



OPEN ACCESS

EDITED BY

Chunhe Song,
Shenyang Institute of Automation (CAS),
China

REVIEWED BY

Yong Zhou,
ShanghaiTech University, China
Jinfa Wang,
Institute of Information Engineering
(CAS), China

*CORRESPONDENCE

Ai-Xia Wang,
wangaixia@ise.neu.edu.cn

SPECIALTY SECTION

This article was submitted to Smart
Grids,
a section of the journal
Frontiers in Energy Research

RECEIVED 24 June 2022

ACCEPTED 18 July 2022

PUBLISHED 26 August 2022

CITATION

Wang A-X and Li J-J (2022), A novel
cloud-edge collaboration based short-
term load forecasting method for
smart grid.

Front. Energy Res. 10:977026.
doi: 10.3389/fenrg.2022.977026

COPYRIGHT

© 2022 Wang and Li. This is an open-
access article distributed under the
terms of the [Creative Commons
Attribution License \(CC BY\)](#). The use,
distribution or reproduction in other
forums is permitted, provided the
original author(s) and the copyright
owner(s) are credited and that the
original publication in this journal is
cited, in accordance with accepted
academic practice. No use, distribution
or reproduction is permitted which does
not comply with these terms.

A novel cloud-edge collaboration based short-term load forecasting method for smart grid

Ai-Xia Wang* and Jing-Jiao Li

College of Information Science and Engineering, Northeastern University, Shenyang, China

With the increasing development of smart grid technology, short-term load forecasting becomes particularly important in power system operation. However, the design of accurate and reliable short-term load forecasting methods and models is challenging due to the volatility and intermittency of renewable energy sources, as well as the privacy and individual characteristics of electricity consumption data from user data. To overcome this issue, in this paper, a novel cloud-edge collaboration short-term load forecasting method is proposed for smart grid. In order to reduce the computational load of edge nodes and improve the accuracy of node prediction, we use the method of building a model pre-training pool to train multiple pre-training models in the cloud layer at the same time. Then we use edge nodes to retrain the pre-trained model, select the optimal model and update the model parameters to achieve short-term load forecasting. To assure the validity of the model and the confidentiality of private data, we utilize the model pre-training pool to minimize edge node training difficulty and employ the approach of secondary edge node training. Finally, extensive experiments confirm the efficacy of our proposed method.

KEYWORDS

smart grid, short-term load forecasting, edge computing, cloud-edge collaboration, reinforcement learning

1 Introduction

As the usage of renewable energy grows, the world is seeing a historic energy revolution. Renewable energy, on the other hand, poses significant hurdles to the smart grid due to its volatility, intermittency, and unpredictability (Meliani et al., 2021). To achieve reliable and economical operation of power systems, accurate short-term load forecasting (STLF) is essential (Yin and Xie, 2021). Since the ultimate goal of the smart grid is to effectively control energy supply and balance, accurate STLF plays a crucial role for the energy stakeholders of the smart grid.

Smart grid devices usually produce a large amount of data. Processing and analyzing these data can objectively reflect the operation of smart grid. However, if the data of smart grid devices are uploaded to the cloud for processing, it will cause great pressure on the

cloud computing center, and the upload rate will also be affected by the delay and bandwidth, resulting in the data cannot be processed in time, making it difficult to deal with emergencies. Recently, a large number of literatures on load forecasting have been examined. The three basic categories of load forecasting methods are statistical learning approaches, machine learning methods, and deep learning methods. For statistical analysis, statistical learning methods commonly employ historical time series data. Statistical learning methods are easy to use and understand, but they are poor for solving nonlinear problems and are typically bound by assumptions (Li, 2020). To overcome the constraints of physical and statistical models, machine learning approaches can integrate external information such as meteorological data. Because of their high generalization capacity, deep learning approaches have also garnered a lot of interest in the prediction and management of smart grids (Song et al., 2021a). However, machine learning methods and deep learning methods have many shortcomings, such as being easy to fall into local optimum, overfitting, and low convergence speed (Feng et al., 2017; Li, 2020; Kotsiopoulos et al., 2021). At the same time, the data of power users often have privacy and personality characteristics, the privacy data cannot be used for model training directly, and the global model cannot represent the user's personality characteristics (Chen et al., 2021).

In response to the above problems, this paper proposes a short-term load forecasting method for smart grids based on edge computing, which aims to achieve effective STLF of smart grids while protecting user privacy. The main contributions of this paper are as follows:

- (1) A method of constructing a model pre-training pool is proposed to reduce the computational load of edge nodes and improve the accuracy of node prediction.
- (2) A pre-training model for secondary training of edge nodes is proposed, the optimal model is selected and the model parameters are updated to achieve short-term load forecasting.
- (3) Considering that a single model may lead to large prediction deviations, an optimal optimization method for multi-model selection is proposed to improve the prediction accuracy.

This paper is organized as follows: The related work is presented in Section 2. Section 3 introduces the framework of an edge computing-based STLF system. Section 4 introduces a short-term load forecasting method based on edge computing. Section 5 provides experiments and corresponding analyses. Finally, the conclusion is given in Section 6.

2 Related work

Edge computing refers to the use of an open platform that integrates network, computing, storage, and application core

capabilities on the side close to the object or data source to provide the nearest end service (Wang et al., 2021). In China, the edge computing alliance ECC is working hard to promote the integration of three technologies, that is, the integration of OICT (Operational, Information, and Communication Technology). The computing objects of edge computing are mainly in four fields, namely the device field, the network field, the data field and the application field (Dong et al., 2022), (Song et al., 2022).

Smart grid is the future development direction of the power industry. Luo et al. (2019) proposed a short-term energy prediction system based on edge computing. Song et al. (Taïk and Cherkaoui, 2020) proposed a method based on cloud-edge collaboration to identify equipment defects in smart grids. Taïk et al. (Song et al., 2020) proposed a method for power load forecasting using edge computing and federated learning. Feng et al. (2021) realized the technology roadmap of edge computing for smart grid from the perspective of the power industry. Song et al. (2021b) proposed a smart grid intrusion detection method based on a multi-classifier architecture. Samie et al. (2019) studied edge computing solutions for smart grids. Zhang et al. (2022) proposed a scheme that can aggregate multi-dimensional data at the distribution edge for privacy protection of smart grids. These studies demonstrate the great potential of edge computing for smart grid applications.

3 Framework of a short-term load forecasting system based on edge computing

Figure 1 shows the STLF architecture of the smart grid based on edge computing. The architecture mainly consists of four parts: cloud layer, main grid, edge node layer, and multi-energy network. In order to share the pressure of cloud computing center and minimize data transmission delay, edge computing nodes can be deployed at IOT devices. Edge computing is more suitable for local and real-time data processing and analysis. Edge computing nodes have certain computing capabilities, can be responsible for data storage and processing within the deployment scope, can independently judge and deal with problems, and support the real-time intelligent decision-making and execution of local businesses. At the same time, the data processed by edge computing nodes is of great value, and these data still need to be gathered and concentrated in the cloud computing center. The cloud computing center carries out data mining, data sharing, training and upgrading of algorithm model. The upgraded algorithm can be applied to edge side data analysis.

3.1 Cloud layer

The main function of the cloud layer is to pre-train the model and pass the pre-trained model to the edge nodes. The

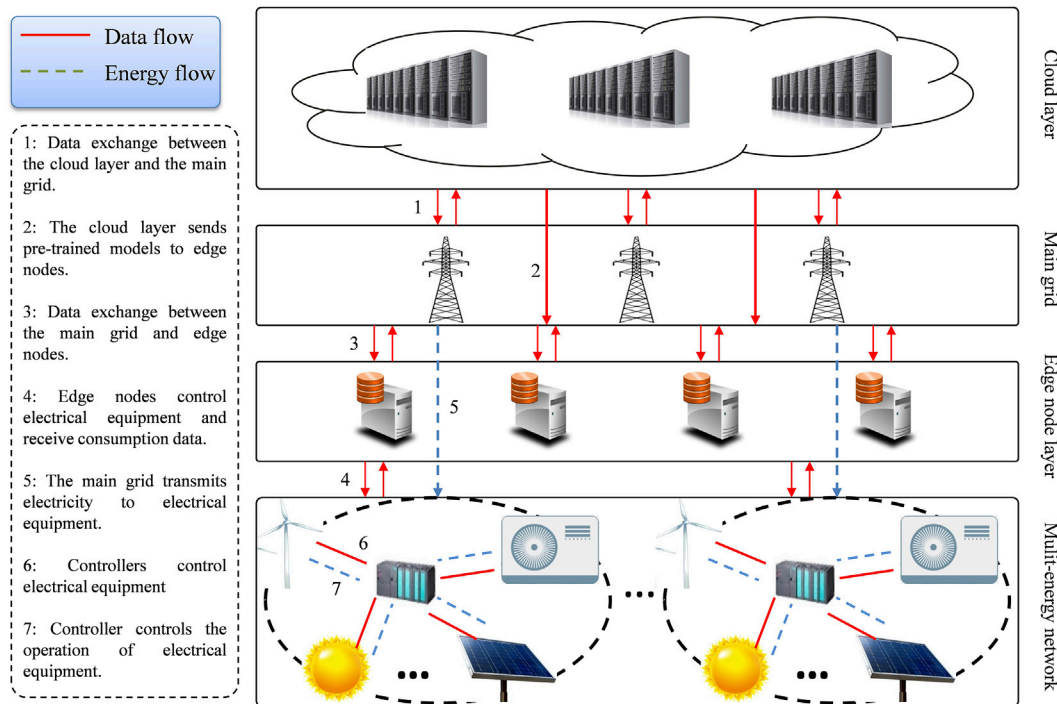


FIGURE 1

Framework of STLF system based on edge computing. 1: Data exchange between the cloud layer and the main grid; 2: The cloud layer sends pre-trained models to edge nodes; 3: Data exchange between the main grid and edge nodes; 4: Edge nodes control electrical equipment and receive consumption data; 5: The main grid transmits electricity to electrical equipment; 6: Controllers control electrical equipment; 7: Controller controls the operation of electrical equipment.

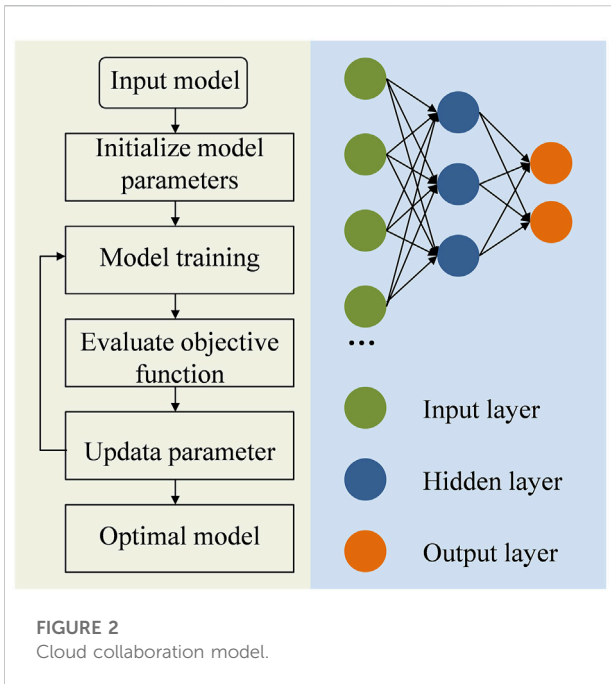
responsibility of the cloud layer is not only to provide pre-trained models for energy devices, but also to ensure that the proposed models meet the computing needs of edge nodes. In the proposed architecture, the cloud layer connects the main grid to obtain electricity consumption data, trains the data, obtains a pre-trained model and transmits it to the edge nodes. We consider arranging a set of STLF models based on machine learning in the cloud layer to form a prediction model pool, and use the data provided by the main grid to pre-train the model in the model training stage. The model pool we used here consists of eight models with three machine learning algorithms and different training algorithms. Specifically, we choose three artificial neural networks (ANN), namely the standard back-propagation network, the momentum-augmented back-propagation network, and the elastic back-propagation network. We selected three Support Vector Regression (SVR) models whose kernels are linear, polynomial, and radial basis functions. We chose 2 gradient boosting machine (GBM) models, namely the GBM model with squared loss function and Laplacian loss function. These models are trained in the cloud layer and passed to edge nodes as initial models for nodes.

TABLE 1 Model pool.

Algorithm	Model	Training algorithm
ANN	M1	Standard back propagation network
	M2	Momentum enhanced back propagation network
	M3	Elastic back propagation network
SVM	M4	Linear kernel
	M5	Polynomial kernel
	M6	Radial basis function kernel
GBM	M7	Square loss function
	M8	Laplace loss function

3.2 Main grid

The main grid is in charge of gathering power use statistics from users, transferring data to the cloud, and supplying energy to electrical devices. The main grid is the major network that provides power to users and acts as a secure data transmission centre.



3.3 Edge node layer

Edge nodes can be deployed at gateways, base stations, etc. to calculate, cache, and transmit energy data. Edge nodes are connected to energy devices through different communication technologies, such as 5G, WIFI or in-vehicle networks. The edge node layer receives the pre-trained model from the cloud service

layer, and adopts the reinforcement learning-based method for model selection and retraining. Because user data often has personalized characteristics and involves sensitive privacy issues, users are often reluctant to disclose specific electricity consumption information. Considering the similarity of users' electricity consumption habits, we adopt edge nodes to receive pre-trained models from the cloud layer for secondary training to improve training efficiency and improve model accuracy. The prediction results of edge nodes can help the stable operation and load shedding of the local energy network (Song et al., 2018).

3.4 Multi-energy network

A multi-energy network consists of multiple energy devices. An energy device can be any entity, user, or device in the network that provides and requires energy. Multi-energy devices are connected to edge nodes through controllers. Energy devices can collect and generate energy data according to their species (Liu et al., 2019).

4 Short-term load forecasting method based on edge computing

In this section, we provide an edge computing-based short-term load forecasting approach. It comprises two sections in our scheme: (1) the collaboration model deployed at the cloud layer; (2) the edge RL model deployed at the edge nodes. The cloud layer pre-trains its pre-made model using data supplied by the main network in the

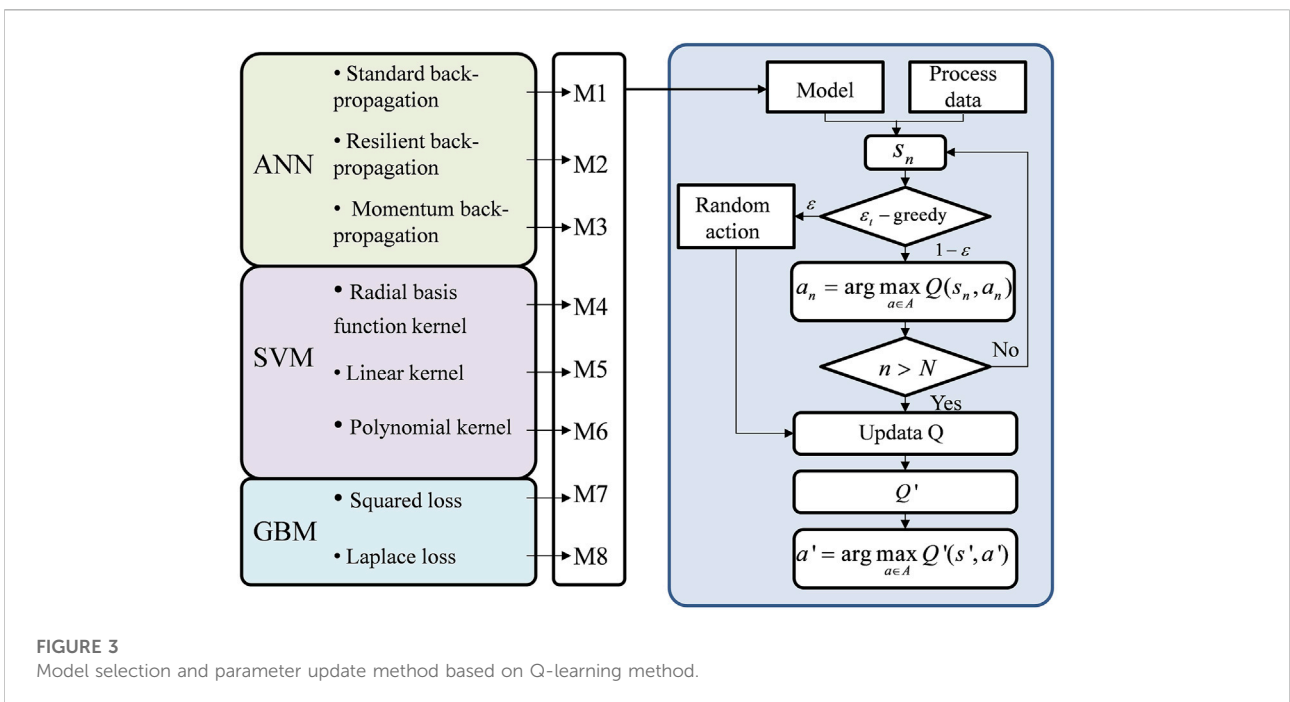
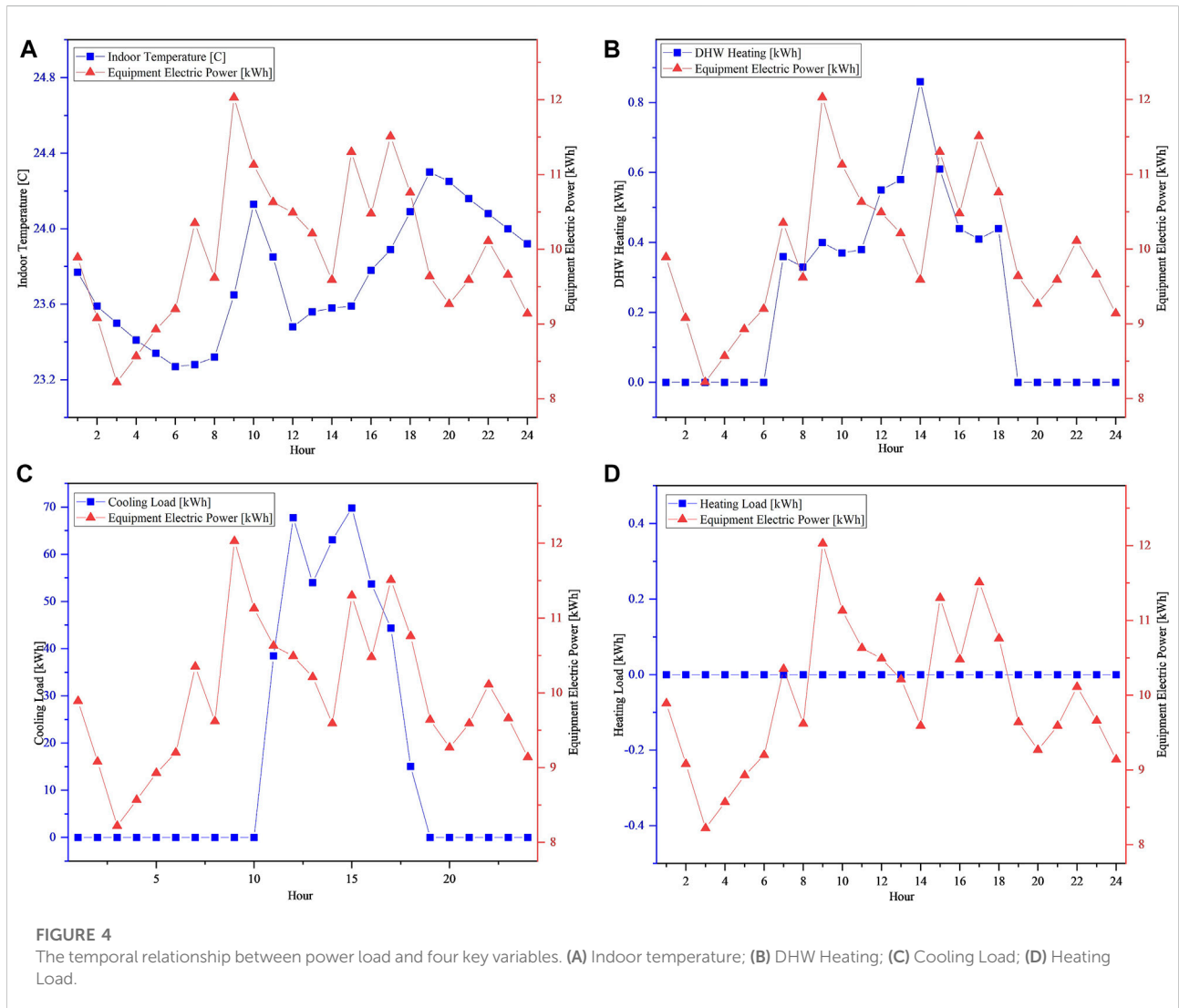


TABLE 2 The descriptive statistical characteristics of the study samples.

Type	Length	Mean	Max	Min	Median	Standard
Indoor Temperature [C]	8760	23.3392	26.34	17.97	23.65	1.4471
Indoor Relative Humidity [%]	8760	46.6581	67.46	21.13	47.86	6.5663
Equipment Electric Power [kWh]	8760	24.8337	70.91	7.44	15.88	17.0915
DHW Heating [kWh]	8760	1.3871	6.59	0	0.66	1.5440
Cooling Load [kWh]	8760	63.0640	29.23	0	19.11	77.2372
Heating Load [kWh]	8760	0	0	0	0	0



collaborative DRL model of the cloud layer. To perform network edge cache updates, edge nodes employ the DRL approach to identify the best prediction model based on user privacy data (Song et al., 2021c). In our model, the collaborative model is trained and offloaded to edge nodes to reduce the computational

cost, improve computational efficiency and protect users' private data. According to the pre-trained model from the cloud layer, the edge node adopts the model-free adaptive dynamic programming algorithm Q-learning to find the optimal strategy of the best prediction model at each prediction time step.

4.1 Cloud collaboration model

In the cloud collaborative reinforcement learning model, the cloud layer obtains electricity consumption data from the main grid and trains the data to obtain a pre-trained model and pass it to the edge nodes. We consider arranging a set of STLF models based on machine learning in the cloud layer to form a prediction model pool and use the data provided by the main network to pre-train the model in the model training phase. Here our prefabricated model pool consists of eight models (As shown in Table 1), including: (1) Three artificial neural networks (ANN), namely standard back-propagation network, momentum-enhanced back-propagation network, and elastic back-propagation network; (2) Three gradient boosting machine (SVM) models, which are linear, polynomial, and radial basis function SVM models; (3) Two gradient boosting machine (GBM) models, which are GBM models with squared loss function and Laplacian loss function. We train the model at the cloud layer and pass it to the edge node as the initial model for the edge node.

The data acquired from the energy grid is used to train the cloud collaboration improvement model. That is, model training is done independently on each pre-made model. Figure 2 depicts the related flowchart. When the model has been learned, the cloud layer sends it to the edge nodes.

4.2 Edge reinforcement learning methods

In the smart grid, although users often have similar electricity consumption habits, they also have their characteristics (Xu et al., 2022). Using user privacy data to train prefabricated models at edge nodes not only ensures the security of user privacy but also ensures the accuracy and efficiency of prediction results. We adopt the reinforcement learning method for model secondary training. We define the state space, action space, and reward function of the edge RL model as follows:

4.2.1 State space

The state of an edge node agent consists of its requirements. For user i , its requirement $R_{min} \leq R_i \leq R_{max}$. Then the state vector is expressed as: $[R_1, R_2, \dots, R_N]$, and N is the total equipment of user i .

4.2.2 Action Space

To meet the needs of different users, we consider edge nodes to divide devices into different categories at each decision epoch. Therefore, after determining the predicted device type, the edge node determines the activity set for deriving a class of devices.

4.2.3 Rewards

We use rewards to improve the accuracy of model predictions to meet device demands. We define the immediate reward received by edge nodes as $R_{i,d} = R_{max,d} - R_{real,d}$. Among them, $R_{max,d}$ is the maximum reward that can be obtained by

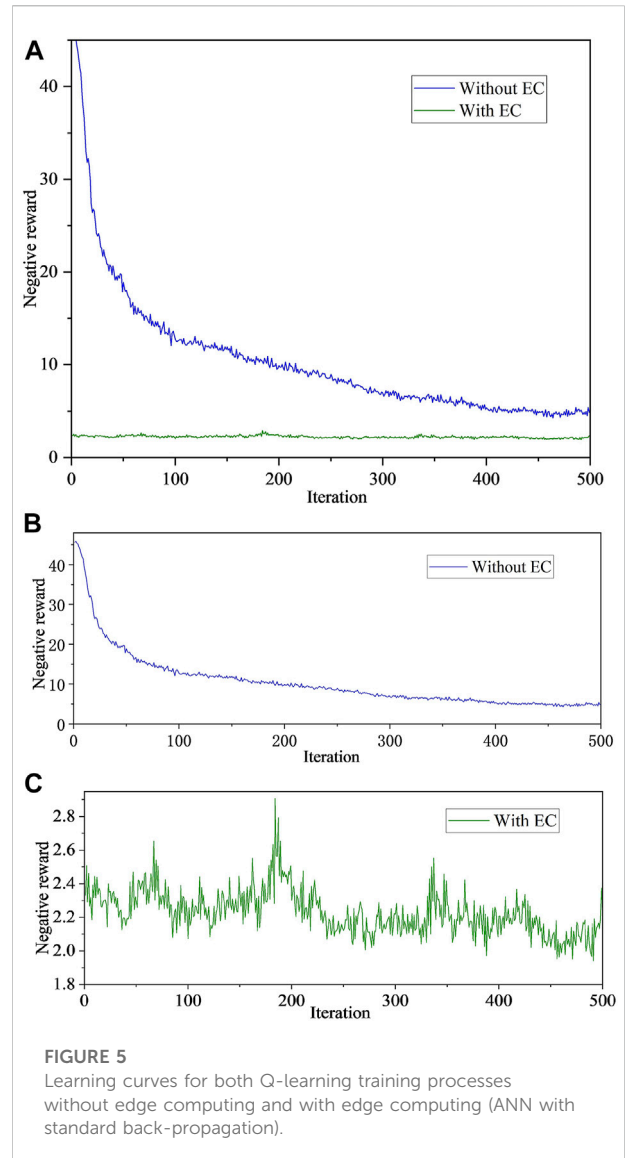


FIGURE 5
Learning curves for both Q-learning training processes without edge computing and with edge computing (ANN with standard back-propagation).

estimating the power consumption of device d , and $R_{real,d}$ is the real reward obtained by estimating the actual power consumption of device d , and the actual score can be obtained after each state estimation.

To train the model, we utilise the Q-learning approach, as illustrated in Figure 3, which is stated as follows:

Step 1. The edge nodes calculate the Q function values of the eight models based on the parameters of the prefabricated models in each cycle.

Step 2. Choose the action to conduct for a specified class of devices using the ϵ -greedy technique. In this stage, the action with the highest estimated Q-value is chosen with probability $(1 - \epsilon)$, and the action is chosen at random with probability.

Step 3. Based on the score, the edge device picks the best model as the device's model and updates the associated parameters to produce the best solution.

Step 4. The edge node obtains the load prediction result according to the optimal solution, and performs energy management.

Step 5. The device class state transition function information will be stored in the node's historical information and used as an alternative state for the next state time step after the immediate reward is observed.

Step 6. After the decision cycle is over, the state transition is used to obtain the loss function of the edge node to update the model parameters and perform model selection.

In all tests in this paper, the maximum number of iterations N is set to 50,000, the learning rate α is set to 0.1, discount factor γ is set to 0.95, the round iteration number n is set to 30.

Input: Pre-trained models [M1, M2, ..., M8], maximum number of iterations N , Learning rate α , discount factor γ , Round iteration number n
Output: The selected optimal model and its parameters
1. Initialize Q, input M_i
2. for $i = 1$ to 8 do
3. With the probability of ϵ select a random action a
4. With the probability of $(1 - \epsilon)$ select $a = \operatorname{argmax} Q(s, a)$
5. Calculate R
6. Update Q by $Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[R(s, a) + \gamma Q(s, a)]$
7. $\epsilon = \epsilon - 1/n$
8. end for
9. for $l = 1$ to N do
10. Take action $a' = \operatorname{argmax} Q'(s', a')$
11. end for

Algorithm 1. Q-learning model selection and parameter update algorithm.

5 Case study

5.1 Experiment described and evaluation index

We test the effectiveness of our proposed method on data from one of four anonymized climate zones in the United States. The dataset consists of four sets of data, each from nine buildings¹. Each building contains information on DHW demand, electrical power consumption, solar origin, and other variables. Research data is based on hourly information as a

research sample. We take all the data of each group as global data and each building as the data of an edge node. We divide the data into global training set data, edge training set data, and validation data set, of which the global training set data accounts for 50%, the edge training set data accounts for 20%, and the validation set data accounts for 30%. In the simulation, we consider the energy forecast as the target indicator. The statistical characteristics of the research sample (taking one year's data for one of the buildings as an example) are shown in Table 2. The load of the building on a certain day and the time statistical data characteristics of the four key variables are shown in Figure 4. On this day, the load is mostly weather, DHW, and cooling, not heating, which may be a local feature.

The model evaluation is represented by the deviation between the actual value and the predicted value, the deviation value D is:

$$D = \frac{|x_{real} - x_{eval}|}{x_{eval}} \quad (1)$$

where x_{real} is the actual value, x_{eval} is the predicted value. We use D -value changes to assess overall forecast persistence.

5.2 Algorithm validation test

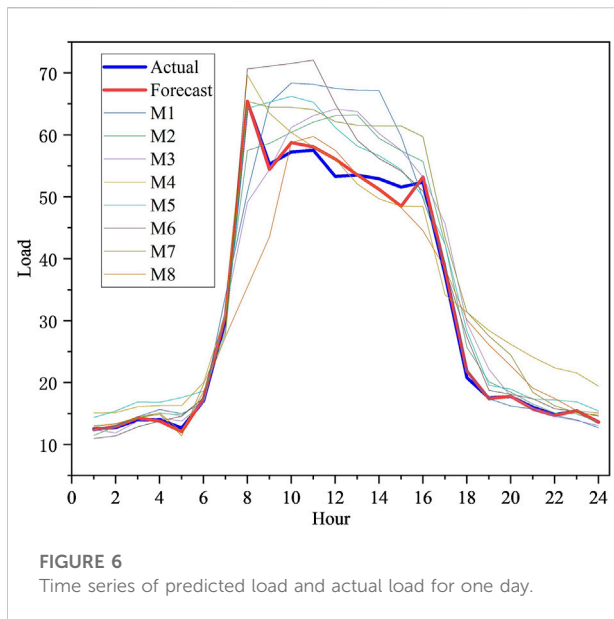
We first tested the effectiveness of the algorithm. We use ANN with standard back-propagation to compare the convergence efficiency of the algorithm, nodes directly using reinforcement learning training and the convergence efficiency of our method, as shown in Figure 5.

It can be seen from Figure 5 that under our method, the convergence speed of edge nodes is faster and the effect is better. This proves that our designed model guarantees the effectiveness and efficient convergence of Q-learning.

We next verified the accuracy of the predictions. We use data from a certain climate region as an example to evaluate and test the validity of the model. The climate zone contains 9 buildings, that is, there are 9 edge nodes, and we use the prediction result of one of the nodes to prove the accuracy of our model prediction. Figure 6 shows the time series of actual and predicted loads for a day. Specifically, we show the prediction results of 8 prediction models and the prediction results finally adopted, and the edge nodes tend to choose the best model results as the prediction results. In Figure 6, the thin line represents the model prediction results in the model pool, the thick blue line represents the actual load, and the thick red line represents the predicted load result.

The experimental findings in Figure 6 show that our proposed method may fully use each model's benefits, increase overall forecast accuracy, overcome the constraints of the present smart grid forecast model, and give direction for decision-makers in the smart grid's long-term growth. From Figure 6 it can be seen that, because of their high generalization capacity, machine learning approaches can integrate external information such as meteorological data. deep learning

¹ <https://sites.google.com/view/citylearnchallenge/previous-edition-2020?authuser=0>.



approaches have also garnered a lot of interest in the prediction and management of smart grids. Compared with other algorithms, the predicted load value curve and the real load value curve of the proposed algorithm on the used power load data set can be close or even coincide at most times. For the sample points that do not fit well with some single models, the fitting degree of the model proposed in this paper has been significantly improved. The reason comes from the fact that machine learning methods and deep learning methods have the issues that are easy to fall into local optimum, overfitting, and low convergence speed. Compared with other methods, in our method, a model pre-training pool is proposed to reduce the computational load of edge nodes and improve the accuracy of node prediction; a pre-training model for secondary training of edge nodes is proposed, the optimal model is selected and the model parameters are updated to achieve short-term load forecasting. Finally, an optimal optimization method for multi-model selection is proposed to improve the prediction accuracy.

6 Conclusion

The transition from renewables to energy and power systems can improve a city's sustainability goals. However, the volatility and intermittency of renewable energy and the privacy and personality characteristics of users make short-

term load forecasting models challenging. This paper develops a short-term load forecasting method for the smart grid based on edge computing. We first build a model pool on the central server and perform model pre-training to reduce the training volume and accuracy of edge nodes. We use edge nodes to retrain the pre-trained model, select the optimal model and update the model parameters to achieve short-term load forecasting. We use the model pre-training pool to reduce the training difficulty of edge nodes and adopt the method of secondary training of edge nodes to ensure the validity of the model and the security of private data. The experimental results show that the method proposed in this paper can make full use of the advantages of each model, overcome the limitations of the current smart grid forecasting models, and can provide guidance for decision-makers in the sustainable development of smart grids.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

A-XW designed the algorithm and wrote the paper. J-JL designed the overall framework of the paper and reviewed the content of the paper.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Chen, X., Song, C., and Wang, T. (2021). Spatiotemporal analysis of line loss rate: A case study in China. *Energy Rep.* 7, 7048–7059. doi:10.1016/j.egy.2021.09.116
- Dong, J., Song, C., Zhang, T., Li, Y., and Zheng, H. (2022). Integration of edge computing and blockchain for provision of data fusion and secure big data analysis for internet of things. *Wirel. Commun. Mob. Comput.* 2022, 1–9. doi:10.1155/2022/9233267
- Feng, C., Cui, M., Hodge, B. M., and Zhang, J. (2017). A data-driven multi-model methodology with deep feature selection for short-term wind forecasting. *Appl. Energy* 190, 1245–1257. doi:10.1016/j.apenergy.2017.01.043
- Feng, C., Wang, Y., Chen, Q., Ding, Y., Strbac, G., and Kang, C. (2021). Smart grid encounters edge computing: Opportunities and applications. *Adv. Appl. Energy* 1, 100006. doi:10.1016/j.adapen.2020.100006
- Kotsiopoulos, T., Sarigiannidis, P., Ioannidis, D., and Tzouvaras, D. (2021). Machine learning and deep learning in smart manufacturing: the smart grid paradigm. *Comput. Sci. Rev.* 40, 100341. doi:10.1016/j.cosrev.2020.100341
- Li, C. (2020). Designing a short-term load forecasting model in the urban smart grid system. *Appl. Energy* 266, 114850. doi:10.1016/j.apenergy.2020.114850
- Liu, Y., Yang, C., Jiang, L., Xie, S., and Zhang, Y. (2019). Intelligent edge computing for IoT-based energy management in smart cities. *IEEE Netw.* 33 (2), 111–117. doi:10.1109/mnet.2019.1800254
- Luo, H., Cai, H., Yu, H., Sun, Y., Bi, Z., and Jiang, L. (2019). A short-term energy prediction system based on edge computing for smart city. *Future Gener. Comput. Syst.* 101, 444–457. doi:10.1016/j.future.2019.06.030
- Meliani, M., Barkany, A. E., Abbassi, I. E., Darcherif, A. M., and Mahmoudi, M. (2021). Energy management in the smart grid: State-of-the-art and future trends. *Int. J. Eng. Bus. Manag.* 13, 184797902110329. doi:10.1177/18479790211032920
- Samie, F., Bauer, L., and Henkel, J. (2019). Edge computing for smart grid: An overview on architectures and solutions. *IoT smart grids* 2019, 21–42. doi:10.1007/978-3-030-03640-9_2
- Song, C., Han, G., and Zeng, P. (2021). Cloud computing based demand response management using deep reinforcement learning. *IEEE Trans. Cloud Comput.* 10 (1), 72–81. doi:10.1109/tcc.2021.3117604
- Song, C., Jing, W., Zeng, P., Yu, H., and Rosenberg, C. (2018). Energy consumption analysis of residential swimming pools for peak load shaving. *Appl. Energy* 220, 176–191. doi:10.1016/j.apenergy.2018.03.094
- Song, C., Liu, S., Han, G., Zeng, P., Yu, H., and Zheng, Q. (2022). Edge intelligence based condition monitoring of beam pumping units under heavy noise in the industrial internet of things for industry 4.0. *IEEE Internet Things J.* 2022, 1. doi:10.1109/JIOT.2022.3141382
- Song, C., Sun, Y., Han, G., and Rodrigues, J. J. (2021). Intrusion detection based on hybrid classifiers for smart grid. *Comput. Electr. Eng.* 93, 107212. doi:10.1016/j.compeleceng.2021.107212
- Song, C., Xu, W., Han, G., Zeng, P., Wang, Z., and Yu, S. (2020). A cloud edge collaborative intelligence method of insulator string defect detection for power IIoT. *IEEE Internet Things J.* 8 (9), 7510–7520. doi:10.1109/jiot.2020.3039226
- Song, C., Xu, W., Wu, T., Yu, S., Zeng, P., and Zhang, N. (2021). QoE-driven edge caching in vehicle networks based on deep reinforcement learning. *IEEE Trans. Veh. Technol.* 70 (6), 5286–5295. doi:10.1109/tvt.2021.3077072
- Taik, A., and Cherkaoui, S. (2020). Electrical load forecasting using edge computing and federated learning. ICC 2020-2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, June 7–11, 2020. IEEE, 1–6.
- Wang, Y., Bennani, I. L., Liu, X., Sun, M., and Zhou, Y. (2021). Electricity consumer characteristics identification: A federated learning approach. *IEEE Trans. Smart Grid* 12 (4), 3637–3647. doi:10.1109/tsg.2021.3066577
- Xu, J., Li, Z., Gao, L., Ma, J., Liu, Q., and Zhao, Y. (2022). A comparative study of deep reinforcement learning-based transferable energy management strategies for hybrid electric vehicles. *arXiv preprint arXiv:2202.11514*.
- Yin, L., and Xie, J. (2021). Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems. *Appl. Energy* 283, 116328. doi:10.1016/j.apenergy.2020.116328
- Zhang, X., Huang, C., Gu, D., Zhang, J., Xue, J., and Wang, H. (2022). Privacy-preserving statistical analysis over multi-dimensional aggregated data in edge computing-based smart grid systems. *J. Syst. Archit.* 127, 102508. doi:10.1016/j.sysarc.2022.102508