqi, Xinjiang, 832000, China. Xinjiang Energy Internet Big Data Laboratory, Urumqi, Xinjiang, 832000, China (ShutingChen3@126.com)

[‡] State Grid Xinjiang Electric Power Co., Ltd. Information and Communication Company, Urumqi, Xinjiang, 832000, China. Xinjiang Energy Internet Big Data Laboratory, Urumqi, Xinjiang, 832000, China (XinmiaoHu8@163.com)

[§] Xinjiang Energy Internet Big Data Laboratory, Urumqi, Xinjiang, 832000, China. State Grid Xinjiang Electric Power Co., Ltd, Urumqi, Xinjiang, 832000, China (SenXu7@126.com)

[¶] Xinjiang Energy Internet Big Data Laboratory, Urumqi, Xinjiang, 832000, China. State Grid Xinjiang Electric Power Co., Ltd, Urumqi, Xinjiang, 832000, China (MaoFan738@163.com)

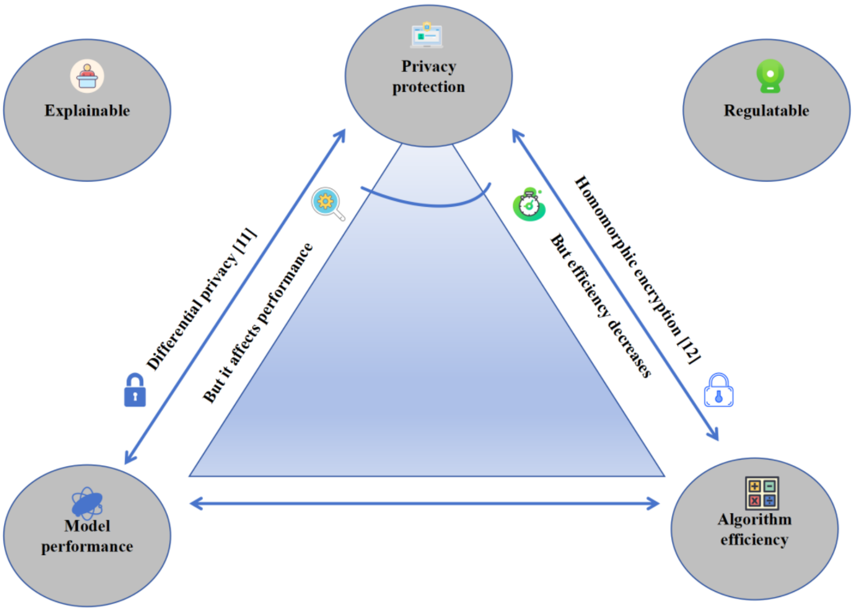

Fig. 1.1: Power Data Analysis for Federated Learning

the fact that the power supply of smart grids is generally provided based on real-time electricity consumption data [4]. If real-time electricity consumption data is not accurate, resulting in a mismatch between electricity supply and actual electricity consumption, it will not only disrupt the normal power supply order, bring huge economic losses to power companies and legitimate users, but also cause damage to the stability of the power grid. Therefore, how to effectively detect abnormal data in smart grids is an important research direction at present. As shown in Figure 1.1:

2. Literature Review. For anomaly detection of electricity consumption data in smart grids, researchers are committed to improving detection models to obtain more efficient and accurate model performance, such as from ordinary neural network models to convolutional neural network models, and then to XGBoost (Xtreme Gradient Boosting), hybrid models, etc. [5]. Truong, N uses the XGBoost model to detect normal and abnormal users, which is based on multidimensional smart meter data such as electricity consumption data, longitude and latitude data, and communication protocol data of instruments. The final accuracy is as high as 91% [6]. Ma, C. proposed a hybrid neural network model that combines convolutional and fully connected methods, specifically combining convolutional computation as a wide component and fully connected as a deep component to detect abnormal users, resulting in an AUC of up to 78% [7].

Li, Z attempted to use feedforward neural networks (FFN) to implement privacy protected abnormal electricity consumption data detection based on inner product function encryption, which is one of the function encryption methods. In addition, the scheme also achieves dynamic billing and load monitoring, without the need to learn fine-grained power consumption readings [8]. However, this scheme currently does not support the combination with convolutional neural networks. In addition, function encryption can only provide one ciphertext for a general function, which has the drawback of not being able to provide any number of keys for multiple general functions. This limits its ability to adapt to smart grid frameworks that are currently developing towards distributed frameworks.

Federated learning is a distributed machine learning environment, whose core concept is to model data without moving, make data available but not visible, and achieve collaborative joint modeling of data without leaving the local client [9]. Federated learning solves the problem of ”data silos” caused by data fragmentation and can fully utilize participant data, at the same time, the data does not leave the domain, protecting the data security of the participating parties and reducing the possibility of leakage at the original data level.

Liu, X. first proposed a user level differential privacy federated learning framework, which provides users with different privacy protections by calculating the upper bound of the sensitivity of any user to protect all data of any user. The purpose of differential privacy protection is to hide the contribution of individual users in model training, balancing privacy loss and model performance. Users who trust the server will use user level differential privacy. This way, users can safely join federated learning [10]. Mothukuri, V. proposed a differential privacy federated learning protection method based on Gaussian mechanism, which can more accurately obtain privacy loss during model training [11]. Wang, Y. proposed a new framework NbAFL, a staged differential privacy federated learning model, which calculates gradients for each client, adjusts its variance, adds noise that follows a Gaussian distribution to prevent parameter information leakage, and meets the requirements of global DP. While protecting user privacy, the accuracy and convergence speed of the model are guaranteed. In addition, this method compares the convergence performance of the model under different pruning thresholds through pre training and selects the best pruning threshold [12]. Wang, R. proposed a federated learning protection method based on localized differential privacy mechanism, which mainly adds noise disturbance to user local data and passes it to the central server, improving model accuracy and reducing performance loss [13].

The author designed a federated learning load forecasting model for active distribution networks to protect industry user reading data. Industry user datasets and corresponding data preprocessing schemes were constructed according to industrial industry categories. Based on long and short time series networks (LSTNet), a customer local load forecasting model was established. The model parameters were aggregated using the Federated Average (FedAvg) algorithm, and a sub industry power load forecasting framework based on federated learning was established using the FedML framework. The industry load forecasting model was obtained while protecting user data privacy. The analysis results of the example show that this scheme has better predictive performance, fewer models, and shorter time consumption, and can obtain accurate predictive models without mastering user data.

3. Research Methods.

3.1. Load forecasting based on LSTNet.

3.1.1. Factors affecting load. The load curve can be decomposed into regular components, uncertain components, and noise components; The regular component is the periodic variation of load in the time dimension; The uncertain component is the non cyclical changes in load caused by factors such as weather and economy; The noise component refers to the impact of other factors that cannot be physically explained on the load [14]. Common weather factors include temperature, humidity, precipitation, and air pressure, which have a significant impact on short-term load changes; The time factor is reflected in the combined effect of short-term and long-term repetition patterns, mainly including the influence of workdays, holidays, and seasons.

3.1.2. Construction of Industry User Datasets. The author selects weather, load, and time as the characteristic types of input data for the load forecasting model. Weather includes temperature, relative humidity, precipitation, wind speed, and pressure to reflect the impact of weather factors on load; Time data includes holiday information, workday information, year, month, day, hour, and week information, used to reflect the periodicity of load, among them, holiday information is determined based on the statutory holidays in China in 2019 and 2020, and workday information includes workdays generated during the week due to compensatory leave. These constructed industry datasets only need to be retained locally by users and do not need to be provided to active distribution network operators, effectively protecting user data privacy.

3.1.3. Load forecasting model based on LSTNet. The author uses LSTNet as the load forecasting model, which consists of three parts: convolutional layer, loop layer, loop jump layer, fully connected layer, and autoregressive layer. LSTNet can use convolutional layers to extract short-term patterns and dependencies between variables from the input multivariate time series, and then use loop layers and loop jump layers to capture long-term and longer-term dependencies between variables [15]. The input of the model is a multivariate time series

$$X = \begin{bmatrix} x_0(1) & x_1(1) & \cdots & x_n(1) \\ x_0(2) & x_1(2) & \cdots & x_n(2) \\ \cdots & \cdots & & \cdots \\ x_0(m) & x_1(m) & \cdots & x_n(m) \end{bmatrix}$$

where n represents the number of features of the multivariate time series, where there are a total of 27 features including load, weather, and time; M represents the length of the input time series.

The weight coefficients corresponding to the convolutional layer are W_C and b_C, the weight coefficients corresponding to the cyclic layer are W_R and b_R, the weight coefficients corresponding to the cyclic jump layer are and , the bias of the fully connected layer is b_D, the weight coefficients corresponding to the autoregressive layer are W_{AR} and b_{AR}, respectively. Record the weight coefficients $W = \{W_C, W_R, W_S, W_{AR}\}$ and $b = \{b_C, b_R, b_S, b_D, b_{AR}\}$ bias of each layer in LSTNet as equation 3.1:

$$\omega \in \{W, b\} \tag{3.1}$$

In the formula, ω represents the weight coefficients and biases of each layer of neurons, i.e. the model parameters of LSTNet.

3.2. Industry specific load forecasting framework based on federated learning.

3.2.1. Overview of Federated Learning. Federated learning is a machine learning method based on distributed datasets, first proposed by Google's McMahan. Federated learning includes two processes: Model training and model inference. During the model training process, multiple clients train locally with their own datasets, and the server collects model parameters to update the global model. Each client continues to train based on this update, ultimately obtaining a model that can be shared by multiple parties [16]. Model inference is the application of trained federated learning models to new data. Federated learning is divided into horizontal federated learning, vertical federated learning, and federated transfer learning based on the overlap between feature space and sample space of data from different clients [17]. Among them, horizontal federated learning is suitable for situations where the client has overlapping feature spaces but different sample spaces. The applicable conditions for horizontal federated learning are shown in equation 3.2.

$$\begin{cases} X_i = X_j \\ I_i \neq I_j \end{cases} \quad \forall D_i, D_j, i \neq j \tag{3.2}$$

In the formula, D_i and D_j represent the datasets of client i and client j, respectively; X_i and X_j represent the feature spaces of client i and client j, respectively;I_i and I_j are the sample spaces for client i and client j, respectively.

In the author's application scenario, different power end-users in the same industry each have their own dataset, with different samples in the dataset, but the corresponding features of the samples are similar and the same (load shape, weather information, time information). If future users refuse to share their meter reading data from a privacy protection perspective, active distribution network operators can use the author's proposed federated learning algorithm framework for industry user load forecasting, users only need to train their own load forecasting model locally using their own load data, and share the load forecasting model parameters with active distribution network operators instead of specific user reading load data. Active distribution network operators use the FedAvg algorithm to aggregate these model parameters, and use the FedML framework to establish a sub industry power load forecasting framework based on federated learning. Under the premise of protecting user data privacy, they obtain the industry's load forecasting model.

3.2.2. Federated training process based on FedAvg algorithm. Federated learning is based on the FedAvg algorithm to obtain a global model. The FedAvg algorithm is a horizontal federated optimization algorithm proposed by McMahan, which runs gradient descent algorithm in parallel on multiple user side clients. In each round of interaction, the central server collects parameters for aggregation and sends them back to the client for further training.

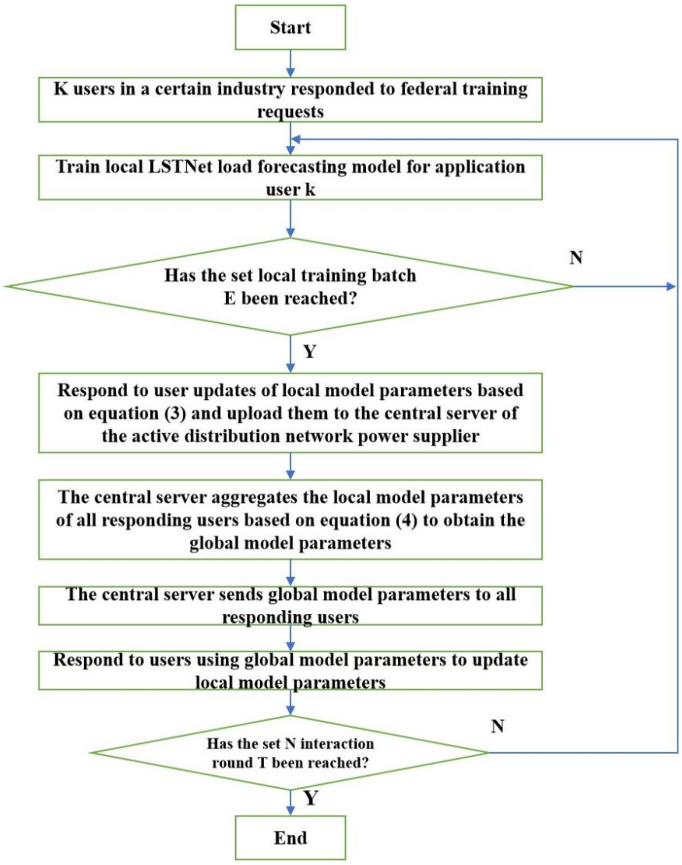
Start
K users in a certain industry responded to federal training requests
Train local LSTNet load forecasting model for application user k
Has the set local training batch E been reached?
N
Y
Respond to user updates of local model parameters based on equation (3) and upload them to the central server of the active distribution network power supplier
The central server aggregates the local model parameters of all responding users based on equation (4) to obtain the global model parameters
The central server sends global model parameters to all responding users
Respond to users using global model parameters to update local model parameters
Has the set N interaction round T been reached?
N
Y
End

Fig. 3.1: Federated training flowchart based on FedAvg algorithm

The users participating in federated learning are responsive users, and the model parameters are the weight coefficients or biases of neurons in each layer of LSTNet. The active distribution network operator/power supply company sets up a central server to aggregate local model parameters uploaded by response users. Response users use local datasets for local training based on LSTNet load forecasting models, and do not share datasets with each other. After the local training reaches the set training batch, the corresponding user uploads the local model parameters to the central server. After collecting the local model parameters uploaded by all responding users participating in federated training, the central server generates global model parameters based on the FedAvg algorithm and sends them back to each responding user. Respond to users updating their local model parameters with global model parameters and continue training, repeating this process until the set interaction round is reached. The specific process is shown in Figure 3.2.

Firstly, the central server located in the active distribution network operator sends the initialized model parameters to ω_0 to all responding users. After the user receives the model parameters, the LSTNet model is trained locally based on the gradient descent algorithm. In the t-th round of interaction, the application user k first performs local training, and the model parameters $\omega_{k,t}$ after training are shown in equation 3.3.

$$\omega_{k,t} = \omega_{k,t-1} - \eta \nabla f(\omega_{k,t-1}) \tag{3.3}$$

In the formula, $\omega_{k,t}$ represents the weight coefficients or biases of neurons in each layer of LSTNet in response to user k's completion of local training in the t-th round of interaction; η is the learning rate; ∇f represents gradient descent.

Then, each responding user will input the trained model parameters from equation 3.3 ωk. Send to the central server located in the active distribution network operator/power supply company. The central server collects model parameters updated by all responding users ωk. After t, aggregate the parameters according to equation 3.4 to generate global parameters [18].

$$\omega_t = \sum_{k=1}^{K} \frac{n_k}{n} \omega_{k,t} \tag{3.4}$$

In the formula, n_k represents the number of samples in response to user k; N is the total number of samples for all responding users; K is the total number of responsive users. The central server sends the global parameter ω_t to all responding users, who update the local parameters based on the global parameters and continue training until all interaction rounds are completed, that is

$$\omega_{k,t} = \omega_t \tag{3.5}$$

3.2.3. Industry specific load forecasting framework based on federated learning. The main body of the industry specific load forecasting framework based on federated learning includes active distribution network operators/power supply companies, central servers, users, and electricity sales companies. The entire process is divided into the following 7 steps:

Step 1: The active distribution network operator/power supply company sends a training request to the corresponding users in a certain target industry.

Step 2: Considering that users need to complete training locally, they can decide whether to participate in federated training. The participating users are response users who preprocess their own dataset locally and input it into the LSTNet load forecasting model for local training. After completing one round of training, the neuron parameters of each layer of LSTNet are passed to the central server [19].

Step 3: The central server aggregates local model parameters passed by response users based on the FedAvg algorithm to generate global model parameters, and passes the global model parameters to the response users. After multiple interactions, the central server obtains the industry global model.

Step 4: The central server transfers the industry global model to the active distribution network operator/power supply company.

Step 5: The active distribution network operator/power supply company distributes corresponding rewards based on the contribution level of the responding users to the overall industry model.

Step 6: The electricity sales company submits a model demand application to the active distribution network operator/power supply company based on the required industry.

Step 7: Actively return the required industry global model to the distribution network operator/power supply company and obtain profits.

3.3. Example analysis. The author verifies the effectiveness of the proposed scheme through numerical examples. The experimental environment used is described as follows: CPU is 3.70 GHz Intel Core i7-8700K, GPU is NVIDIA TITAN XP 12 GB, memory is 30 GB, operating system is Ubuntu 18.04 LTS, Python version is 1.7.2, CUDA version is 11.3, Python version is 3.8, federated learning is implemented based on FedML library, federated optimization algorithm is FedAvg algorithm, and single machine deployment mode is adopted [20]. The parameter settings for the LSTNet model used for load forecasting are shown in Table 3.1.

3.3.1. Dataset Description. The dataset selected for the calculation is a 730 day multivariate feature dataset classified by industry in a certain city from January 1, 2021 to December 30, 2022, with a time granularity of 48 points per day. Randomly select three responsive users from three industries: pharmaceutical manufacturing, food manufacturing, and rubber and plastic products in the constructed dataset. Each responsive user dataset is divided into training, validation, and testing sets in a ratio of 8:2:2 (corresponding to 584, 74, and 74 d, respectively).

3.3.2. Evaluation indicators. In order to measure the predictive performance of the model, it is necessary to use evaluation indicators to calculate the prediction error of the model. Common evaluation indicators include mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE), and symmetric mean absolute percentage error (SMAPE) [21]. The author selected RMSE and SMAPE (their

Table 3.1: LSTNet model parameters

Parameter Name	Parameter values
The number of hidden units in convolutional layers	130
Convolutional layer convolution kernel width	7
The length of convolutional kernels in convolutional layers	28
Number of hidden units in the loop layer	129
Loop - skip layer hidden unit count	129
Loop skip layer skip hidden unit count	50
Learning rate	0.002
Loss rate	0.3
Number of samples in a batch	65
Training batch	210 times

Table 4.1: Federated Learning Training Scenario Settings

Scene	Interaction round/time	Local training batch/time	Is it divided by industry
1	15	20	yes
2	15	25	yes
3	20	15	yes
4	25	15	yes
5	25	15	no

values are λ RMSE and λ SMAPE, as an evaluation indicator for short-term load forecasting results, is expressed in equations 3.6 and 3.7, respectively.

$$\lambda_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y_i} - y_i)^2} \tag{3.6}$$

$$\lambda_{SMAPE}\& = \frac{200}{N}\sum_{i=1}^{N}\frac{|\hat{y_i} - y_i|}{|\hat{y_i}| + |y_i|} \tag{3.7}$$

In the formula: N represents the total number of samples; $\hat{y_i}$ and y_i are the predicted and actual values at time i, respectively. RMSE amplifies the difference between larger and smaller errors using the square term, making it more sensitive to data with larger prediction deviations. A smaller RMSE indicates better model prediction accuracy. SMAPE is a revised version of MAPE, which solves the problem of MAPE being unable to calculate when the actual value is 0 and MAPE punishing negative errors more than positive errors. The range of SMAPE values is [0210], and a smaller SMAPE indicates better predictive performance.

4. Result analysis.

4.1. Analysis of Federated Learning Optimization Results. This section sets up federated learning training scenarios based on Table 4.1. Among them, the interaction round refers to the number of times the central server in each industry updates global parameters to each responding user, while the local training batch refers to the number of local iterations for each responding user. The load forecasting model based on industry electricity consumption characteristics is trained on an industry basis. This section sets up three industries, namely pharmaceutical manufacturing, food manufacturing, rubber and plastic products. Each industry uses three datasets that respond to users. The FedAvg algorithm is used to capture the common electricity consumption characteristics of a single industry and obtain the final industry load forecasting model [22].

According to Tables 4.2 and 4.3, it can be seen that Scenario 4 has the best predictive performance. Comparing scenarios 1 to 5, it can be seen that under the same total number of training iterations, the

Table 4.2: Comparison of RMSE for Predictive Accuracy of Federated Learning in Different Scenarios

	RMSE/kW		
Scene	pharmaceutical industry	Food manufacturing industry	Rubber and plastic products industry
1	131.57	79.56	162.58
2	130.48	80.79	166.43
3	129.78	80.02	167.36
4	127.12	76.55	160.78
5	131.79	84.37	178.34

Table 4.3: Comparison of SMAPE prediction accuracy for federated learning in different scenarios

	SMAPE/p.u.		
Scene	Pharmaceutical industry	Food manufacturing industry	Rubber and plastic products industry
1	3.58	5.68	4.53
2	3.57	5.85	4.52
3	3.55	5.72	4.65
4	3.47	5.45	4.38
5	3.58	5.99	4.89

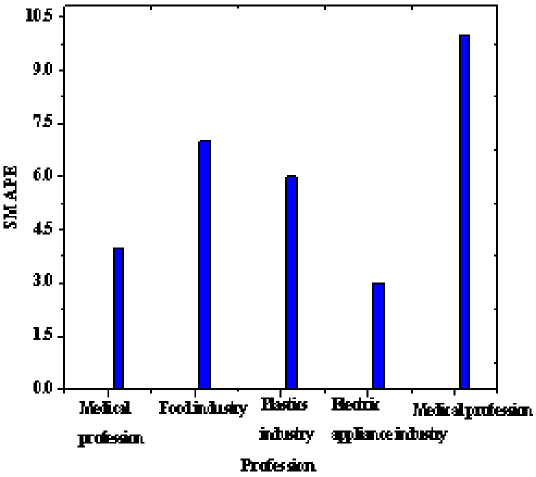

Fig. 4.1: Comparison of SMAPE prediction accuracy in different industries

interaction round has a greater impact on the performance of load forecasting than the local training batch. Comparing Scenario 4 and Scenario 5, it can be seen that the scenario based on different industries has better predictive performance than the scenario based on no industry, as it captures the electricity consumption characteristics of the industry.

Apply scenario 4 to more industries to test the generalization and adaptability of this method in different industries. The prediction accuracy SMAPE under different industries is shown in Figure 4.1.

From Figure 4.1, it can be seen that the author's proposed federated learning based industry specific load forecasting framework has a prediction accuracy SMAPE value of less than 9 p.u. in different industries,

Table 4.4: Comparison of RMSE prediction accuracy between this scheme and scheme 1

Programme	RMSE/kW		
	pharmaceutical industry	Food manufacturing industry	Rubber and plastic products industry
Option 1	126.59	79.56	158.83
Author's Proposal (Scenario 4)	127.11	76.55	160.74

Table 4.5: Comparison of RMSE prediction accuracy between this scheme and scheme 1

Programme	SMAPE/p.u.		
	pharmaceutical industry	Food manufacturing industry	Rubber and plastic products industry
Option 1	3.45	5.68	4.38
Author's Proposal (Scenario 4)	3.45	5.45	4.37

while the theoretical maximum value of SMAPE is 210 p.u., indicating that the method has universality and universality and can be applied in different industries.

4.2. Industry specific prediction plan. In industry specific prediction schemes that do not consider privacy protection, all users are required to upload raw table reading data. The smart meter uploads the user's local meter reading data to the server of the active distribution network operator/power supply company through a gateway, and accumulates it by industry. The active distribution network operator/power supply company establishes an industry load forecasting model based on the accumulated data of all users [23]. This section takes Scenario 4 in Table 4.1 as an example to compare and analyze it with industry specific prediction schemes that do not consider privacy protection. For the convenience of expression, scheme 1 in the following text refers to industry specific prediction schemes that do not consider privacy protection.

Tables 4.4 and 4.5 show the comparison of prediction accuracy between the proposed federated learning scheme and scheme 1 by the author. It can be seen that in predicting the cumulative load of industry users, the performance of this scheme is similar to that of scheme 1, indicating that the performance of the proposed privacy protection based federated learning prediction model for power load in different industries is similar to that of traditional independent prediction schemes without considering privacy protection. However, this scheme does not require uploading original meter reading data to active distribution network operators/power supply companies, effectively protecting user privacy.

4.3. Independent prediction scheme for individual users. In a user independent prediction scheme that does not consider privacy protection, users need to upload raw meter reading data, and the active distribution network operator/power supply company establishes different prediction models for each user's meter reading data. This section takes scenario 4 in Table 4.1 as an example to compare and analyze it with the sub user independent prediction scheme that does not consider privacy protection [24]. For the convenience of expression, scheme 2 in the following text refers to a sub user independent prediction scheme that does not consider privacy protection.

The accuracy of this scheme is slightly inferior to scheme 2, but it considers privacy protection and only requires the establishment of three industry prediction models, which has advantages in computing speed and memory usage.

4.4. Comprehensive comparative analysis. Table 4.6 provides a comprehensive comparison between this scheme and the scheme that does not consider privacy protection.

In terms of predictive performance, this scheme is similar to scheme 1 but slightly inferior to scheme 2. In terms of the number of models, this scheme is the same as scheme 1, both representing the number of industry users. However, compared to scheme 2 (where the number of models and users is the same), this scheme has a relatively fewer number of models, and its advantage is more pronounced as the number of users participating

Table 4.6: Comprehensive comparison between this scheme and traditional schemes

Programme	Do you consider privacy protection	pharmaceutical industry	SMAPE/p.u. Food manufacturing industry	Rubber and plastic products industry	Number of models /piece	Transfer data volume /MB	time consuming /min
Option 1	no	3.46	5.68	4.38	3	65.06	278
Option 2	no	3.56	5.48	4.32	8	65.07	838
Author's proposal	yes	3.48	5.45	4.37	4	1070	94

in training increases. In terms of data transmission, traditional solutions only require the transmission of user read table data, which is the number of users (8) × the number of features (27) × the number of days (730) × the number of load points within a day (49) × the size of floating-point numbers (8). In this scheme, each responding user needs to transmit local model parameters to the central server during each interaction round, with a data transmission amount of users (8) × the number of model parameters x the number of interaction rounds (20) × the size of floating-point numbers (8). The number of model parameters is very large, with only the convolutional layer, loop layer, and loop skip layer having 129 × 129 × 49 parameters . Due to the consideration of privacy protection, this scheme generates a large amount of communication data, and the transmission cost is higher compared to traditional schemes. In terms of time consumption, the average time for a single training session is 94 minutes. Scheme 1 requires the establishment of a model for each industry dataset, so the time consumption is the number of industries (3) × the time for a single training session (94 minutes); Option 2 requires the establishment of a model for each user's dataset, so the time required is the number of users (9) multiplied by the single training time (94 minutes); The training method of this scheme is parallel, although it increases the interaction time by 1 minute, the interaction time accounts for a smaller proportion compared to the training time, and the time consumption is interaction time (1 minute)+single training time (94 minutes). In summary, although this scheme increases the amount and cost of data transmission, it ensures the privacy of table reading data, maintains better predictive performance, fewer models, and shorter time consumption.

5. Conclusion. The author proposes a sub industry power load forecasting framework based on federated learning from the perspective of data protection for meter reading. Firstly, select weather, economic, and time factors as the main influencing factors of load to construct a dataset; Then, establish a load forecasting model based on LSTNet; Finally, a sub industry load forecasting framework based on federated learning was established using the FedAvg algorithm and the FedML framework. Case analysis shows that active distribution network operators can obtain predictive models without knowing user data. Although it increases the amount of transmitted data, it has better predictive performance, fewer models, and shorter time consumption.

Federated learning has high requirements for communication, and in situations where the required global model size is large, network bandwidth limitations and the number of working nodes may exacerbate the communication bottleneck of federated learning. Subsequently, feature filtering can be performed on weather features to determine the optimal weather features to reduce irrelevant meteorological interference, reduce the amount of data transmitted by the model, improve prediction accuracy, and enhance the generalization ability of the prediction model. At the same time, further consideration will be given to quantifying and certifying the contribution of users to the model. Rewards will be distributed based on the contribution of responsive users to the prediction model. Economic measures will be used to encourage more users to actively participate in federated learning, resulting in economic benefits for users. Active distribution network operators will have a more accurate understanding of industry load forecasting models, forming a virtuous cycle.

REFERENCES

[1] Wei, K., Li, J., Ding, M., Ma, C., Su, H., Zhang, B., & Poor, H. V. (2021). User-level privacy-preserving federated learning: Analysis and performance optimization. IEEE Transactions on Mobile Computing, 21(9), 3388-3401.

[2] Wei, K., Li, J., Ding, M., Ma, C., Yang, H. H., Farokhi, F., ... & Poor, H. V. (2020). Federated learning with differential privacy: Algorithms and performance analysis. IEEE transactions on information forensics and security, 15, 3454-3469.

[3] Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., ... & He, B. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. IEEE Transactions on Knowledge and Data Engineering, 35(4), 3347-3366.

[4] Li, Y., Wang, R., Li, Y., Zhang, M., & Long, C. (2023). Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach. Applied Energy, 329, 120291.

[5] Liu, D., & Simeone, O. (2020). Privacy for free: Wireless federated learning via uncoded transmission with adaptive power control. IEEE Journal on Selected Areas in Communications, 39(1), 170-185.

[6] Truong, N., Sun, K., Wang, S., Guitton, F., & Guo, Y. (2021). Privacy preservation in federated learning: An insightful survey from the GDPR perspective. Computers & Security, 110, 102402.

[7] Ma, C., Li, J., Ding, M., Yang, H. H., Shu, F., Quek, T. Q., & Poor, H. V. (2020). On safeguarding privacy and security in the framework of federated learning. IEEE network, 34(4), 242-248.

[8] Li, Z., Sharma, V., & Mohanty, S. P. (2020). Preserving data privacy via federated learning: Challenges and solutions. IEEE Consumer Electronics Magazine, 9(3), 8-16.

[9] Jia, B., Zhang, X., Liu, J., Zhang, Y., Huang, K., & Liang, Y. (2021). Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT. IEEE Transactions on Industrial Informatics, 18(6), 4049-4058.

[10] Liu, X., Li, H., Xu, G., Chen, Z., Huang, X., & Lu, R. (2021). Privacy-enhanced federated learning against poisoning adversaries. IEEE Transactions on Information Forensics and Security, 16, 4574-4588.

[11] Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., & Srivastava, G. (2021). A survey on security and privacy of federated learning. Future Generation Computer Systems, 115, 619-640.

[12] Wang, Y., Bennani, I. L., Liu, X., Sun, M., & Zhou, Y. (2021). Electricity consumer characteristics identification: A federated learning approach. IEEE Transactions on Smart Grid, 12(4), 3637-3647.

[13] Wang, R., Lai, J., Zhang, Z., Li, X., Vijayakumar, P., & Karuppiah, M. (2022). Privacy-preserving federated learning for internet of medical things under edge computing. IEEE journal of biomedical and health informatics, 27(2), 854-865.

[14] Badr, M. M., Mahmoud, M. M., Fang, Y., Abdulaal, M., Aljohani, A. J., Alasmary, W., & Ibrahem, M. I. (2023). Privacy-preserving and communication-efficient energy prediction scheme based on federated learning for smart grids. IEEE Internet of Things Journal, 10(9), 7719-7736.

[15] Fang, C., Guo, Y., Wang, N., & Ju, A. (2020). Highly efficient federated learning with strong privacy preservation in cloud computing. Computers & Security, 96, 101889.

[16] Song, M., Wang, Z., Zhang, Z., Song, Y., Wang, Q., Ren, J., & Qi, H. (2020). Analyzing user-level privacy attack against federated learning. IEEE Journal on Selected Areas in Communications, 38(10), 2430-2444.

[17] Lu, Y., Huang, X., Dai, Y., Maharjan, S., & Zhang, Y. (2020). Federated learning for data privacy preservation in vehicular cyber-physical systems. IEEE Network, 34(3), 50-56.

[18] Zhou, C., Fu, A., Yu, S., Yang, W., Wang, H., & Zhang, Y. (2020). Privacy-preserving federated learning in fog computing. IEEE Internet of Things Journal, 7(11), 10782-10793.

[19] Lakhan, A., Mohammed, M. A., Nedoma, J., Martinek, R., Tiwari, P., Vidyarthi, A., ... & Wang, W. (2022). Federated-learning based privacy preservation and fraud-enabled blockchain IoMT system for healthcare. IEEE journal of biomedical and health informatics, 27(2), 664-672.

[20] Yin, L., Feng, J., Xun, H., Sun, Z., & Cheng, X. (2021). A privacy-preserving federated learning for multiparty data sharing in social IoTs. IEEE Transactions on Network Science and Engineering, 8(3), 2706-2718.

[21] Wen, M., Xie, R., Lu, K., Wang, L., & Zhang, K. (2021). Feddetect: A novel privacy-preserving federated learning framework for energy theft detection in smart grid. IEEE Internet of Things Journal, 9(8), 6069-6080.

[22] Thapa, C., Chamikara, M. A. P., & Camtepe, S. A. (2021). Advancements of federated learning towards privacy preservation: from federated learning to split learning. Federated Learning Systems: Towards Next-Generation AI, 79-109.

[23] Ali, M., Naeem, F., Tariq, M., & Kaddoum, G. (2022). Federated learning for privacy preservation in smart healthcare systems: A comprehensive survey. IEEE journal of biomedical and health informatics, 27(2), 778-789.

[24] Singh, S., Rathore, S., Alfarraj, O., Tolba, A., & Yoon, B. (2022). A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology. Future Generation Computer Systems, 129, 380-388.

Edited by: Bradha Madhavan
Special issue on: High-performance Computing Algorithms for Material Sciences
Received: May 10, 2024
Accepted: Jun 25, 2024