A Load Forecasting Model for Smart Grid Based on RF-BiLSTM and Federated Learning

Wenhui Li, 17799376253@163.com, Yancheng Institute Of Technology, Yancheng, China
Huilin Jiang, huilin.jiang@njxzc.edu.cn, NanJing XiaoZhuang University, Nanjing, China
Lei Zhang, zl_srd@njxzc.edu.cn, NanJing XiaoZhuang University, Nanjing, China

Abstract

Accurate load forecasting is essential for improving the efficiency and reliability of smart grids. Traditional forecasting models face challenges in integrating high-dimensional data and protecting privacy. To address these issues, a novel load forecasting model is presented in this paper. The proposed approach combines a random forest (RF) and a bi-directional long short-term memory (BiLSTM) network based on a federated learning framework. To effectively process high-dimensional heterogeneous data in smart grids, the proposed model exploits the feature extraction capability of radio frequency. In addition, the sequential data processing capability of BiLSTM provides a means of processing time-varying load data in smart grids. The federated learning method protects user privacy and achieves the goal of global model training adopting distributed data. Experimental results demonstrate that the proposed model is superior to traditional models in accuracy and efficiency.

Key words: smart grid, federated learning, load prediction, BiLSTM

Ⅰ. Introduction

Smart grids play a critical role in the evolution of power systems. Traditional power systems employ a top-down management model, suitable for stable and predictable power usage. Due to the intermittent nature of distributed energy sources and the limitations of the power infrastructure, it presents a challenge to maintain a balance between supply and demand[1]. Accurate load forecasting is therefore critical to solving this problem. As a result, many advanced technologies have been developed to optimize the matching of supply and demand requirements for further improving the power system stability. Such as support vector machines (SVMs), artificial neural networks and federated learning. SVMs algorithm effectively addresses machine learning challenges with small sample sizes and demonstrates robust performance on nonlinear and high-dimensional data. Nonetheless, it requires extended training times for large-scale sample data, and the ultimate prediction accuracy of the model significantly depends on the selection of model parameters.

Applying AI technologies in smart grids, researchers have developed data-driven models based on deep learning (DL)[2], which significantly improved predictive performance and adaptability over traditional methods[3]. For example, the recurrent neural network (RNN)-based household load forecasting model proposed by Shi et al.[4] solves the overfitting problem in the traditional DL method. The Long Short-Term Memory (LSTM) network proposed by Liu et al. effectively solves the gradient vanishing problem encountered by RNNs when dealing with long-term sequences[5]. The predictive performance of the model structure was further validated by Kong et al.[6] and Taik et al[7]. Alhussein et al combined the feature extraction capabilities of convolutional neural networks (CNNs) with the sequence processing advantages of LSTM to further improve forecasting accuracy[8]. To consider the time series dependence of the load itself, Xu et al.[9] and Chen et al.[10] proposed a hybrid deep learning model using BiLSTM. The field of smart grid load forecasting has been propelled forward by these studies.

These successful prediction methods enhance the research ideas in this paper. Such

methods require centralized upload of data to a server, potentially compromising users' privacy. Federated learning (FL), known for its privacy preserving and distributed computing features. To improve privacy, the researchers implemented federated learning methods in the smart grid. Venkataramanan et al. applied FL to distributed energy forecasting and achieved positive prediction results in real-world grid services[11]. Meanwhile, due to the privacy of electricity consumption data, Husnoo et al. proposed a FL model based on an improved differential privacy algorithm to improve the privacy preserving performance of load forecasting systems[12]. In addition, Qin et al. constructed a high-performance forecasting model using data resources from multiple buildings to verify the effectiveness of the method[13]. However, these studies did not fully consider the interactions between individual characteristics, such as privacy and personalisation. Although Zhang et al. proposed a strategy combining federated learning and privacy to facilitate short-term demand forecasting[14], the method still has limitations in dealing with model generalization capabilities and noisy data. On the other hand, data quality and distribution affect the prediction performance of federated learning algorithm.

In conclusion, to address the shortcomings of current research, this paper proposes a smart grid load forecasting model based on RF-BiLSTM with federated learning. The major contributions are as follows:

(1) RF is applied to extract features from high-dimensional data sets to overcome the problem of interference from incomplete and irrelevant data features in smart grids. To cope with different areas of smart grids with different features, BiLSTM is deployed to fill in the gaps and correct errors in the before and after data.

(2) A Federated Proximal (FedProx) algorithm is employed in smart grid federated learning. In smart grids, data from different nodes exhibit non-independent and identically distributed (Non-IID) patterns. Compared with Federated Averaging (FedAvg), FedProx better addresses the differences in data distribution among nodes, making it more suitable for smart grid datasets.

(3) The proposed model is validated on real data. The experimental results indicate that the method not only significantly improves the prediction accuracy, but also

protects the user's data.

The remainder of the paper is structured as follows: Section II describes the model structure. Section III outlines the specific steps of the prediction model and the algorithm used in the method. Section IV presents the experimental results. Section V summarizes the paper.

Ⅱ. System model

A. Bidirectional Long Short-Term Memory

The Long Short-Term Memory network is a variant of RNN[15]. As illustrated in Figure 1, it comprises three key components: the input gate, the forget gate, and the output gate, which are crucial for achieving data protection and control.

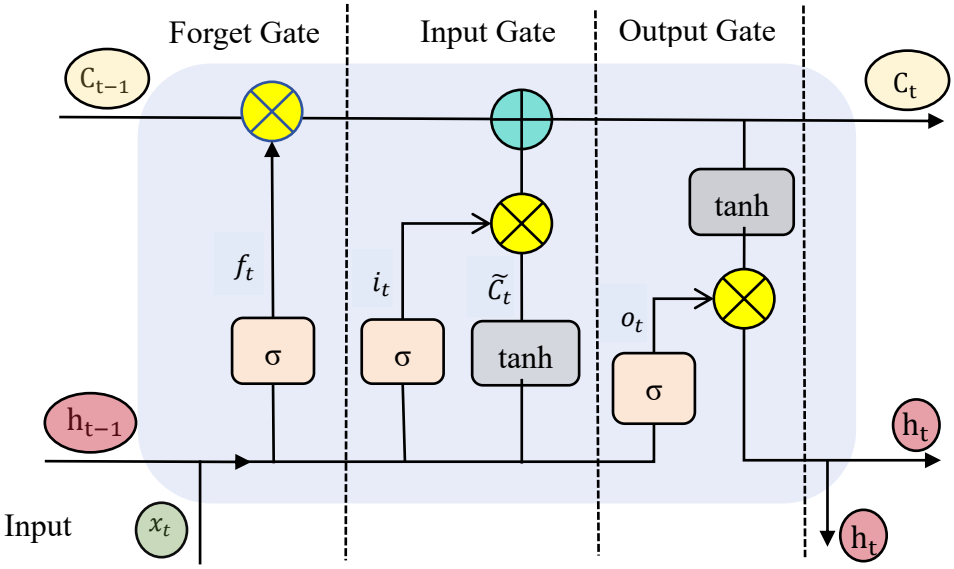


Fig. 1 Structural diagram of the LSTM cells

In Figure 1, C_{t-1} represents unit state at the previous moment, h_{t-1} denotes the output vector of the hidden layer at the previous moment, x_t is the current input vector of the LSTM neuron, C_t indicates the current unit state, and h_t is the output vector of the current hidden layer. The activation function used is tanh, while σ denotes the sigmoid function. The core of the LSTM unit is the unit state, through which related data is transmitted in a time-series format and updated via three gates: input (i), forget (f), and output (o). The forget gate selectively erases the input from the prior node as defined in Equation (1):

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

where W_f and b_f are the weight vector and bias vector of the forget gate, respectively.

The input gate selectively processes the input information as described in equations (2) and (3). The input gate and the forget gate collectively determine the update of the cell state, as depicted in Equation (4).

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

$$\widetilde{C_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{3}$$

and

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}, \tag{4}$$

where W_i and b_i represent the weight vector and bias vector of the input gate, respectively. The weight vectors for the current state of the unit are denoted by W_c and b_c. $*$ is the inner product of the vectors.

According to the definition of equations (5) and (6), the output gate determines the output of the current state:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{5}$$

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

where W_o and b_o represent the weight vector and bias vector of the output gate, respectively.

The BiLSTM model employed in this study combines a forward LSTM and a backward LSTM in an inverse configuration[16]. This structure enables a more comprehensive understanding of the context within the sequence by simultaneously processing past and future information. The outputs from the two units are concatenated to form the final output. As seen in Figure 2.

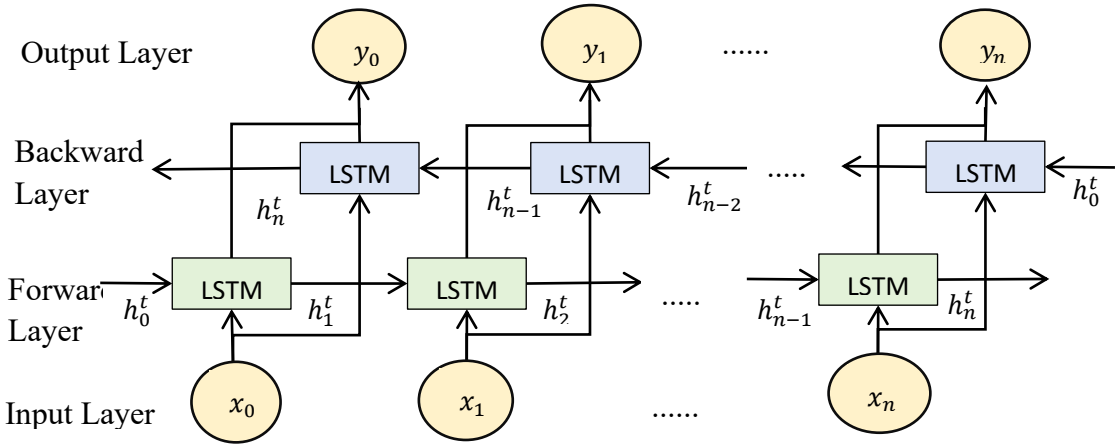


Fig. 2 Structure of BiLSTM model

The process of combining the states at each level of the hidden layer of in a BiLSTM is depicted in Equation (7):

$$\begin{cases} h_t = LSTM(x_t, h_{t-1}) \\ h_i = LSTM(x_i, h_{i-1}) \\ h'_t = a_t h_t + b_t h_i + c_t \end{cases}, \tag{7}$$

where h'_t represents the state at each hidden layer, h_t denotes the forward layer state, h_i indicates the backward layer state, a_t and b_t are the weight, and c_t is the bias.

B. Load prediction model construction

Random Forest is an ensemble learning method that consists of a number of decision trees. As shown in Figure 3, each decision tree is trained on a randomly selected subset of features and then the final decision is made by voting or averaging. In smart grid load forecasting, Random Forest reduces the risk of overfitting by integrating multiple decision trees and effectively extracts the most influential features from a multitude of features, thereby enhancing the model's prediction accuracy.

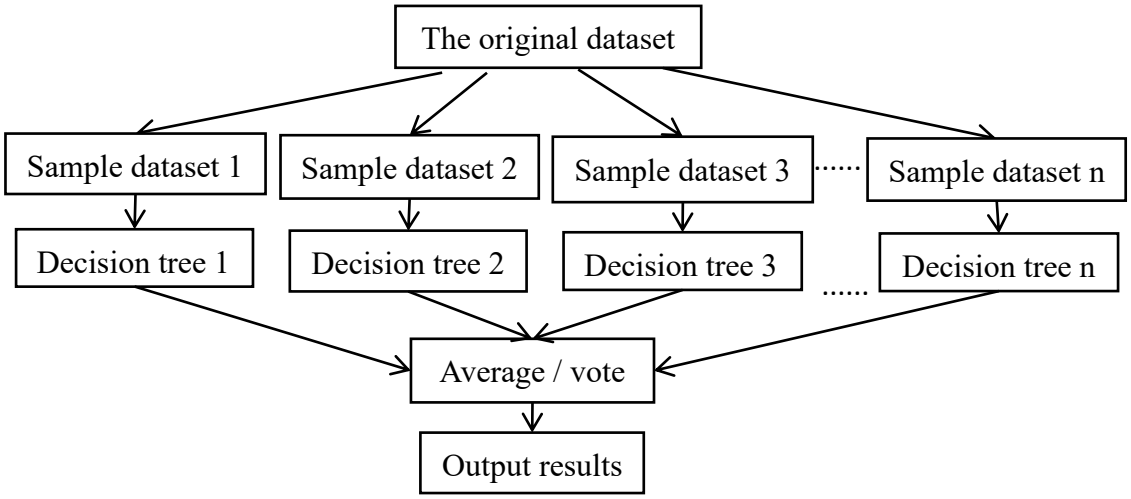


Fig. 3 Schematic diagram of the random forest

The newly constructed model comprises three main components: data feature extraction and segmentation, the hidden layer, and the predictive output layer. This process is illustrated in Figure 4. Initially, data analysis is conducted to cleanse missing feature columns. Subsequently, correlation analysis is carried out on the raw data. Ultimately, the data are normalized using the MinMaxScaler formula. The normalization formula is presented in Equation (8):

$$X_{std} = \frac{(X - X_{min})}{(X_{max} - X_{min})}, \tag{8}$$

where X_{max} and X_{min} represent the maximum and minimum values of the data sequence, respectively. The final normalized value is X_{std}.

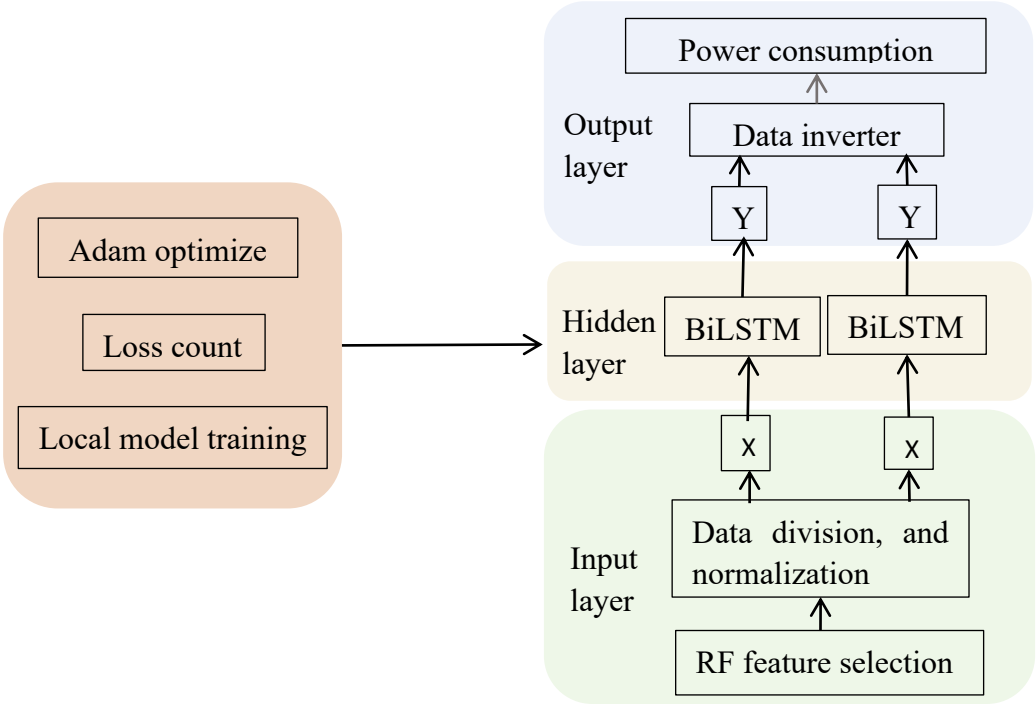


Fig. 4 Diagram of model training process

III. Federated learning and algorithms

To address concerns over user data privacy, the Google research team proposed an innovative machine learning approach known as federated learning in 2016[17]. Originally, it was implemented in the keyboard application of a mobile device to predict the next word[18]. As it can be trained without sharing data, it has found widespread applications in sectors such as finance and healthcare. In terms of privacy and the handling of distributed data, federated learning has proved to have significant advantages. Consequently, it has also been applied in the smart grid domain, where the federated learning load forecasting system is depicted in Figure 5. Here, the aggregation server (utility) conducts global model updates, refreshes the model parameters for participants in each training round, and uploads the local updates.

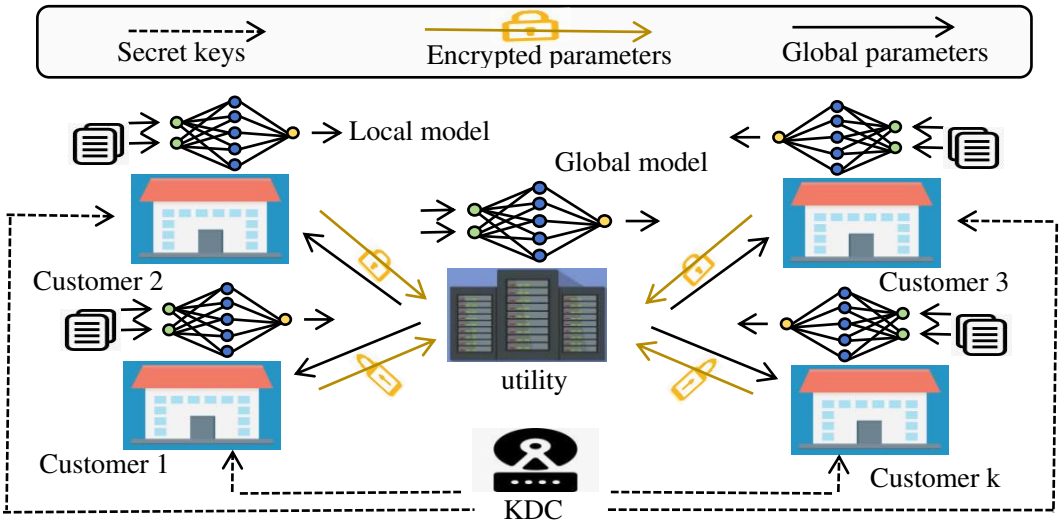


Fig. 5 Load prediction based on federated learning

Assuming that there are N devices participating in federated learning, each device has a local dataset $D_i = \left\{\left(x_j^{(i)}, y_j^{(i)}\right)\right\}_j^{n_i}$, where $x_j^{(i)} = R^d$ denotes the input feature vector of the j-th sample of the i-th device, and $y_j^{(i)}$ denotes the label value of the j-th sample of the i-th device. Let $f_\theta(x)$ denote the output value predicted by a model with parameter $\theta$for input x, and $L\left(y_j^{(i)}, f_\theta(x)\right)$ denote the loss function.The sum of the loss function of the local models for each device is minimized in federated learning, computed as $min \sum_{i=1}^{N} \frac{1}{n_i} \sum_{j=1}^{n_i} L(y_j^{(i)}, f_\theta(x))$.

In the federated learning process, each device trains its local model using the local dataset D_i to obtain the parameters θ_i of the local model. Let $F(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{n_i}{|D|} \sum_{j \in D_I} L(y_j^{(i)}, f_\theta(x))$ denote the loss function of the global model, where $|D| = \sum_{i=1}^{N} n_i$ represents the sum of the dataset sizes for all devices.

The FedProx algorithm introduces a regularization term as a penalty in the original objective function of FedAvg to ensure that the model parameters of each device remain similar. Let $R(\theta)$ denote the regularization term; then, the optimization objective function of the global model is expressed in equation (9):

$$F_{fedprox}(\theta) = F(\theta) + \frac{\mu}{2} \|\theta - \theta^*\|^2 + R(\theta), \tag{9}$$

where μ is the hyperparameter representing the regularization coefficient, and θ^* denotes the initial parameter set of the global model.

The FedProx algorithm uses optimization algorithms such as gradient descent to optimize the objective function and solve for the updated parameters of the global model, as depicted in Algorithm 1.

Algorithm 1 FedProx. The K clients are indexed by k, B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

At the Server Side: Initialize weights θ^* of the global model M
For each round t = 1, 2, ... **do**
  $S_t \leftarrow$ randomly selected m clients
  Send model M to S_t clients
  for each client k **do**
    $\theta_{t+1}^{k} \leftarrow$ ClientUpdate (k, θ_t)
    $\theta_{t+1} \leftarrow \sum_{m=1}^{M} \frac{n_m}{n} L_m(\theta)$
  end
  Send model M to all clients
  At the Client Side:
  ClientUpdate(k, θ_t) **procedure**
  B $\leftarrow$ Split P_k into batches of size B
  for each epoch e < E do
    for batch b ∈ B **do**
      $\Delta\theta_i \leftarrow \frac{1}{B} * \sum \Delta L(\theta_i, b)$
      $\Delta\theta_{ireg} \leftarrow \eta * (\theta_i - \theta_i^k)$
      $\Delta\theta_i \leftarrow \Delta\theta_i + \Delta\theta_{ireg}$
    end
    send $\Delta\theta_i$ to server
  end
end

Start

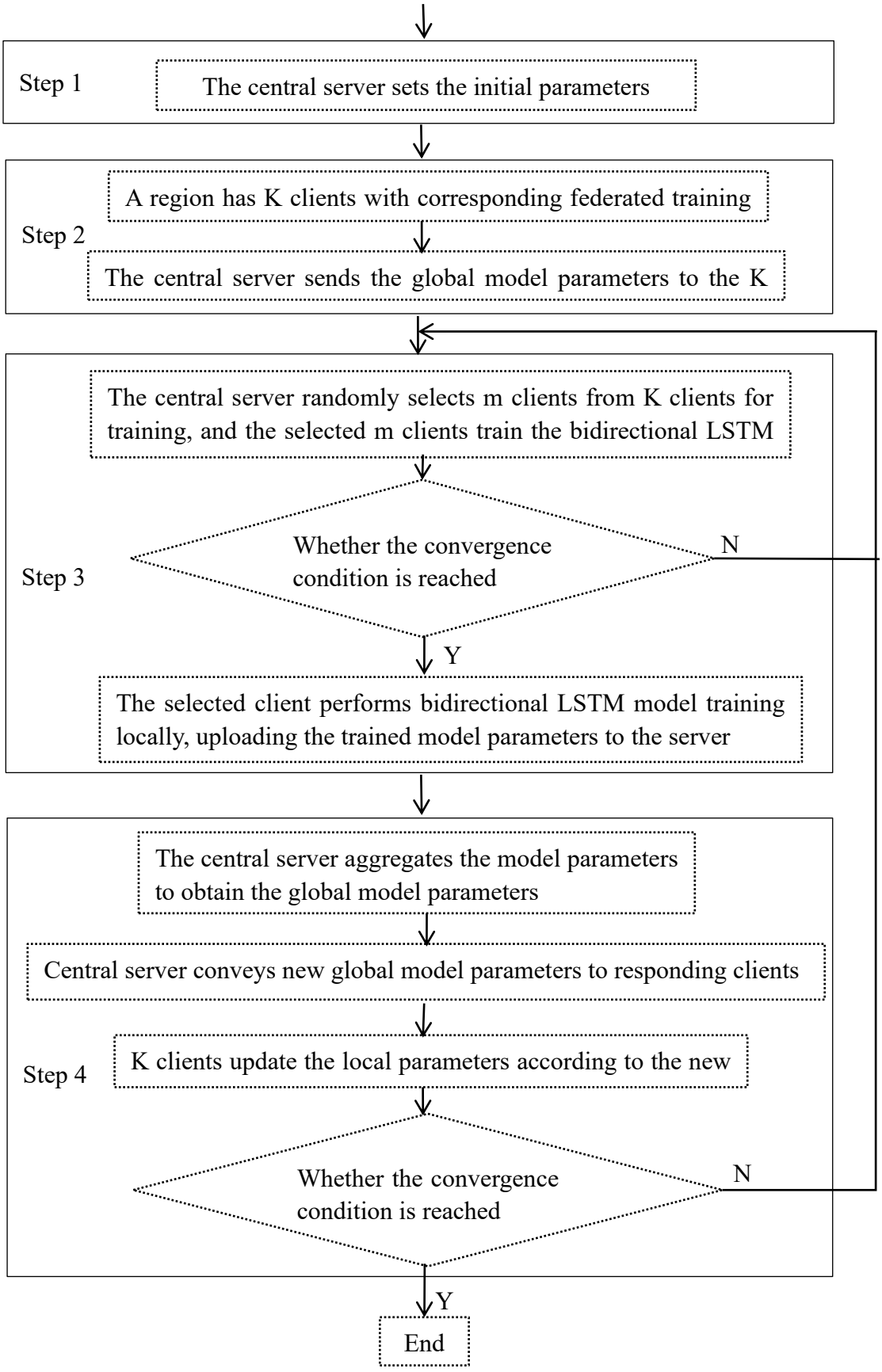


Fig. 6 The proposed federated learning based algorithm

Figure 6 shows the flowchart of the proposed federation learning based algorithm.

The formulae for updating the local model parameters are given in (10):

$$\begin{cases} W_{z,t}^{i} = W_{z,t} - \eta g_{z,i} \\ W_{f,t}^{i} = W_{f,t} - \eta g_{f,i} \end{cases}, \tag{10}$$

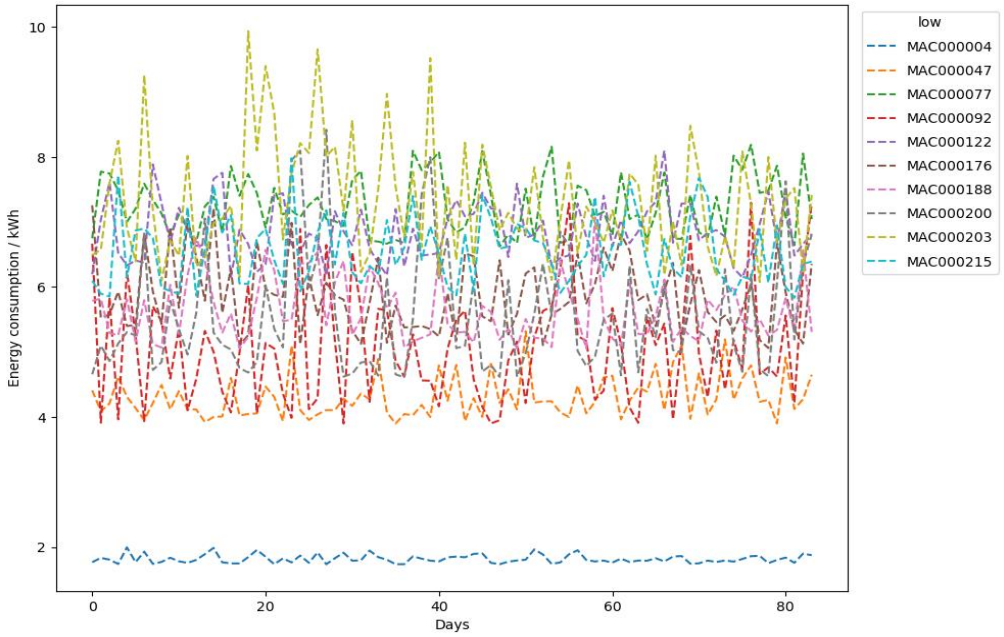
where $W_{z,t}$, $W_{f,i}$ are initialized parameters. $W_{z,t}^{i}$, $W_{f,t}^{i}$ are parameters updated after model training. $g_{z,i}$, $g_{f,i}$ denote the gradient. The formula for server aggregation of all parameters is as follows:

$$\begin{cases} W_{z,t+1} = \sum_{k} \frac{n_k}{n} W_{z,t} + \sum_{k} \frac{n_k}{n} W_{z,t}^{I} \\ W_{f,t+1} = \sum_{k} \frac{n_k}{n} W_{f,t} + \sum_{k} \frac{n_k}{n} W_{f,t}^{I} \end{cases}. \tag{11}$$

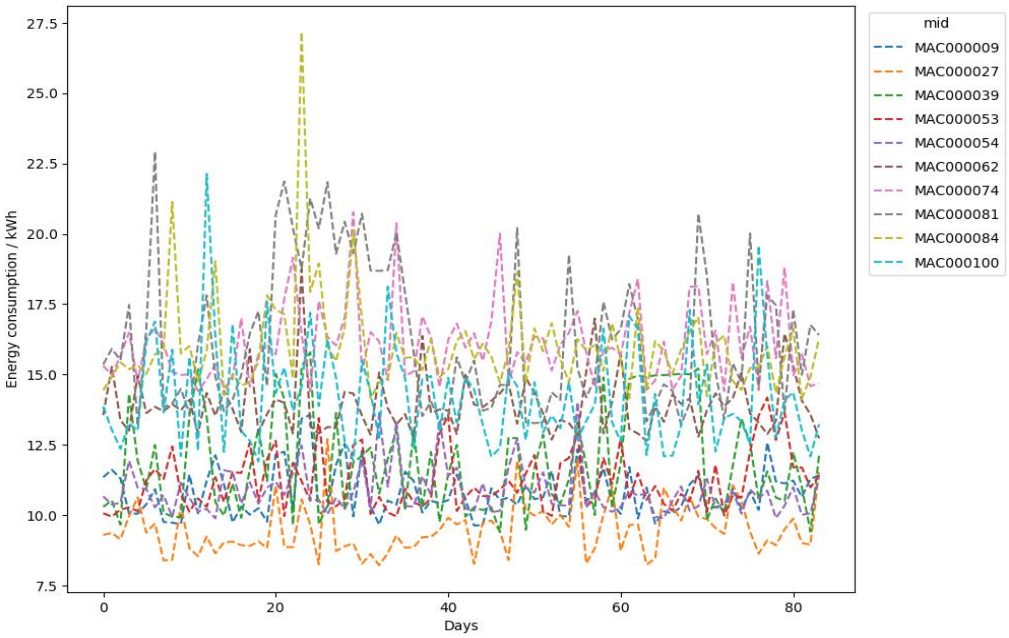
Ⅳ. Experimental analysis

A. Data set

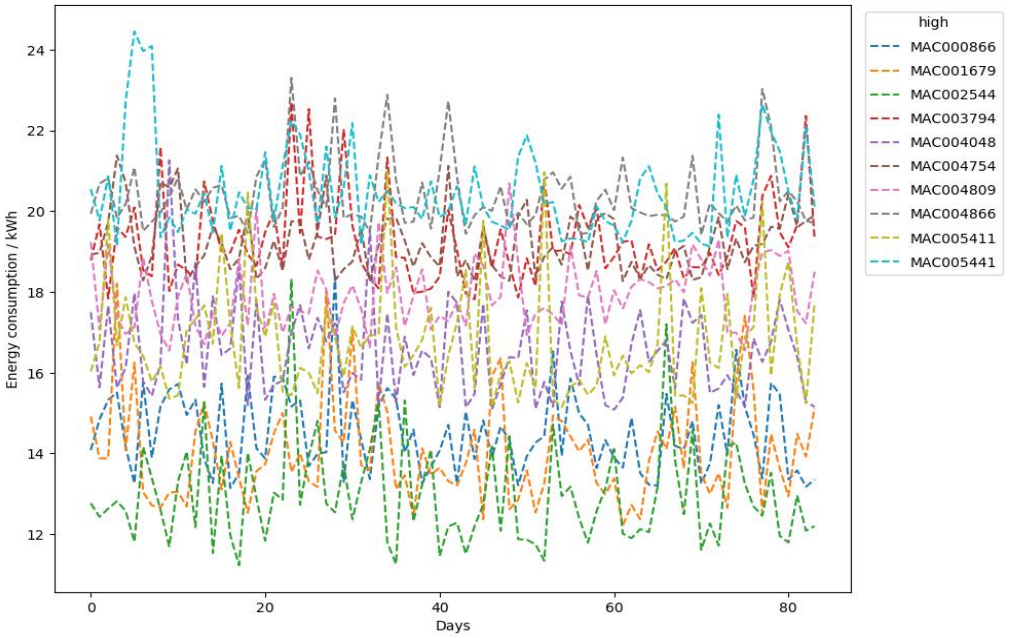
This study draws on a household load dataset from the London community as an experimental dataset. It was collected from November 2011 to February 2014, giving a total of 28 months of available data. The load data was collected by smart meters every half hour. This dataset is divided into three groups based on the boundaries of 9 kWh and 18 kWh of electricity consumption per household per day. Figure 7 presents the electricity consumption of customers in each group, Figure (a) is the low group below 9 kWh, Figure (b) displays the medium group and Figure (c) represents the high group above 18 kWh.



(a) Low group



(b) Mid group



(c) High group

Fig. 7 Daily electricity consumption per household group

B. Feature selection and model evaluation

In the experiments, a random forest regression model is used to determine how much each feature influences the prediction results. The model calculates the importance of each feature by summarizing the split contribution of each feature to the decision tree nodes. Its expression is given by Equation (12).

$$Improtance(X_j) = \frac{\sum_{t \in T} I(t,X_j)p(t)}{\sum_{t \in T} p(t)}, \quad (12)$$

where T denotes all decision trees, X_j denotes feature j, $I(t, X_j)$ denotes the degree of influence of feature X_j on the distribution of nodes in decision tree t, and $p(t)$ denotes the proportion of decision tree t in the set. The significance of a feature is directly proportional to its importance value, indicating its greater influence on the model's prediction outcome. As Figure 8 illustrated, the correlation coefficients of parentTemperature, temperature and electrical load are 0.3404 and 0.2748, indicating that the correlation between these two factors and electrical load is high, while the correlation coefficients of the other factors are low, indicating that the correlation between these factors and electrical load is low. Therefore, electrical load, parentTemperature and temperature were used as input characteristics.

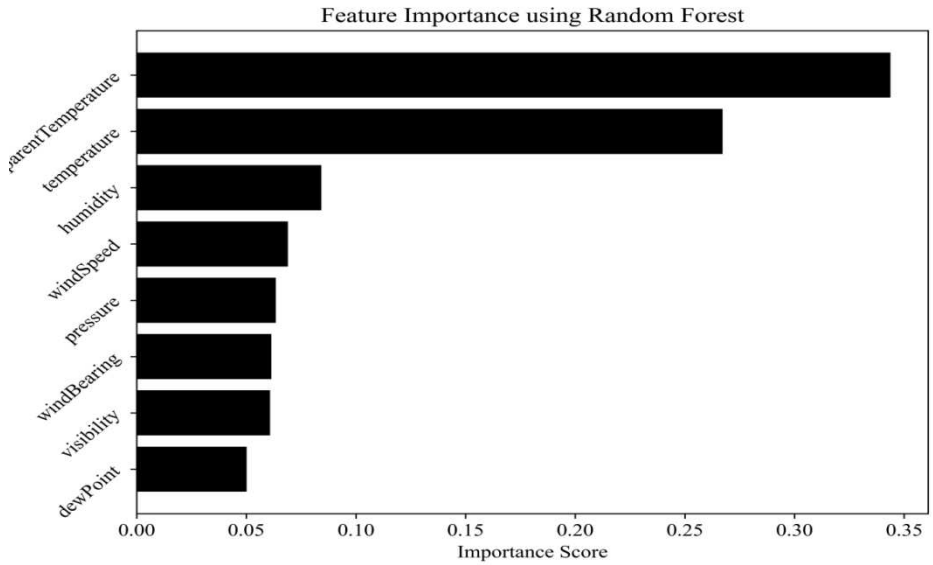


Fig. 8 A feature importance plot

Each client uses the BiLSTM model. The model has two hidden layers, each with 150 cells. The hyperbolic cut function is used as an activation function for the hidden layer. The learning rate is set to 0.005, with a step size of 7. The model predictions' accuracy is determined by evaluating the mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE). Using MSE as a loss function. The formulas for these three evaluation methods are as follows.

$$MSE = \frac{\sum_{i=1}^{n}(r_i - r_i^*)^2}{n}, \quad (13)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(r_i - r_i^*)^2}{n}}, \tag{14}$$

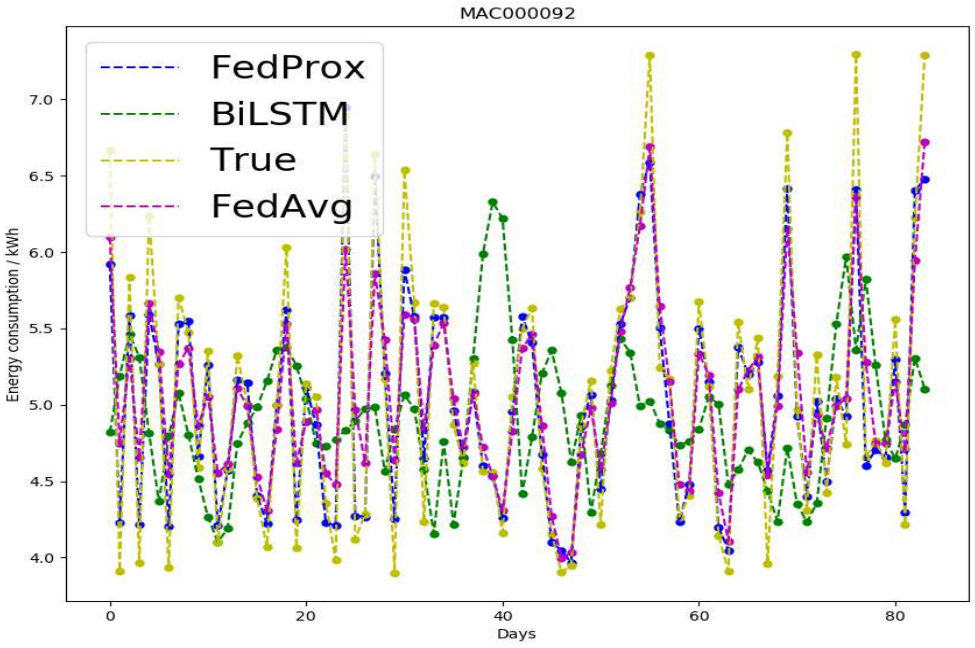
and

$$MAE = \frac{\sum_{i=1}^{n}|r_i - r_i^*|}{n}, \tag{15}$$

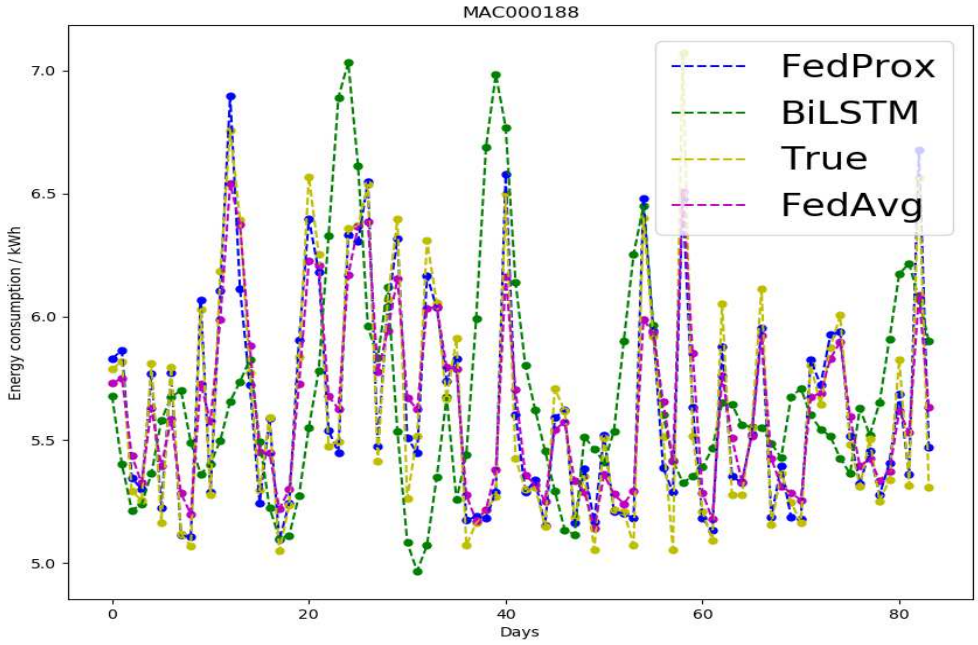
where r_i is the true value of the load to be predicted, r_i^* is the predicted value, and n is the total number of test samples.

C. Experimental result

To verify the accuracy of the model prediction results. In this paper, the prediction results of the combined model are compared with the prediction results of other learning methods and the actual values. Figure 9 displays the results for two households in the low group. Where the X-axis indicates the time series of each day and the Y-axis indicates the total electricity consumption of each day. The yellow curve represents the actual value. The blue curve represents the prediction result after training using BiLSTM and FedProx algorithm (i.e., the combined model proposed in this paper). The purple curve represents the experimental results using the combined model of BiLSTM and FedAvg algorithm. The green curve represents the results of the traditional centralized BiLSTM algorithm. It can be seen that the prediction results of the combined model are closer to the real values.



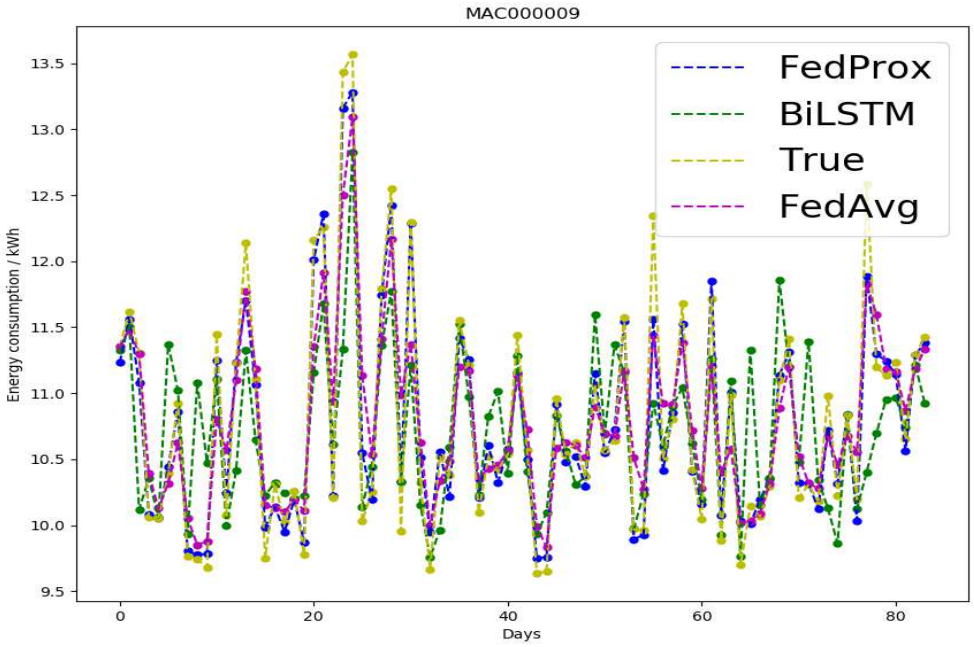
(a) MAC000092



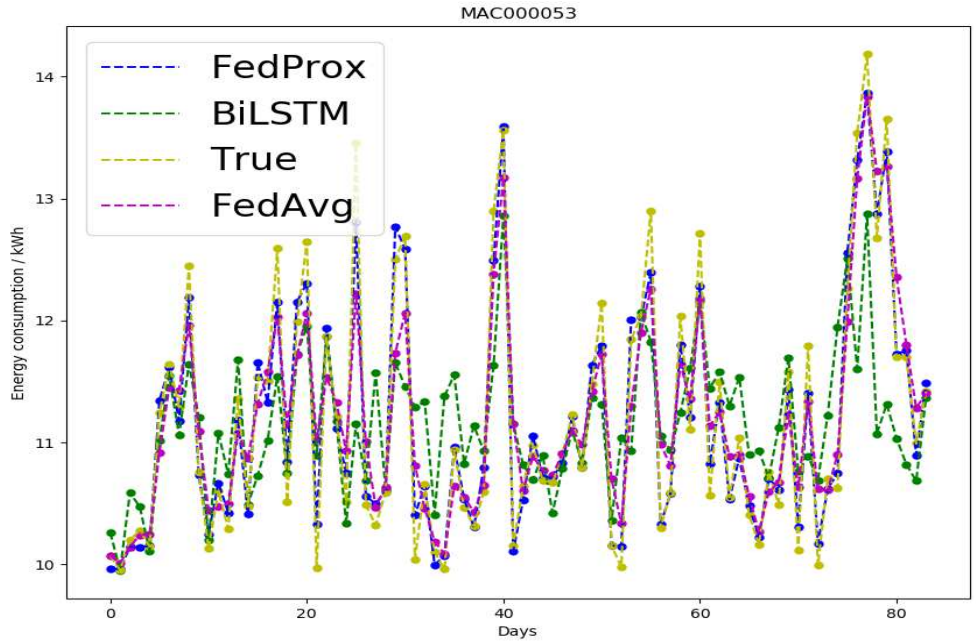
(b) MAC000188

Fig. 9 Predictions for the low group

The model also performs well in predicting customers with different levels of electricity consumption. As shown in Figures 10 and 11, the prediction results are plotted for two households in the medium and high groups, respectively.

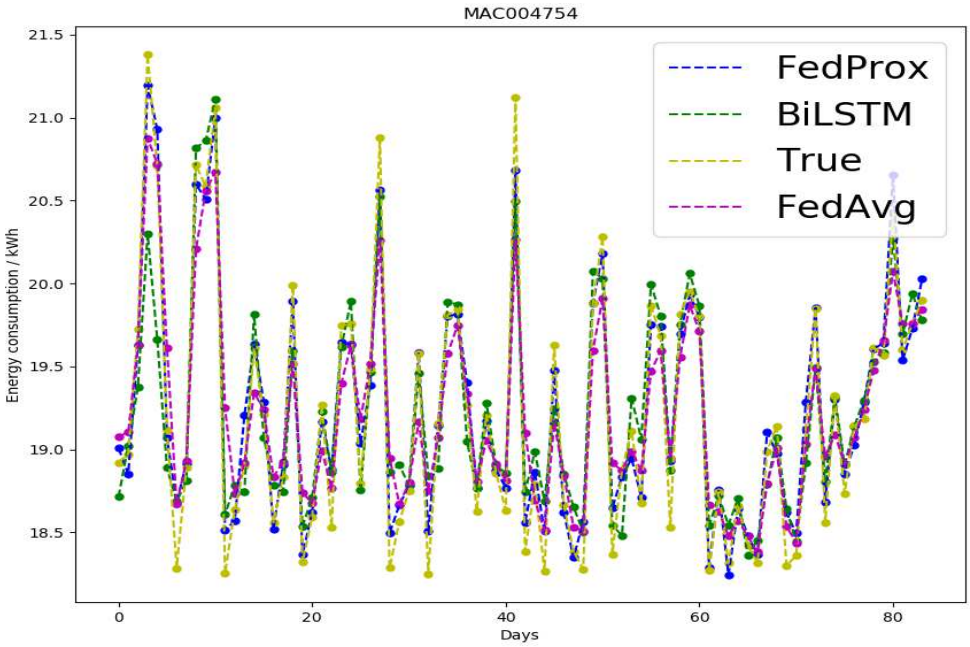


(a) MAC00009

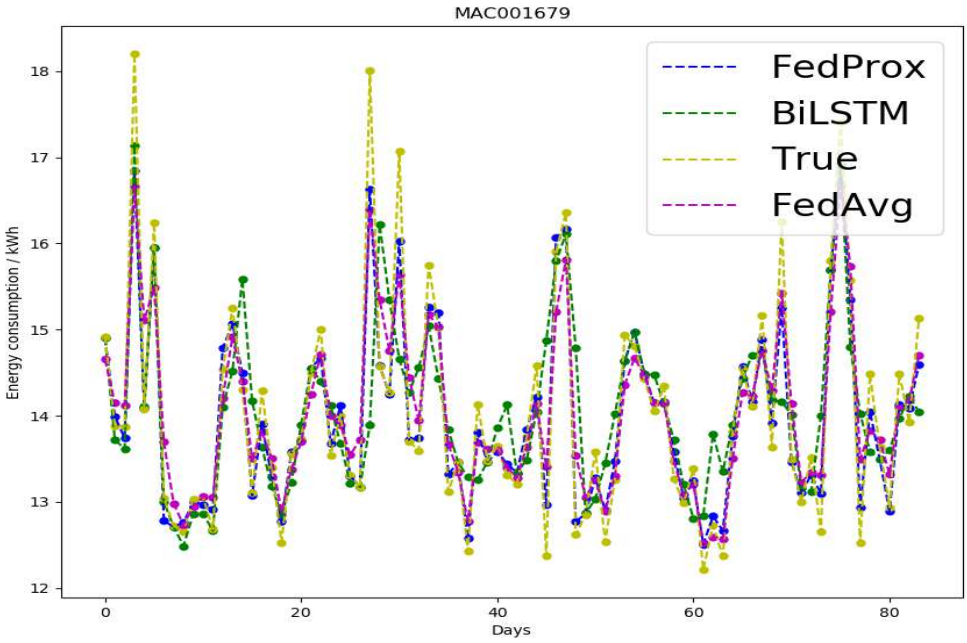


(b) MAC00053

Fig. 10 Predictions for the mid group



(a) MAC004754



(b) MAC001679

Fig. 11 Predictions for the high group

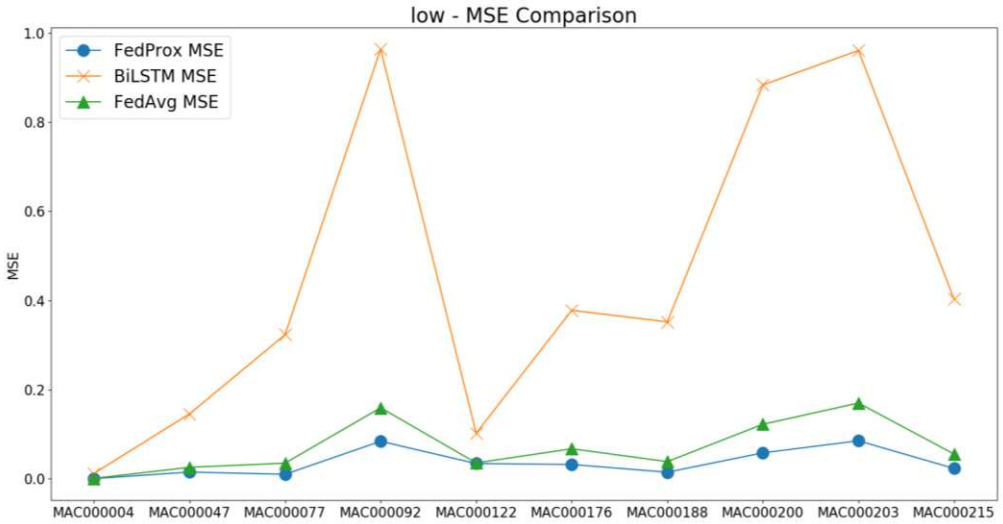
To further validate the effectiveness of the method for households with different levels of electricity consumption. Therefore the prediction results were evaluated as illustrated in the Table 1. From the table, it can be seen that the prediction performance of the method is basically the same for users with different electricity consumption levels. This proves that the method has high generalisability.

Table 1 Predictive performance for different users

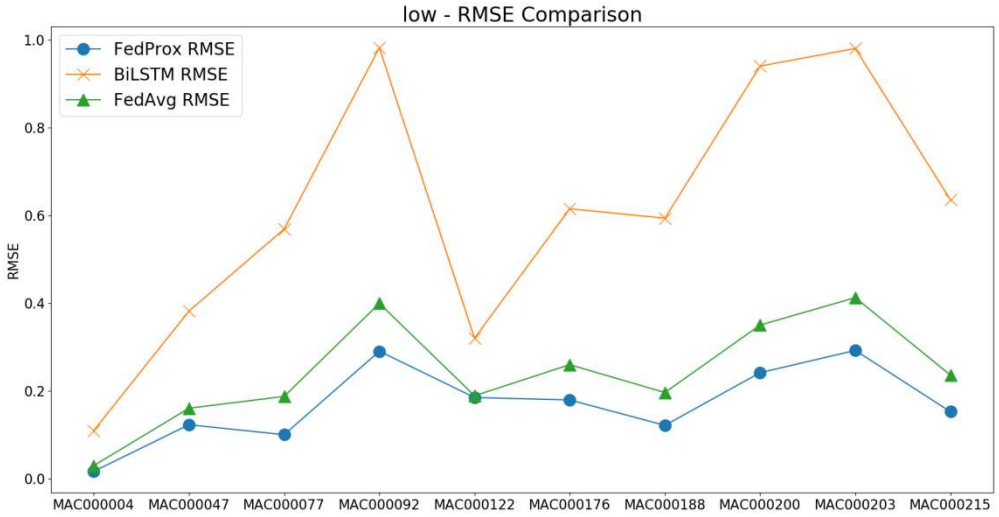
Consumer		Metrics		
		MSE	RMSE	MAE
Low	MAC000203	0.16	0.41	0.33
	MAC000092	0.16	0.40	0.32
Mid	MAC000062	0.21	0.46	0.26
	MAC000074	0.18	0.42	0.26
High	MAC004048	0.24	0.49	0.27
	MAC005411	0.22	0.47	0.31

In this study, different groups of training prediction methods were evaluated. The prediction accuracy of the method is described more intuitively. Figure 12 shows a comparison of the MSE, RMSE and MAE values for the three prediction methods in the low group. Figure 13 represents the medium group and Figure 14 exhibits the high group. The blue curves in the figure represent the prediction methods presented in this

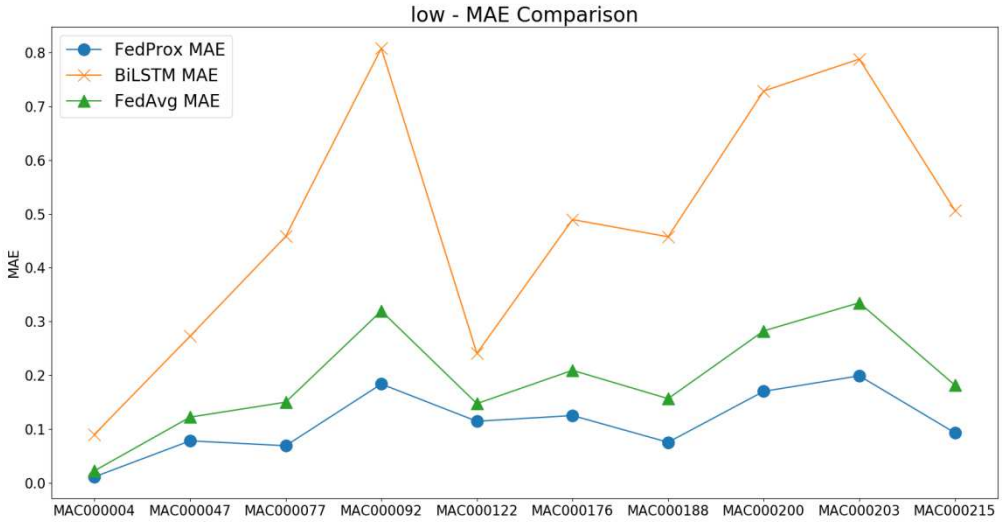
study. It is obvious from the figure that the prediction performance of the proposed in this paper is significantly better than the other methods. This indicates that the method significantly improves the prediction of household electricity consumption while maintaining user privacy.



(a) MSE comparison

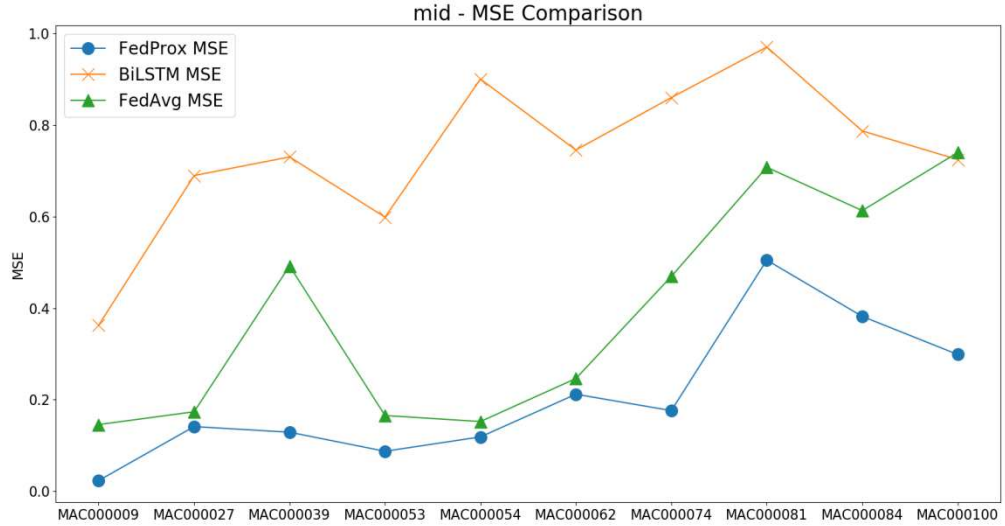


(b) RMSE comparison

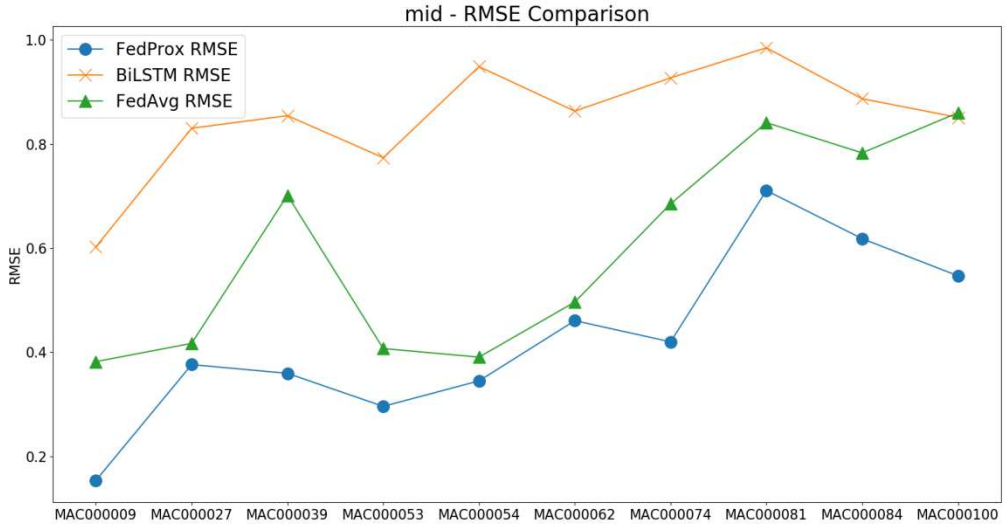


(c) MAE comparison

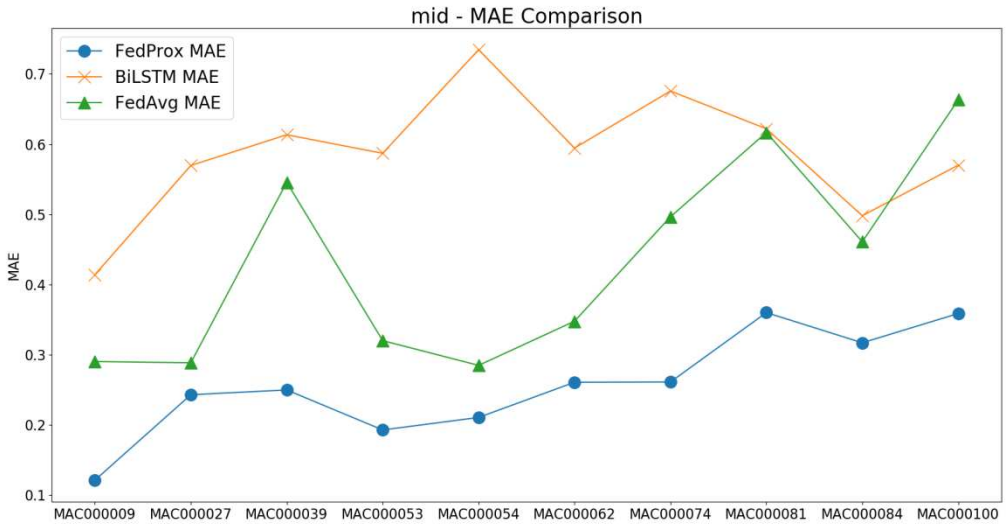
Fig. 12 Comparison of the prediction performance in low group



(a) MSE comparison

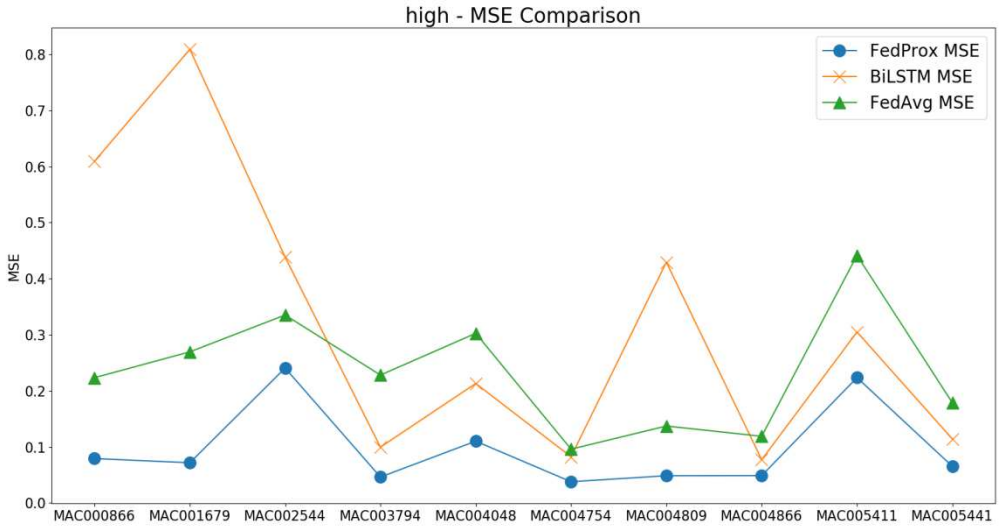


(b) RMSE comparison

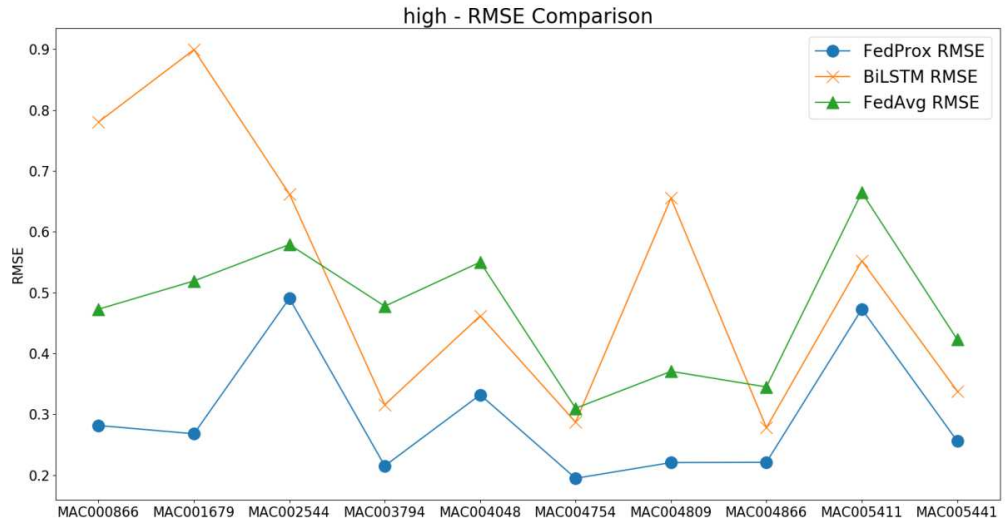


(c) MAE comparison

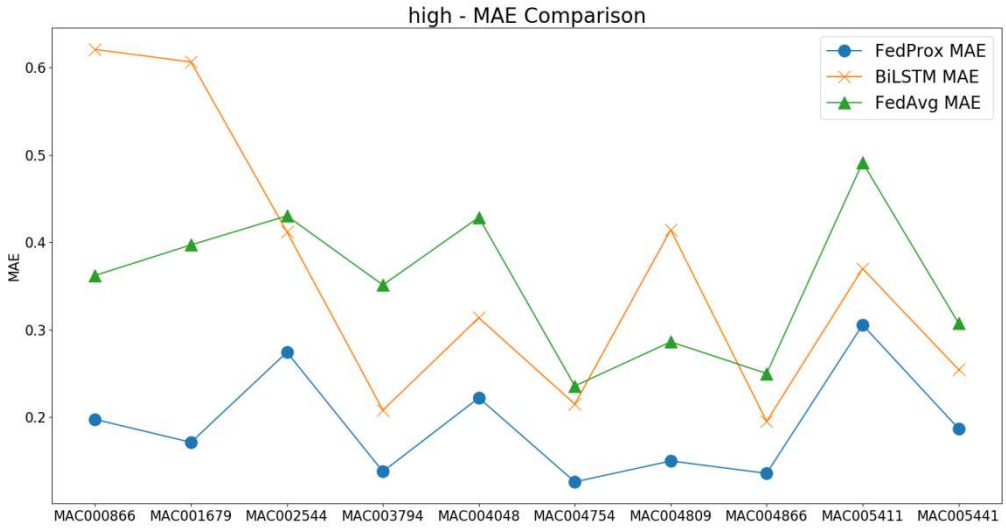
Fig. 13 Comparison of the prediction performance in mid group



(a) MSE comparison



(b) RMSE comparison



(c) MAE comparison

Fig. 14 Comparison of the prediction performance in high group

The proposed method performs better in terms of prediction results and prediction performance. And the time required for train the models is also of great concern. Therefore Table 2 compares the training time of each model, which further confirms the feasibility of the combined model proposed in this paper in smart grid load forecasting.

Table 2 Training time for each model

Group	Training time		
	FedProx	BiLSTM	FedAvg
Low	1350.80s	1664.30s	1720.54s
Mid	1361.62s	1693.97s	1727.52s
High	1314.18s	1672.22s	1749.92s

Ⅴ. Summary

In this paper, a smart grid load forecasting model combining Random Forest (RF) and Bidirectional Long Short Term Memory Network (BiLSTM) is proposed. A federated learning framework is adapted to address the challenges of traditional forecasting models in handling high-dimensional heterogeneous data and protecting user privacy. The prediction accuracy and generalization ability are significantly improved by combining the feature extraction capability of Random Forest with the time series processing advantage of BiLSTM. The effectiveness and superiority of the model is verified by experimental results on real datasets.

Data complexity and size continue to increase and privacy regulations become more stringent in the context of the modern smart grid. Future work will focus on refining the model, integrating other advanced machine learning techniques, and applying it to other aspects of smart grid management. Such as demand response, distributed resource optimization and fault detection. Ultimately, implementing the efficient operation of the smart grid.

Data availability statement

The datasets used and analyzed during the current study available from the corresponding author on reasonable request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China, under Grant 61801227; in part by the Qinglan Project of Jiangsu
Province under Grant QLGC2021; and in part by the Future
Network Scientific Research Fund Project under Grant FNSRFP-2021-YB-28.

References

[1] Y. Yang, W. Li, T. A. Gulliver and S. Li, "Bayesian Deep Learning-Based Probabilistic Load Forecasting in Smart Grids," in IEEE Transactions on Industrial Informatics, vol. 16, no. 7, pp. 4703-4713, July 2020, doi: 10.1109/TII.2019.2942353.
[2] A. Almalaq and J. J. Zhang, "Evolutionary Deep Learning-Based Energy Consumption Prediction for Buildings," in IEEE Access, vol. 7, pp. 1520-1531, 2019, doi: 10.1109/ACCESS.2018.2887023.
[3] X. Lin, R. Zamora, C. Baguley, and A. Srivastava, "A Hybrid ShortTerm Load Forecasting Approach for Individual Residential Customer," IEEE Transactions on Power Delivery, 2022.
[4] H. Shi, M. Xu and R. Li, "Deep Learning for Household Load Forecasting—A Novel Pooling Deep RNN," in IEEE Transactions on Smart Grid, vol. 9, no. 5, pp. 5271-5280, Sept. 2018, doi: 10.1109/TSG.2017.2686012.
[5] C. Liu, Z. Jin, J. Gu and C. Qiu, "Short-term load forecasting using a long short-term memory network," 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Turin, Italy, 2017, pp. 1-6, doi: 10.1109/ISGTEurope.2017.8260110.
[6] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," in IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 841-851, Jan. 2019, doi: 10.1109/TSG.2017.2753802.
[7] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in ICC 2020-2020 IEEE International Conference on Communications (ICC), 2020: IEEE, pp. 1-6.
[8] M. Alhussein, K. Aurangzeb and S. I. Haider, "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," in IEEE Access, vol. 8, pp. 180544-180557, 2020, doi: 10.1109/ACCESS.2020.3028281.
[9] H. Xu, G. Fan, G. Kuang and Y. Song, "Construction and Application of Short-Term and Mid-Term Power System Load Forecasting Model Based on Hybrid Deep Learning," in IEEE Access, vol. 11, pp. 37494-37507, 2023, doi: 10.1109/ACCESS.2023.3266783.
[10] Y. Chen et al., "CNN-BiLSTM Short-Term Wind Power Forecasting Method Based on Feature Selection," in IEEE Journal of Radio Frequency Identification, vol. 6, pp. 922-927, 2022, doi: 10.1109/JRFID.2022.3213753.

[11] V. Venkataramanan, S. Kaza and A. M. Annaswamy, "DER Forecast Using Privacy-Preserving Federated Learning," in IEEE Internet of Things Journal, vol. 10, no. 3, pp. 2046-2055, 1 Feb.1, 2023, doi: 10.1109/JIOT.2022.3157299.
[12] M. A. Husnoo, A. Anwar, N. Hosseinzadeh, S. N. Islam, A. N. Mahmood and R. Doss, "A Secure Federated Learning Framework for Residential Short-Term Load Forecasting," in IEEE Transactions on Smart Grid, vol. 15, no. 2, pp. 2044-2055, March 2024, doi: 10.1109/TSG.2023.3292382.
[13] D. Qin, C. Wang, Q. Wen, W. Chen, L. Sun and Y. Wang, "Personalized Federated DARTS for Electricity Load Forecasting of Individual Buildings," in IEEE Transactions on Smart Grid, vol. 14, no. 6, pp. 4888-4901, Nov. 2023, doi: 10.1109/TSG.2023.3253855.
[14] X. -Y. Zhang, J. -R. Córdoba-Pachón, P. Guo, C. Watkins and S. Kuenzel, "Privacy-Preserving Federated Learning for Value-Added Service Model in Advanced Metering Infrastructure," in IEEE Transactions on Computational Social Systems, vol. 11, no. 1, pp. 117-131, Feb. 2024, doi: 10.1109/TCSS.2022.3204361.
[15] LI Ran, DING Xing, SUN Fan, et al. Ultra-short-term photovoltaic power prediction based on Wide & DeepXGB2LSTM model[J]. Electric Power Automation Equipment, 2021, 41（7）: 31−37(in Chinese).
[16] A. F. Ganai and F. Khursheed, “Predicting next word using RNN and LSTM cells: Stastical language modeling,” Proc. of International Conference on Image Information Processing (ICIIP), pp. 469–474, Nov.2019.
[17] Q. Yang, Y. Liu, T. Chen, Y. Tong, "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, no. 2, pp. 1-19, 2019.
[18] MCMAHANB, MOOREE, RAMAGED, et al. Communication-efficient learning of deep networks from decentralized data[C]//Proceedings of the 20th International Conference on Artificial Intelligence and Statistics．Fort Lauderdale：PMLR，2017 ： 1273-1282