International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-023-01796-8

ORIGINAL ARTICLE

Enhanced neighborhood node graph neural networks for load forecasting in smart grid

Jiang Yanmei[1,3] · Liu Mingsheng[1] · Li Yangyang[2] · Liu Yaping[3] · Zhang Jingyun[4] · Liu Yifeng[2] · Liu Chunyang[5]

Received: 30 July 2022 / Accepted: 27 January 2023

Abstract
Deep learning technology creates the condition for the optimization of the smart grid, and the big data analytical technique has the most efficient way to analyze and share the power load spatio-temporal data in the smart grid. Utilizing the graph-based method to learn the structure of load date distribution and load prediction has become hot-spot research. This paper proposes EnGAT-BiLSTM, an enhanced graph neural networks framework to realize short-term load prediction. The EnGAT-BiLSTM model aims to improve the prediction accuracy of the load and solve the sampled data sparsity in the short-term prediction. In this model, the Box-Cox transformation technology is used to denoise and obtain the effective load sampled data set; a dynamic load knowledge graph (DLKG) is designed to map the internal attribute of the various electrical load and the correlation of the external influencing factors; the graphic attention mechanism is introduced in the local network structure of graph neural network (GNN) to extract the high-quality load spatio-temporal features; the deep bi-directional long short-term memory (BiLSTM) framework is used for the lifelong learning of the load prediction. The extensive load-sampled datasets in the real world are employed to evaluate our method. The experimental results indicate that EnGAT-BiLSTM improves significantly in load prediction accuracy and has good robustness. The model will provide a valuable theoretical basis and guidance for the smart grid IoT system.

Keywords Load power prediction · Lifelong learning · Graph neural network · Box-Cox transformation · Multi-attention mechanism

1 Introduction

With the rapid construction of the smart grid, the distributed energy IoT system generates massive spatial-temporal load data and forms the power load data cyberspace [1]. Utilizing the intelligent data analysis technology to analyze and process the power load data effectively can bring the following benefits: to improve production efficiency and realize the coordinated development of all parts in IoT; to regulate load demand-side energy consumption by digging deep into the

✉ Liu Yaping
yapingliu_82@163.com

Jiang Yanmei
weiyi_jim@163.com

Liu Mingsheng
liums601001@sina.com

Li Yangyang
liyangyang@cetc.com.cn

Zhang Jingyun
zhangjingyun@buaa.edu.cn

Liu Yifeng
liuyifeng3@cetc.com.cn

Liu Chunyang
liuchunyang@didiglobal.com

1 State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, No. 5340 Xiping Road, Beichen District, Tianjin 300401, China

2 National Engineering Research Center for Risk Perception and Prevention, CAEIT, Wanshou Road, Haidian District, Beijing 100846, China

3 Department of Rail Transportation, Hebei Jiaotong Vocational and Technical College, No.219 Zhujiang Avenue, Shijiazhuang 050035, China

4 School of Cyber Science and Technology, Beihang University, No. 37 Xue Yuan Road, Haidian District, Beijing 100191, China

5 Didi Chuxing, No. 8 Dongbei Wang Road, Haidian District, Beijing 100193, China

growing electricity consumption demand for users. Load prediction is a crucial part of demand-side management in the smart grid, and accurate daily load prediction can be a valuable reference for intelligent management and optimal control of the smart grid [2, 3]. For families, accurate load prediction can adjust their daily power consumption habits effectively and to save electricity. Furthermore, the effective load prediction of the actual electrical equipment can alleviate the energy crisis and optimize the feedback mechanism for energy scheduling [4].

Generally, load prediction mainly includes ultra-short-term, short-term, medium-term, and long-term predictions. Specifically, the ultra-short-term load prediction indicates the load prediction within 1 h; the short-term mainly shows the daily and weekly load prediction; the medium-term ranges from months to years; and the long-term refers to the load prediction in the next 3 to 5 years or even longer [5, 6]. As the power system scale increases, the type of load data is more complex and diverse in the smart grid, and the existence of various external influencing factors aggravates the randomness and nonlinear of the power load [7]. Short-term load prediction is significant in the smart grid, which is responsible for improving the machine performance, ensuring the water-fire and electricity coordination, and as a basis for drawing up the technical solution [8, 9].

Research on short-term load forecasting has become the research hotspot, and in the previous literature, improving the accuracy and performance of the model becomes an urgent requirement [10]. Currently, the existing short-term load prediction methods mainly include the based time series and the based data-driven models. The former takes the time as the basis to analyze the law of load change, such as the differential autoregressive sliding mean model (ARIMA), the autoregressive moving average model (ARMA), and the moving average model (MA). In [11], the paper combines the generalized autoregressive conditional heteroskedasticity (GARCH) and the autoregressive integrated moving average (SARIMA) to realize short-term load forecasting, but the complex parameter adjustment leads to limited prediction accuracy. Constructing the mathematical model of similar months to predict the residential electricity load, but the input parameters setting is relatively complicated, which is challenging to achieve the model optimization [12]. For the optimal parameter selection, an improved cuckoo search (CS) algorithm is added to Fractional Auto-regressive Integrated Moving Average (FARIMA), which can improve the prediction accuracy by the dynamic adjustment [13]. Fusion algorithm is becoming a new idea for improving the accuracy of the model, such as utilizing the statistical method to realize the optimization of the time-series prediction model [14]; the author uses the grey wolf optimizer (GWO) can effectively solve the nonlinear relationship of sampled data, but the correlation of time series data is not considered. Unfortunately, the time-series prediction models have the following problems: the limited ability to process data and low efficiency; model parameter setting is tedious and high time complex; online modeling has insufficient commonality.

With machine learning technologies (MLT) being used extensively in industries, data-driven methods are gradually used to predict and analyze load data, such as random forest (RF) [15], relevance vector machine (RVM), support vector machine (SVM) [16], and deep learning algorithms (DLA). In paper [17], the SVM algorithm is used for solving the convex quadratic programming to realize the load prediction; the SVM model with a variable input structure is designed for load prediction, but the high-dimensional input variables lead to the computational burden. The article [18] employs the chaos theory and the particle swarm algorithm to optimize the SVM algorithm. In [19], an integrated RF model with the grey catastrophe theory is constructed to reduce the randomness of the load breakpoints, and then the experimental results prove that SVM has a noticeable optimization effect in optimizing the model performance. The RF method is introduced into the kernel-based principled component analysis to extract the load features and enhance the performance of the load extractor, which is more advantageous for improving the model precision [20].

As a typical algorithm of MLT, DL technology has a wide range of applications in the load data cyberspace, such as analysis, processing, and forecasting. In some research, to reduce the prediction error, the abstract feature information of load and the historical load sampled data of the distributed systems are inputted into the training model, and then the general performance of the load model is enhanced [21–23]. In [24], an ANN model based on Fuzzy-ARTMAP (FAM) is presented to achieve load monitoring by an online data acquisition system; the simulation experiment proves the performance of FAM-ANN is more stable. Considering the LSTM model has advantages in the time-series prediction. In [25], a combined model BiGRU-CNN is presented to predict the short-term load, and the experimental result shows that this method has a remarkable prediction. [26] proposes a CNN-BiLSTM framework based on Bayesian optimization to realize the optimization of the model parameters; finally, the paper employs an attention mechanism to perfect the neural network representation. In [27], a DL model based on the pooling LSTM and CNN is presented to address the model nonlinear, and the error of the model prediction is minimum. In [28], an advanced LSTM model is used for middle-term electric load forecasting, and exponential smoothing (ETS) is employed to learn the dynamic seasonal relationships of load time series, enhancing the model interpretability.

However, the methods mentioned above are only for analyzing and predicting the electricity demand of load cyberspace in a single time scale and ignore the heterogeneous characteristics of the distributed power system in the multi-time scale data space. These are not conducive to the generalization of the load forecasting model. Meanwhile, the correlation of electricity consumption in different time series is significant for the power system's production scheduling and operation management [29–31]. In [32], the author considers that improving the time dependence between consecutive loads can effectively improve the prediction accuracy of multi-energy loads in energy systems, which is the basis for energy control and scheduling. In [33], the author proposes the analysis and prediction of the key characteristics in the multi-scale time-series energy load consumption to improve the energy efficiency in the energy system. Furthermore, on the one hand, the existing short-term load forecasting models need to be revised to solve the high volatility of user power demand behavior and the complexity of the actual electricity load. On the other hand, ignoring the influence of external factors such as weather, holidays, and temperature on the user's power consumption behavior tends to cause the prediction model's poor stability and slow convergence [34, 35].

To address the above issues, we propose an enhanced graph neural network based on the spatial attention mechanism and BiLSTM framework (EnGAT-BiLSTM), which uses the attention modules in the GNN framework to improve the learning ability of the load features representation in the heterogeneous data cyberspace. EnGAT-BiLSTM aims to the analytical prediction of the power load in the short-term and then to achieve fine-grained learning for the load features with unknown, random, and nonlinear characteristics; to enhance the representation and interpretability of load features; to improve the load prediction model accuracy and the model performance with the limited load sampled data sets. We survey 500 families and find that the complexity of the electrical equipment and the diversity of consumer habits lead to the heterogeneous structure of the power load cyberspace, which challenges the existing load forecasting model on a single scale. Meanwhile, equipment measurement errors and noise are unavoidable in the raw sampled data, affecting the prediction accuracy. Furthermore, it is easy to waste resources and time to train and analyze the model with many noisy data. To this end, we use the Box-Cox transformation technology to eliminate the nonlinearity noise of the raw load data and improve data validity. Box-Cox transformation technology introduces the parameters to create a functioning family with the monotone transformation for denoising and data dimensionality reduction. The method has been mentioned in other fields for the forecasting model [36–38].

We design a dynamic heterogeneous load knowledge graph structure (DLKG) to capture the internal correlation of the different power loads and to model the external correlation of the load data with the influencing factors. For example, the internal load correlation includes the power estimation of the various electrical loads, service time, load time-domain waveform, and frequency-domain image; the external factors information of load, such as weather, temperature, holidays, seasons, etc., is a few literature mentions. After that, we utilize an enhanced neighborhood node graph neural network with a multi-layer attention module to improve the efficiency of the vital feature learning in the load extraction and to enhance the load interpretability. Finally, the deep BiLSTM network is used to deal with the temporal dependency issues, realize the accurate load prediction, and build the lifelong learning model.

The paper makes significant technical contributions as the following:

- It utilizes the Cox-Box transformation algorithm to reduce the nonlinearity of the load sampled data and to enhance the data validation;
- It constructs a dynamic load Knowledge graph structure (DLKG) to store the power load feature relation in a graph way and to improve the load interpretability;
- It proposes an ENGAT model based on a graph attention mechanism to enhance the learning ability of the load time-series feature representation;
- It structures a deep BiLSTM framework for load lifelong learning and prediction.

Other sections are arranged in the following: review research background and related work are presented in Sect. 2; important algorithms are described in Sect. 3; experimental implementation and results discussion in Sect. 4; the conclusion is in Sect. 5.

2 Review research background and related work

2.1 Load forecasting methods

Load forecasting is beneficial to optimize the real-time scheduling of the smart grid and improve the reliability of the power system. It is more important to establish the correlation between the external factors (the weather, the economic development, and the holidays) and the load sampled data as the knowledge of load relations for improving the prediction accuracy in the short-term load. There is a great deal of research indicating that DL technology has become the research hotspot. [39] proposes a hybrid model based on extreme gradient boosting and pattern sequence-based matching for forecasting the holiday load and then utilizing the shape similarity strategy based on Euclidean distance to compare the predicates effect. In [40], the author presents a wavelet-based neural network to enhance the prediction model performance, improve the convergence rate, and avoid local optimization. Considering the learning ability of the load feature model can influence the effect of the prediction model. [41] designs a model based on kernel principal component analysis (KPCA) and BiLSTM neural networks to realize the multi-load prediction; KPCA is used to extract the principal components of the weather and calendar rule feature data sets for reducing the load sampled data dimension. [42] presents an adaptability of the enhanced deep neural networks (DNN) model based on a deep reinforcement learning framework to realize the good fitting and tracking of the prediction model network. However, these existing methods still have the following challenges: the nonlinearity of load sampled data results in the low accuracy of load feature extraction; many layers and hyperparameters lead to model complexity and over-fitting; multiple load forecasting has poor stability and accuracy. Improving the quality of load representation to enhance the interpretability of load features is still an urgent issue in load forecasting [43, 44].

2.2 Graph-based representation learning

GNN has a powerful ability to process complex structural data, especially its significant advantages in representation learning [45–47]. GNN converges the state of the neighborhood nodes to update the state of the current node. GNN structure can efficiently solve the following problems: the limitation of the node updates and end-to-end learning in the DL network model; the node heterogeneity in the different types of neighbor nodes on the destination node [48–50]. GNN has been widely used in computational science and biology but is rarely applied to the representative learning of the load data cyberspace. [51] proposes a GNN structure to extract the load features of each power node and use the pattern similarity graph based on the model (PSGM) to forecast the short-term load of the smart grid; the method shows good learning of the feature representation. In [52], GNN is presented to extract the high-dimensional feature of the load data with a graph-structured in the smart grid, and the GNNs framework can effectively solve the interdependency among nodes of the load graph structure. In [53], the author employed GNN to restructure the sampling time-series data of the wind power load in the form of a graph, and then GNN can enhance the spatio-temporal correlation of load nodes. [54] proposes a dynamic GNN model to realize the real-time traffic flow prediction, and the method utilizes the graph structures to represent the spatial features of the dynamic flow more effectively; the induction of the LSTM structural unit improves the generalization performance of the prediction model. [55] uses a reinforced multi-relational graph-structured to achieve graph representation learning and to improve the discrimination and interpretability of graph learning. In [56], authors design an optimal GNN model to capture the information of the brain network structure features and the related function modes and to realize the autonomic prediction. Comparing GNNs with the DNN model, the GNNs framework has a significant advantage for learning the load data based on time series. However, the GNN representation learning in load data cyberspace for load forecasting is still challenging because of the lack of integrity in load sampled data and the highly complex temporal correlation of the real-world load data with the external factors. So, it is crucial to improve the representation quality.

3 Our Proposed Algorithms

3.1 Overview

The overall design framework of our model is presented in Fig. 1. The model includes four parts: 1) the Box-Cox transformation algorithm to perform the de-noising and preprocessing of the raw load data; 2) DLKG is presented as a heterogeneous graph structure to represent the dependency of the various loads, the correlation between load and the external influencing factors from the historical load data; 3) using an encode-decode model based on the multi-layer attention module to extract the spatial-temporal features of load and to reconstruct the load features mapping with the graph structure, which can enhance the interpretability of load and improve the abstract representation ability of load; 4) an EnGAT structure is used for learning the graph-based load spatio-temporal features and then utilizing the BiLSTM framework to achieve the load prediction.

Problem setting Formally, DLKG is defined as G_d, which is from the historical load data of the household electricity consumption and the internal-external correlate factors G_l. Given the historical load data sets $\{x_{hl}^1, x_{hl}^2, \dots, x_{hl}^n\}$, where x_{hl}^n denotes the historical record of load, $n \in N$ denotes the number of the load data sets. We use the historical load data sets and DLKG to predict the future trend of the short-term load y^{*t} at time t, and the formula is described as follows:

$$y^{*t} = g(\{x_{hl}^1, x_{hl}^2, \dots, x_{hl}^n\}, G_l, G_d), \tag{1}$$

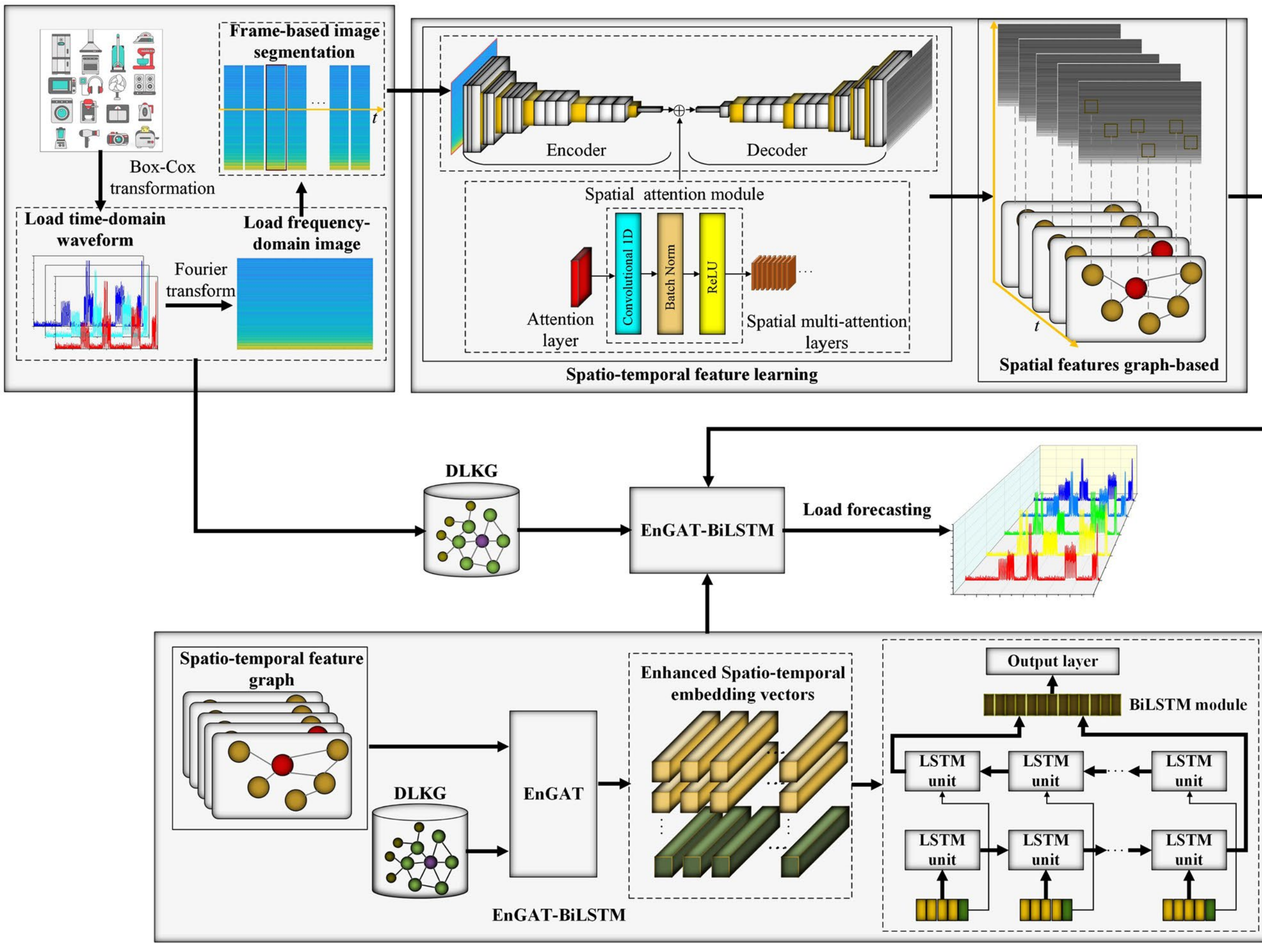


Fig. 1 The overall design structure of our model

3.2 Box-Cox transformation algorithm

In short-term load forecasting, the load sampled data is usually characterized by nonlinear lines and high dimensions affected by internal and external factors, such as the power consumption habits of users, meteorological factors, time factors, and so on. Meanwhile, almost inevitably, there are many measurements and other operational errors in the sampling process. The Box-Cox transformation algorithm can reduce the data nonlinearity by introducing a simple parameter to achieve dimensionality reduction and denoising, improving data utilization [57]. This paper uses the Box-Cox transformation technology to promote the validity of the load data. The mathematical expression of the Box-Cox transformation algorithm is as follows:

$$\widetilde{X} = \begin{cases} \frac{(X)^{\lambda_l^s}-1}{\lambda_l^s}, \lambda_l^s \neq 0 \\ \log X, \lambda_l^s = 0 \end{cases}, \tag{2}$$

where $\widetilde{X}$ is the historical load observation value after conversion, and λ_l^s represents the transformation coefficient. The maximum likelihood estimation method is generally used to solve parameter λ_l^s, and the specific solution process is equivalent to find λ_l^s, which maximizes the following mathematical expression as the final transformation coefficient:

$$Z^*(\lambda_l^s) = \frac{n}{2}\log[\hat{\sigma}^2(\lambda_l^s)] + (\lambda_l^s - 1)\sum_{i=1}^{n}\log(X_i), \tag{3}$$

where $Z^*(\lambda_l^s)$ is the logarithmic natural function; n is the sample number of the historical load observations; the expression of $\hat{\sigma}(\lambda_l^s)$ is as the following:

$$\hat{\sigma}^2(\lambda_l^s) = \frac{[X - K\hat{\beta}]^T[X - K\hat{\beta}]}{n}, \tag{4}$$

where $\boldsymbol{X}$ is the observation vector of the historical load data; $\boldsymbol{K}$ is the number vector of sampling point corresponding to the historical load data; $\hat{\beta}$ can be expressed as

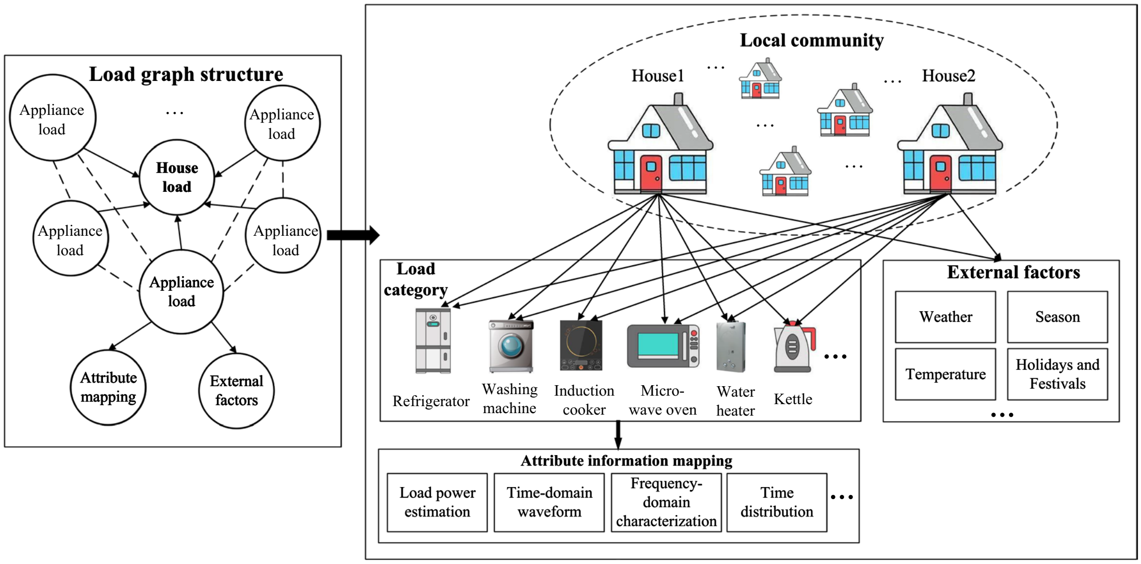


Fig. 2 Detailed design of DLKG

$$\hat{\beta} = (K^T K)^{-1} K^T X. \tag{5}$$

3.3 Dynamic load knowledge graph structure

We design a dynamic load knowledge graph structure (DLKG) to represent the various types of electrical loads and the attributed relational graph (load internal attribute and the external influence factors) in the historical load data. The specific description of DLKG is shown in Fig. 2. In DLKG, the various loads (i.e., refrigerator, washing machine, induction cooker, etc.), the internal attributes of load (i.e., load power estimation, time-domain waveform, frequency-domain characterization, etc.), and the external influence factors (weather, season, temperature, festivals, etc.) are as the nodes, the edges represent the correlations between nodes, which is inspired by [58, 59]. As illustrated in Fig. 2, We set up a relationship sets between the different loads, the dependency relationship between the internal attributes and loads, and the restricted relationship between load and the external factors, which are the critical information for load representation learning.

In our paper, we define the historical load data set of each house as the following: $P = \{p_{w1}, p_{w1}, \ldots, p_{wn}\}$, where p_{wn} indicates the power of the n^{th} load, $n = \{1, 2, 3, \ldots, N\}$, N indicates the total of the household appliances load. The internal attributes information of each electrical load is described K_w, and $K_w = \{k_e, k_{dw}, k_{fi}, k_{td}\}$, where, k_e, k_{dw}, k_{fi}, and k_{td} present the load power estimates, load time-domain waveform, load frequency-domain image, and working time, respectively. The external influencing factors $\lambda = \{\lambda_w, \lambda_c, \lambda_s, \lambda_{hd}\}$, where λ_w, λ_c, λ_s, and λ_{hd} respectively indicates the weather, temperature, season, holidays and festivals.

Considering the time dependence between the power loads of the appliances, and the correlation between the load and the internal attributes information, the correlation between the load and the external influencing factors in DLKG. We draw from the representation method of the heterogeneous information networks (HIN) to describe the complex relational data structure of DLKG, and the specific graph structure data description is shown in Fig. 3. In DLKG, we set up three sets: the load category (LC) $\{P\}$, the internal attributes information (IAM) $\{K_w\}$, and the external influencing factors (EIF) $\{\lambda\}$ by sampling and analyzing the historical load data. Each node of the load graph structure is from the three sets, and the edges include the load internal attribute relationships (i.e., the refrigerator has a time-domain waveform and time distribution) and the external correlations (i.e., daily electricity load is affected by weather).

We construct a graph G_p for DLKG to abstractly describe the loads and the relationship (the internal attributes information and the external influencing factors), $G_p = \{V_p, \varepsilon_p, w_p\}$, where V_p indicates the nodes set of the graph (electrical loads), ε_p indicates the edge set of the graph (the correlation of loads), w_p is the relevance weight, which reflects the degree of correlation between nodes. $V_p \times V_p \rightarrow \varepsilon_p$, $\varepsilon_p \in \{0, 1\}$, and 1 indicates an edge between the nodes V_p, otherwise ε_p is 0.

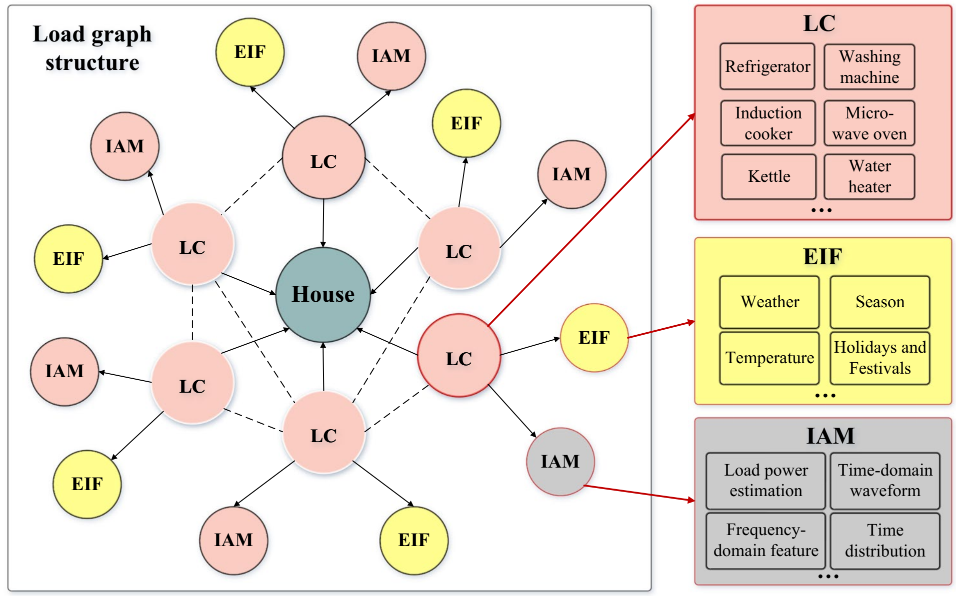


Fig. 3 Specific graph structure data description of DLKG

3.4 Extracting load feature based on frames structure

We design an encode-decode framework based on a spacial attention module for improving the performance of the load feature extraction model. The process of load features extraction is that each of the frame images is inputted to the encoded networks to form the feature maps; by the unpooling layer, the deconvolution layers, the rectification layers of the decoder, and the features map segmentation are generated; utilizing the spatial attention module to enhance the regions of the critical load features can improve the accuracy of the short-term load feature extraction.

This framework includes three parts: the encoder part is the extractor for the load features, the decoder part is to realize the segmentation of the load frame image region, and the spacial attention layer for enhancing the critical load features information and ignoring the minor information.

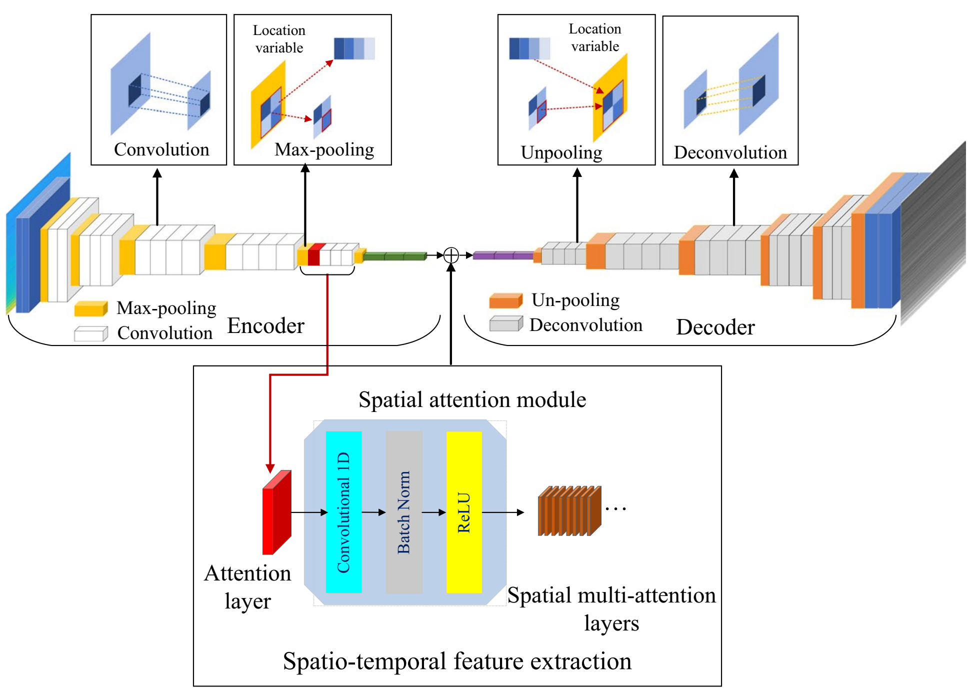


Fig. 4 Encode–decode framework based on a spacial attention module

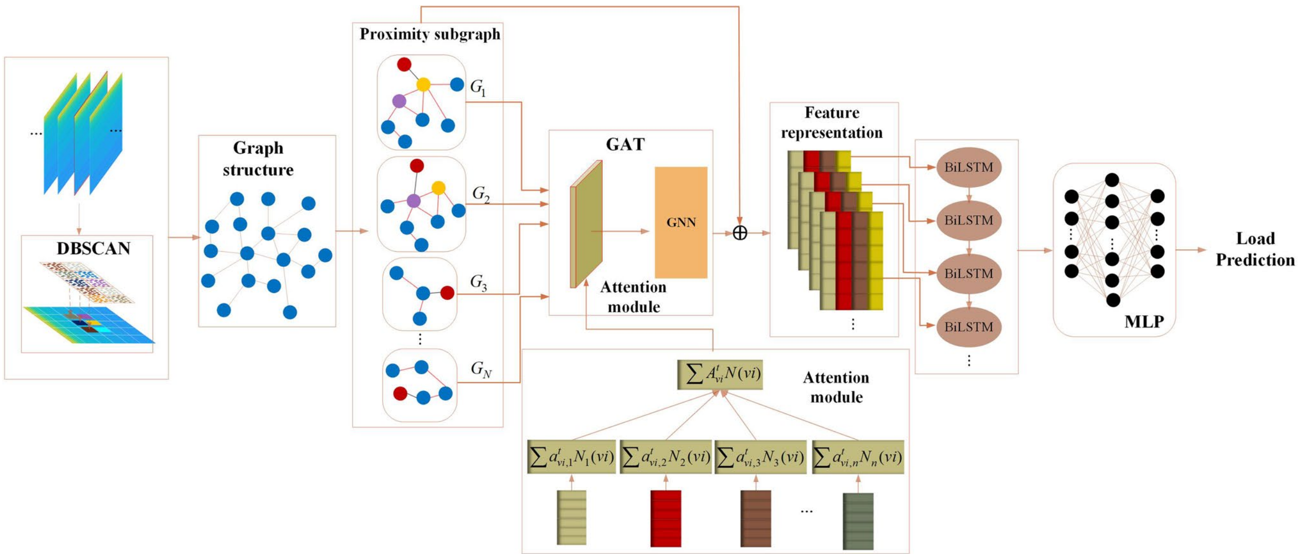


Fig. 5 The detailed design of EnGAT structure

The detailed structure of the extractor is shown in Fig. 4. In our paper, the improved VGG-19 networks [60–62] are employed as the extractor of load features in the encoder part. The decoder is the copy network of the encoder, but it performs the reverse operation of the encoder. The decoder networks include the unpooling layer, the deconvolution layers, and the rectification layers. The function of the unpooling operation is to restore and refactor the size of the original activation, and then a location variable is employed, which retains the place information after the max-pooling operation [63]. The deconvolution layers operation is to capture and splice the enlarged feature maps of each layer. In the encode-decode framework, we introduce the spatial attention module to pay more attention to the critical load feature with the spatial-temporal dimensionality. The spatial attention module includes a convolution layer with the size 1×1, a batch norm layer, and the activation function *ReLU*. By the 1D convolution operation, an unnormalized attention mapping is formed, and the multi-attention mapping layers are formed using a batch norm layer and *ReLU*.

3.5 Enhanced neighborhood GNN for graph representative learning

We design a novelty representative learning framework EnGAT, and the detailed design is shown in Fig. 5. In this framework, a multi-layer attention module is introduced into the GNN networks for the representation learning of the load sampled data. The design is inspired by graph structure learning, and it is more suitable than CNN for the feature representation of irregular and complex relations. In this specific framework, we utilize the strategy of enhancing local neighbor nodes to realize the updating and aggregation of the current node information and using a multi-layer attention module to maximize the feature discrimination of the single frame image.

Different from extracting every pixel of the frame image as a graph node, we use density-based clustering to divide the image into the small pixel regions as the graph nodes, which can improve computational efficiency [64]. The clustering algorithm based on the density, namely DBSCAN, is employed to structure the pixel regions [65]. The core of the DBSCAN algorithm is to create dense regions for all sample points, and these regions are treated as clusters. The algorithm implementation process includes: (1) to determine two parameters: the radius of the adjacent area around a point, *epsilon*, and the adjoining region contains at least the number of points, *minPts*; (2) a random point is selected, and its NBHD (*p*, *Epsilon*) is calculated to determine whether it is a nuclear point; (3) to create a class by iterating over the other sampling points; (4) the operation of steps 1 and 2 are repeated until all points are included in the class (core or edge points) or are peripheral points; (5) the algorithm is evaluated by the contour function, and then the optimal parameters are obtained. The specific algorithm is shown in Algorithm 1:

Algorithm 1 : DBSCAN: Density-based Clustering Algorithm

Input: Sampled dataset $LP = \{p_1, p_2, \dots, p_N\}$;
The number of data set is N.
Parameter: neighborhood radius θ, number of data objects threshold $MinPts$;
Output:dividing cluster $DC = \{DC_1, DC_2, \dots, DC_m\}$
m is the number of the dividing cluster.
Processing:
1: Initialization of the core object node set: $E = \phi$;
2: **for** $i = 1, 2, \dots, I\_N$ **do**
3: Determine the neighborhood $N_\theta(x_i)$ of sampled data x_i;
4: **if** $\mid N_\theta(x_i) \mid \geq MinPts$ then
5: Sample x_i is added to the core object node set: $E = E \cup \{x_i\}$
6: **end if**
7: **end for**
8: Initialization the cluster number: $I\_N = 0$;
9: Initialization the unaccessed data: $U\_N = D$;
10: **while** $E \neq \phi$ **do**
11: Records the unaccessed sample set: $U\_N_{ne} = U\_N$;
12: Random selection the core objects node $n_o \in E$, initializing the queue $N\_Q =< n_o >$;
13: $U\_N = U\_N/n_o$
14: **while** $N\_Q \neq \phi$ **do**
15: Take the first sample in the queue$q$;
16: if $\mid N_E(n\_q) \mid \geq MinPts$ then
17: Set $\tau = N_E(n\_q) \cap U\_N$
18: Add sample of τ to queue n_Q;
19: $U\_N = U\_N/\tau$;
20: **end if**
21: **end while**
22: $m = m + 1$, Generate cluster clusters $C_{I\_N} = \tau_{ne}/\tau$;
23: $E = E/C_m$
24: **end while**

Line1–7: according to the domain parameters $(\theta, MinPts)$ and to find all the core objects.

Line8–24: the algorithm takes any core object node as the starting and finds the reachable sampled node by this starting node density; to create the node cluster until all the core object nodes have been traversed.

Assuming that the size of the load frequency-domain image is $H \times W \times C$, the parameters represent the height, width, and channel, respectively. After the pixel clustering operation, the image is divided into M small pixel areas, namely, $A = \{a_1, a_2, \dots, a_m\}$, $m \in M$, M is the number of the pixel areas. Each small pixel area is inputted into the CNN model for the convolution and pooling operations. And then, each pixel area a_m is converted to the feature vector of the pixel areas $F(a_m)$, $x_m \in R^D$ and D is the dimension of the feature vector. The M small pixel areas can be formed into a graph $G(V, E)$ with M nodes, where the nodes $V = \{v_1, v_2, \dots, v_m\}$ and the edges e_{ij}. e_{ij} represents the correlation between nodes, such as e_{ij} indicates $V_i \rightarrow V_j$. The local neighbor nodes of a node $N(V_m)$ are represented as $N_u(V_m)$, $N_u(V_i) \in N_u(V_m)$. In this paper, we mainly utilize the important local information of the neighbor nodes to enhance the representation of the current node. Finally, the feature representation of graph $g^* = G^*(V_f, E_f)$ is obtained, where all the nodes set is V_f and all the edges set is E_f. The detailed design of the whole model is shown in Fig. 5.

3.6 Propinquity-based node density judgment mechanism

In the EnGAT module, we utilize the propinquity-based node density judgment mechanism (Top-pnd) to select the local neighbor nodes $N_u(v_m)$. The proximity of inter-nodes reflects the strength of the connection between two nodes. Considering the importance of the neighbor nodes to the information updating and transferring of the current node, however, the traditional methods rely on the random walk to judge the importance of nodes, which will impose an additional computational burden. The Top-pnd method is first described using mathematical symbols, which facilitates the reference and calculation in the following section.

Assuming that the node v_m and the set of the neighbor nodes $N_u(v_m) = \{u_1, u_2, \dots, u_m\}$, the neighbor nodes density $P\_N_u(v_m)$ is defined as following formula: $P\_N_u(v_m) = \frac{e_m}{\frac{m(m-1)}{2}}$, where e_m indicates the number of the existing node edges, $\frac{m(m-1)}{2}$ indicates the maximum number of the possible edges. The maximum $P\_N_u(v_m)$ is selected as the neighbor nodes of the node v_m. According to the proximity attribute strategy with the maximum density of the adjacent nodes, we select k neighbor nodes with the strongest proximity before the target node, which is the $top - k$ mechanism in the recommendation algorithm. The advancement and uniqueness of the algorithm are that selecting the adjacency nodes with the most significant local influence enhances the representation of nodes. So, the graph structure is divided into N subgraphs, which can be described by the mathematical theory as the following: $G = \{G_1 \cup G_2, \dots, G_N\}$, N indicates the number of the subgraphs.

In our paper, we introduce a multi-attention module into the GNN structure to improve the interpretability of graph representation. Meanwhile, using the multi-attention layers to improve the computing efficiency with parallel computing. The node features of the input network is described as $H^T = \{H_1^t, H_2^t, \dots, H_i^t\}$, where, $i \in (1, N)$, $H_i \in R^D$, N represents the nodes scale, D indicates the size of the node feature vector H_i^t, t indicates the time. Given $H^{*T} = \{H_1^{*t}, H_2^{*t}, \dots, H_i^{*t}\}$, $H_i^{*t} \in R^F$ represents the output network. The attention coefficient is a parameterized shared linear transformation, showing the influence of inter-nodes. The attention coefficient is defined as: $e_{ij}^t = a(WH_i^t, WH_j^t)$, e_{ij}^t represents the importance of the features from the node H_i^t to the node H_j^t at time t, and $H_j^t \in N_i$, N_i indicates the set of i neighborhood node, and the weight matrix is W, $W \in RD^* \times D$, which is used to convert the input features to the higher-level features. The $a(\bullet)$ is a self-attention mechanism of the nodes and a trainable parameter vector, and $RD^* \times RD^* \to R$, which is used to learn the importance between the nodes and the neighborhood. We use the function softmax to normalize the e_{ij}^t, the expression is shown as follows:

$$a_{ij} = softmax_j(e_{ij}^t)\frac{exp(e_{ij})}{\sum_{k \in N_i} exp(e_{ik})}. \tag{6}$$

The expansion of the formula above is as following:

$$a_{ij} = \frac{exp(LeakyReLU(\vec{a}^T[W_i^t\vec{H}_i^t \parallel W_j^t\vec{H}_j^t]))}{\sum_{k \in N_i} exp(LeakyReLU(\vec{a}^T[W_i^t\vec{H}_i^t \parallel W_j^t\vec{H}_j^t]))}, \tag{7}$$

where $LeakyReLU$ is the function, $[\cdot \parallel \cdot]$ represents the concatenation operation for the features of nodes H_i^t and H_j^t after transformation, W_i^t and W_j^t are the trainable weight matrix, which can realize the linear transformation of the input features. Finally, the output features vector of node are as following:

$$H_i^{t*} = \Vert_{t=1}^T \sigma(\sum_{j \in N_i} a_{ij}^t W^t H_j), \tag{8}$$

a_{ij}^t denotes the normalized attention coefficient calculated by the k^{th} attention mechanism. σ denotes the activation function, W^t refers to the weight matrix that corresponds to the input linear transformation. For the node, H_i^{t*} indicates the state structure of the hidden-layer network i^{th} at time t. In the paper, given a node v_0 and its neighborhood nodes $N_u(v_0) = \{v_1, v_2, \dots, v_m \mid v_0\}$, GAT is defined as a function f that mapping from each nodes of $N_u(v_0)$ to the relevance score, $f^* : v_0 \times N_u(v_0) \to [0, 1]$, and the relevance score represents the relative importance of these neighborhood nodes, which satisfies the following formula:

$$\sum_{i=1}^{|N_u(v_0)|} f^*(v_0, v_i) = 1. \tag{9}$$

3.7 BiLSTM framework for load forecasting

For short-term load forecasting, the longer the load sampling span, the weaker the performance and the poor convergence of the training model. To maintain the time dependence of the load sampled data, we employ BiLSTM architecture to address the low forecasting accuracy and instability performance of the prediction model. In the prediction tasks based on the time-series dependence, the particular gate structure of BiLSTM can link the previous information to the current task, for example, using the past video fragment to infer the understanding of the previous fragment. In the real-world scenario, the load prediction task requires keeping the long-term dependence of the input sequence and using the past load sampled data to infer the state of the current load and predict future trends. So the structure of BiLSTM determines that it can combine the information in the forward and backward directions to remember the long-term information to realize the life-learning of the load prediction. The structure of BiLSTM is demonstrated in Fig. 6.

Giving an input sequence of load features, $X^T_{(l,r)} = \{x^1_{(l,r)}, x^2_{(l,r)}, \dots, x^t_{(l,r)}\}$, and $t \in T$ represents the time steps. We feed each of these sequences $X^T_{(l,r)}$ into the LSTM units. BiLSTM introduces a second layer network to extend the LSTM structure to avoid ignoring the future context information, which includes the input layer, output layer, backward layer, and forward layer. BiLSTM structure uses two sub-nets to deal with the left and right load feature sequence context, which are transmitted forward and backward. Each LSTM unit includes the forget gate, the input gate, and the output gate [66]. At the time t, for the output of the forward LSTM network layer, there are two parts of information: the information of the current and the previous moments in the input data sequence. At the time t, for the backward LSTM network layer, there are two parts of information: the current and the next moments in the input data sequence. The mathematical expression defines each part. The forget gate calculates the forgetting probability of the state in the previous layer, and the formula is described as follows:

$$f^t_{(l,r)} = \sigma(W_{(l,r)\_fX} x^t_{(l,r)} + W_{(l,r)\_fh} h^{t-1}_{(l,r)} + W_{(l,r)\_fs} s^{t-1}_{(l,r)} + b_{(l,r)\_f}), \tag{10}$$

where, $f^t_{(l,r)}$ represents the forget gates in the forward and backward layer; $W_{(l,r)\_fX}$, $W_{(l,r)\_fh}$, and $W_{(l,r)\_fs}$ represent the weight value; $b_{(l,r)\_f}$ represents the bias. $h^{t-1}_{(l,r)}$ is the hidden layer state information of the previous network layer at time $t-1$, $x^t_{(l,r)}$ indicates the input data sequence at time t. We select the activation function sigmoid to obtain the output of the forget gate at time t, and $f^t_{(l,r)} \in [0, 1]$.

The input gate is used to input the current sequence position, and the formula describes it:

$$\begin{aligned} i^t_{(l,r)} = \sigma(&W_{(l,r)\_iX} x^t_{(l,r)} + W_{(l,r)\_ih} h^{t-1}_{(l,r)} \\ &+ W_{(l,r)\_is} s^{t-1}_{(l,r)} + b_{(l,r)\_i}), \end{aligned} \tag{11}$$

$$\begin{aligned} s^{*t}_{(l,r)} = tanh(&W_{(l,r)\_sX} x^t_{(l,r)} + W_{(l,r)\_hs} h^{t-1}_{(l,r)} \\ &+ W_{(l,r)\_ss} s^{t-1}_{(l,r)} + b_{(l,r)\_s}), \end{aligned} \tag{12}$$

where, $i^t_{(l,r)}$ indicates the input information of the input gate at time t, which represents the probability for adding the new information; $s^{*t}_{(l,r)}$ represents the new information. After updating the cell, the output gate controls the amount of information passed to the next state $h^t_{(l,r)}$. The formula is described as

$$\begin{aligned} o^t_{(l,r)} = \sigma(&W_{(l,r)\_oX} x^t_{(l,r)} \\ &+ W_{(l,r)\_oh} h^{t-1}_{(l,r)} + W_{(l,r)\_os} s^{t-1}_{(l,r)} + b_{(l,r)\_o}). \end{aligned} \tag{13}$$

We use additional information S and the amount of forgetting from the forget gates and input gates to update memory units $s^t_{(l,r)}$:

$$s^t_{(l,r)} = f^t_{(l,r)} * s^{t-1}_{(l,r)} + i^t_{(l,r)} * s^{*t}_{(l,r)}. \tag{14}$$

At the time t, $h^t_{(l,r)}$ is the output state of the hidden networks layer, and the expression is described as $h^t_{(l,r)} = o^t_{(l,r)} * tanh(s^t_{(l,r)})$.

We define the output of the hidden layer BiLSTM as $h^t_{l,r} = h^t_l \oplus h^t_r$, where, h^t_l and h^t_r indicate the backward layers and the forward layers, respectively. We use the weight coefficient w^1, w^2, w^3, w^4, w^5 and w^6 to concatenate the output of BiLSTM, and the expression is as:

$$h^t_r = f_r(w^1 x^t + w^2 h^{t-1}_r), \tag{15}$$

$$h^t_l = f_l(w^3 x^t + w^5 h^{t+1}_l), \tag{16}$$

$$O^t = g_{con}(w^4 h^t_r + w^6 h^t_l). \tag{17}$$

Loss function In our model, the definition of the loss function includes Euclidean of the similarity vectors between nodes (ESV), root mean square error (RMSE), and the cross-entropy loss function, and the loss function is described as follows:

$$Loss = Loss\_E + Loss\_R + Loss\_C. \tag{18}$$

Loss_E We use the density-based clustering algorithm to form a graph structure of the load frequency-domain image. And then, a propinquity-based node density judgment mechanism is employed to divide into N propinquity-based subgraphs, which will be the input embedding into the GNN for the load representative learning. We utilize the least Euclidean distance criteria to adjust the embeddings and optimize the model parameter. The similarity matrix in inter-nodes J(G) and the Euclidean distance of the inter-nodes is as follows:

$$J(j^t_{v_a}, j^t_{v_b}) = \sqrt{\sum\nolimits_k^N [Jac\_S(v_a, j^t_{v_k}) - Jac\_S(v_b, j^t_{v_k})]}, \tag{19}$$

where Jac_S represents the correlation coefficient of inter-node, and j_{v_a}, j_{v_b} is the corresponding vectors of the node v_a, v_b in the similarity matrix. According to the formula, the small distance indicates a high similarity inter-nodes and the more likely they belong to the same cluster. The nodes are

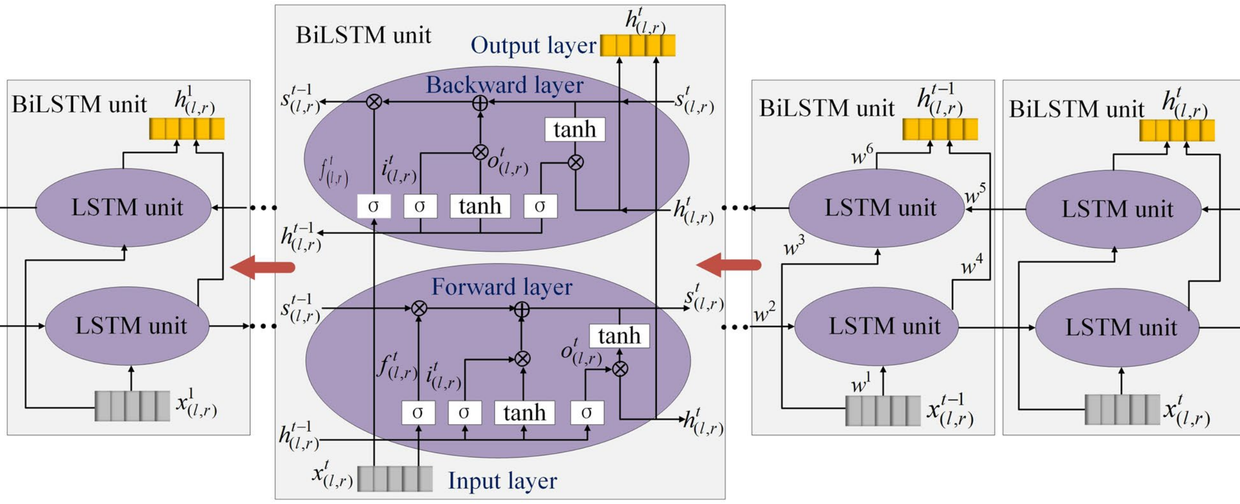


Fig. 6 Internal design structure of BiLSTM

divided into the K clusters where the cluster center with the smallest distance is located.

Loss_R The mathematical expression of RMSE is described as follows:

$$R_{ti} = \sqrt{\frac{1}{M_{ti}} \sum_{j=1}^{M_{ti}} (\hat{y}_{ti}(j) - y_{ti}(j))^2}, \tag{20}$$

where $\hat{y}_{ti}(j)$ represents the predicted value and $y_{ti}(j)$ is the real value predicted value. The gradient descent is used in the model parameter updating. To obtain the mean values of all training samples, the expression of the average Cross-Entropy Loss is as the following:

$$L\_c = -\frac{1}{M} \sum_{ti}^{M} [y_{(ti)} \log y^*_{(ti)} + (1 - y_{(ti)}) \log(1 - y^*_{(ti)})]. \tag{21}$$

4 Experiments settings

The chapter mainly introduces the model evaluation methods, the preparation of the experimental data, the specific experiment, and the discussion of the experimental results.

4.1 Model evaluation index

In this paper, to evaluate the performance of our proposed model, we use the most commonly used evaluation index, such as root-mean-square error (RMSE), the receiver operating characteristic curve (ROC), and the mean prediction accuracy (MAP).

RMSE RMSE is used to evaluate the deviation between the predicted value of the model $f^t(x_i)$ and the measured value y^t_{ci}, and the mathematical expression is as follows:

$$RMSE^t_c(X, f) = \sqrt{\frac{1}{N_c} \sum_{i=1}^{N_c} (f^t(x_i) - y^t_{ci})^2}. \tag{22}$$

ROC According to a series of different dichotomies, namely the cut-off value or determination threshold, ROC curves are described as a sensitivity curve to represent the prediction accuracy of X for Y.

MAP MAP is described as the abscissa representing the false positive rate, and the ordinate represents the true positive rate. MAP is the average of the accuracy at different recall rates, and it is a measure of how good or bad the learned model is in all categories, as shown in the following formula:

$$MAP = \frac{\sum_{i=1}^{N} AP_i}{N}. \tag{23}$$

AP_i indicates the average precision, which averages the accuracy at points with different recall rates. The larger value of AP, the higher the average accuracy of the model. AP is described as follows:

$$\int_0^1 Q(\lambda) d(\lambda), \tag{24}$$

where Q is used as the precision, λ indicates the recall, and we employ the Q as the function that takes λ. AP is the area under the curve.

4.2 Data set preparation

In our paper, the experimental data sets are mainly the sampled short-term power load data from the state grid Shijiazhuang power supply company. The data sets are collected by the smart electricity meter. The sampling period is from March 1, 2013, to December 31, 2017, and the total amount of electricity consumption of each user every day. The composition of the experimental data sets includes two parts: the electricity data recorded of the city, such as user ID, the date of the electricity for daily use, the total amount of electricity, and so on; during that period time, the meteorological factors data of this region, such as daily mean temperature, holiday records and meteorology records, and so on. We prepared two types of experimental data, the sampled data sets with considering the external influence factors; the sampled data sets without considering the external influence factors. Specifically, in the first class, we select the load sampled data from 2013 to 2015 as the training data sets RData1, RData2, and RData3; the load data from 2015 to 2016 as the testing data set RData4, and RData5, the load data from 2017 the validation data set RData6 and RData7. In the second-class, load data sets without considering the external influence factors are named LRData1, LRData2, and LRData3, from 2013 to 2014.

4.3 Experimental platform setting

The experimental results are obtained from a desktop computer: Intel i7 3.4 GHz CPU, 64 G DDR5 RAM memory, and 8*NVIDIA GeForce GTX 3080 Ti GPU. To better the implementation of our proposed method, Python 3.5.2 is employed.

4.4 Model training parameter

To effectively train our proposed model, we present the parameters of our model setting in detail: the number of blocks is set to 6, each block has 32 filters, the size of the filter is 5, the scale of the batch is 1024, the learning rate is 0.0009, *ReLU* is used as the activation function, and optimizer is Adam; the number of the hidden network layers is set 64; the number of GCN is 2, and the load embedding dimension is 120.

4.5 Baseline models

We employ five baseline models for load prediction to verify the performance of our method. The baseline models are introduced as follows:

Autoregressive Integrated Moving Average (ARIMA) [67]: this algorithm mainly utilizes the differential operation by the AR model and MA model to realize the stationary time series transformations of the non-stationary time series. The model can reduce the prediction error rate.

Support vector machine (SVM) [68]: it is a load forecasting method based on statistical theory. This method is a linear classifier that can define the maximum spacing in the feature space and constructs the optimal linear regression function to achieve the optimal prediction. The primary implementation uses a nonlinear transformation technology to make the high-dimensional vectors of the original sampled into the high-dimensional feature vectors.

Multi-scale skip deep long short-term memory (MSD-LSTM) [69]: first, the author uses the MSD method to realize the load feature extraction at different time scales and then utilizes the multilayer LSTM algorithm to solve the time dependence of features in each scale skip for the load prediction. In our paper, we structure the LSTM framework for the lifelong learning of the prediction model.

Generative adversarial network (GAN) [70]: the influence factors of the load are concerned with model training. In this method, a game training of GAN is generated by the influence factors of load, and the load prediction model depends on the generated training data set with the influencing factors. In our paper, we reconstruct the dependencies between the external factors of load and the load data with the graph structure.

Graph convolutional network (GCN) [71]: GCN is applied to solve nonlinear problems of load multi-relation sampling data and then introduces the spatio-temporal multigraph convolutional layer and self-attention mechanism to extract the load features. But this method does not have the ability to self-learning for load forecasting tasks.

4.6 Specific experimental design

The specific experimental designs: firstly, we use the Box-Cox transformation algorithm to achieve the data preprocessing for improving the validity of the experimental data, which provides a way to weak the strong random fluctuation of the load sampled data; we show the comparison results of the data preprocessing by the average accuracy of the model training. Secondly, we use the preprocessed data sets to train our model and five baseline models, respectively, and then the RMSE and average training accuracy are employed to evaluate the performance of various models. Thirdly, to further verify the performance of various models, we discuss the RMSE, ROC, and MAP in the testing and validation data sets, respectively.

4.7 Experimental results and discussion

Question 1: the evaluation of the Box-Cox transformation algorithm on predicting accuracy.

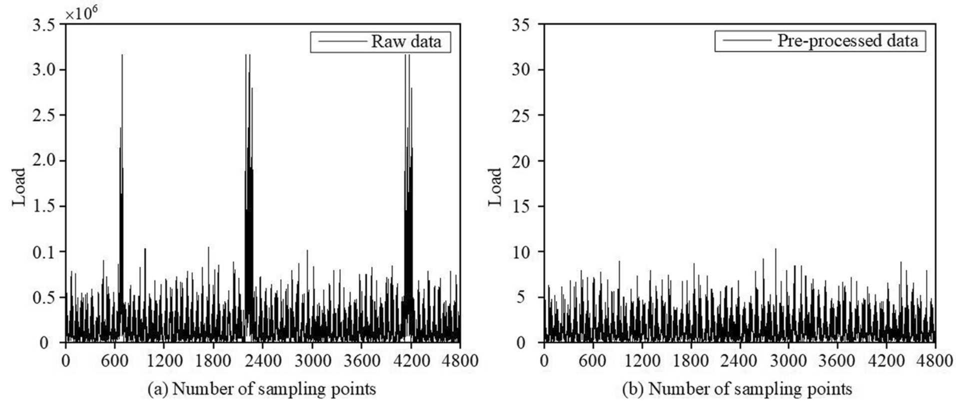


Fig. 7 Box-Cox transformation algorithm is used to preprocess the load sampled data

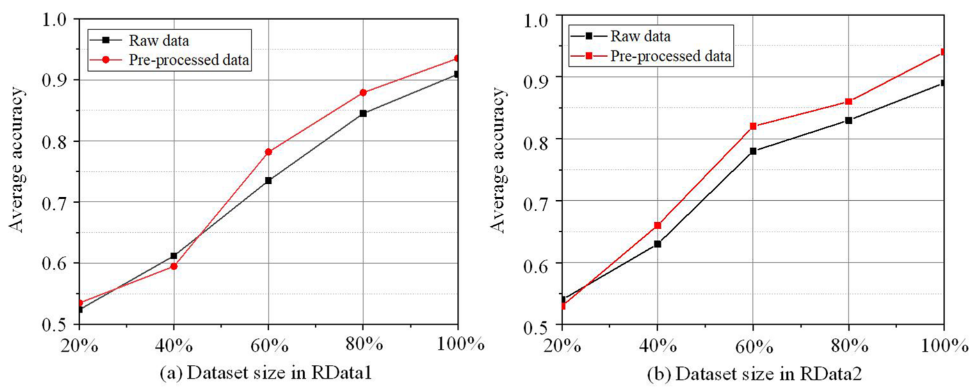


Fig. 8 Comparison of the model average accuracy on the raw data and pre-processed dataset

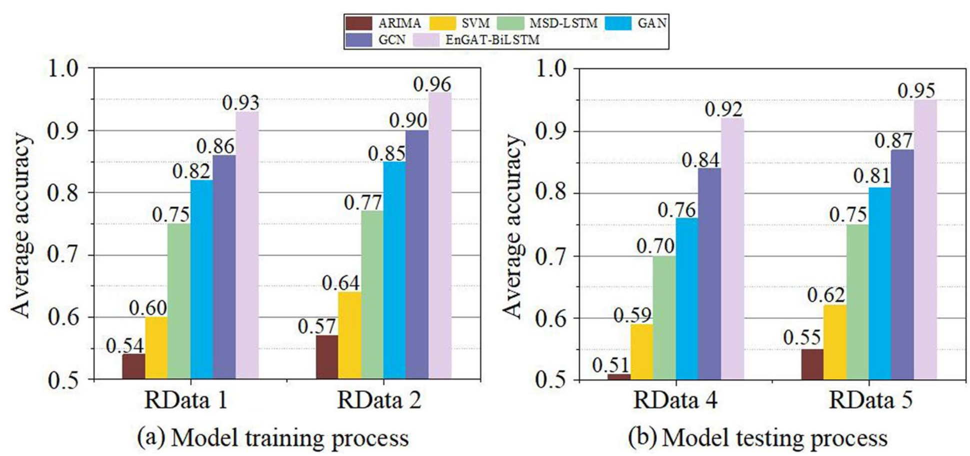


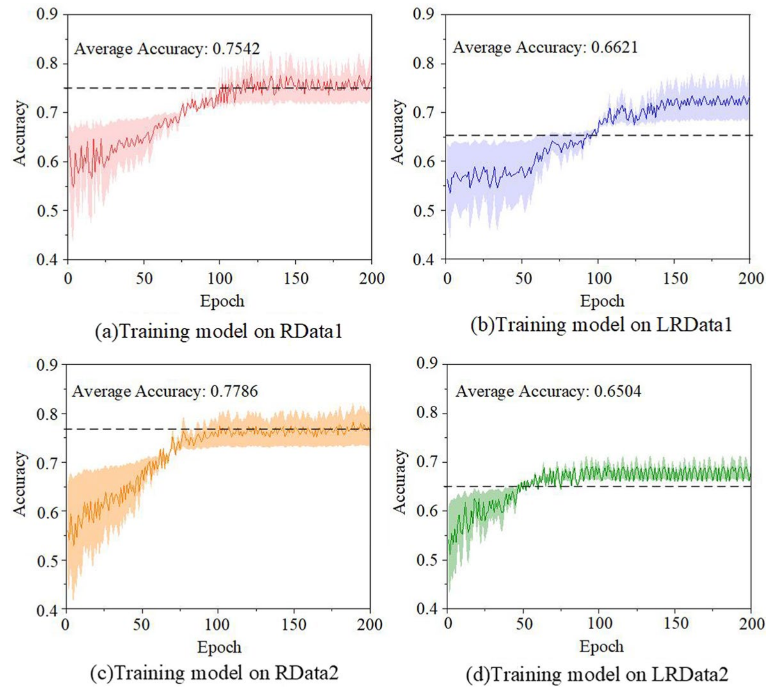
Fig. 9 The average accuracy of six methods in model training and testing process

Firstly, to improve the effectiveness of experimental data, we use the Box-Cox transformation algorithm to achieve the descending dimension and denoising of load raw sampled data. Data noise is significantly reduced after the data preprocessing, and the significant differences are shown in Fig. 7.

Secondly, we use the load sampled data sets RDatat1 and RDatat2 to train our proposed model and then utilize the average accuracy for verifying the effect of the preprocessing data sets on the model training accuracy. The experimental results are presented in Fig. 8. From Fig. 8a, on the training data set RData1, the average accuracy is not the advantage in the initial training stage, but as the data set size increases, the competitive advantage is gradually highlighted. Figure 8b demonstrates a similar result to RData1, and when the size of data sets is more significant than 25%, the average accuracy can improve by about 5%. These experimental results show that the preprocessing data sets for training our model can improve the model training accuracy.

Question 2: the evaluation of the performance of our model.

Fig. 10 The training accuracy obtained by our method in the model training process. **a** and **c** show the training accuracy in the external factors considered; **b** and **d** show the results in no external factors



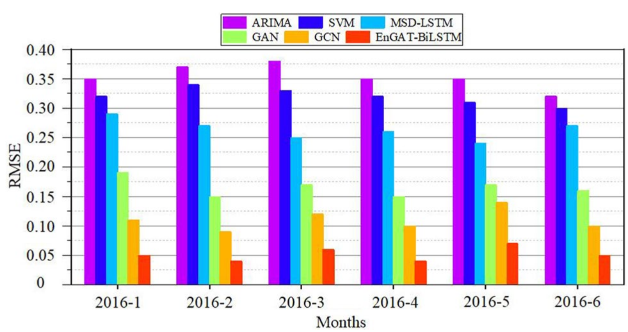


Fig. 11 The evaluation indicators RMSE of six methods are shown on the sampled dataset from 2016

We employ four datasets, RData1, RData2, RData4, and RData5, to evaluate the performance of our model and five baseline models, respectively, and then to achieve a comparative analysis of the experimental results. We implement the cross-validated data sets eight times to reduce the error of the experimental operations. The average accuracy on training sets and testing sets are shown, respectively, in Fig. 9. From Fig. 9a, in the training process, the average accuracy of EnGAT-BiLSTM on RData1 and RData2 is 0.93 and 0.96, respectively, and the accuracy is higher than the second-rank model GCN 0.07 and 0.06; the GAN model is 0.82 and 0.85, and the SVM model is 0.60 and 0.64; the accuracy of the ARIMA model is the lowest. From Fig. 9b in the testing process, the average accuracy of the EnGAT-BiLSTM model is 0.92 and 0.95, respectively; the GCN is 0.84 and 0.87, and the third-rank GAN model is 0.76 and 0.75. These results implicate that the training and testing accuracy of the EnGAT-BiLSTM method is relatively high, and the single LSTM model has no advantages for load prediction; the traditional time series forecasting method ARIMA has its limitations and is not very applicable in a complex scenarios.

To fully present the model performance for load forecasting, we use the load sampled data set with the various external influencing factors (RData1 and RData2) and the data set without the external factors (LData1 and LData2) to train our proposed model, respectively. From Fig. 10, the average accuracy is 0.7542 in RData1, and the average accuracy is 0.7786 in RData2, which is higher than the data set without the external factors 0.0921 and 0.1282, respectively. The results illustrate considering the external factors for load forecasting will improve prediction accuracy.

We use RMSE as the criteria to evaluate our proposed model and five benchmark models. Specifically, we select two load datasets from the six consecutive months (2016–1 to 2016–6, 2017–3 to 2017–8) in the RData1 employed for load forecasting. The experimental results are presented in Figs. 11 and 12. From Fig. 11, the RMSE value of EnGAT-BiLSTM is smaller than other models every month and has a slight fluctuation. The average RMSE value is 0.03 and shows that EnGAT-BiLSTM has stable predictive performance in prediction. In Fig. 12, a result similar to that

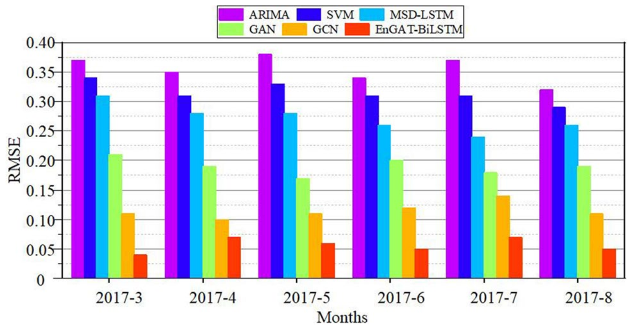


Fig. 12 The evaluation indicators RMSE of six methods are shown on the sampled dataset from 2017

presented in Fig. 11, the average RMSE value of EnGAT-BiLSTM is 0.05 and is lower than other baseline methods. The traditional methods based on time series and single neural network models are not better at load prediction.

This is because the time series model ARIMA only uses the raw data for load prediction, which cannot capture the complete load time series feature with nonlinear relationships. The single neural network models, such as SVM, MSD-LSTM, and GAN, without modeling the internal properties and external influences of load, have distinct disadvantages in realizing load forecasting. The GCN model does not consider time dependence, and the prediction is poor. EnGAT-BiLSTM is based on load heterogeneous information modeling, and fusion load dynamic attribute mapping knowledge domain DLKG, which benefits the representation learning of abstract load feature attributes and correlation to improve prediction accuracy.

We use RMSE, F1-score, and MAP criteria to achieve the comparative performance testing on datasets RData3, RData4, RData5, and RData6. The results are shown in Table 1, and the best results are marked in bold.

In Table 1, we verify the performance of the six methods on four datasets. The evaluation metrics RMSE, F1-score, and MAP for various models are obtained, and the EnGAT-BiLSTM model has achieved a remarkable effect. The RMSE of the EnGAT-BiLSTM model is 0.0523, 0.0423, 0.0519, and 0.0451, respectively, and the values are lower than the GCN models 0.0998, 0.1201, 0.0897, and 0.0841. The F1-score of the EnGAT-BiLSTM model is 0.9368, 0.9247, 0.9412, and 0.9287, respectively, which presents a good prediction property. The scores show that the graph structure model has the best performance, and the single time series model and the neural network model do not have an evident effect on dealing with the time-dependent data. Introducing external factors for load forecasting leads to a more complex data structure. At the beginning of model training, the GCN model presents good trends, but the model convergence needs to improve as the sample size increases. The EnGAT-BiLSTM model has excellent effects for the representation learning of the data based on graph structure. The BiLSTM framework can better realize load forecasting with complex dependency. The MAP value of the EnGAT-BiLSTM model is 0.9558, 0.9421, 0.9308, and 0.9452, which further validates the stable predictive performance of our model.

Table 1 Comparison of the performance indicators in five methods

Datasets	Method	$RMSE$	$F1-score$	MAP
RData3	ARIMA	0.3752	0.4918	0.6548
	SVM	0.3223	0.6345	0.7348
	MSD-LSTM	0.2318	0.7364	0.7926
	GAN	0.1916	0.8147	0.8435
	GCN	0.1511	0.8513	0.8913
	EnGAT-BiLSTM	**0.0523**	**0.9368**	**0.9558**
RData4	ARIMA	0.3931	0.5195	0.6157
	SVM	0.2923	0.7077	0.7656
	MSD-LSTM	0.2516	0.8156	0.8243
	GAN	0.1932	0.8578	0.8665
	GCN	0.1624	0.8745	0.9014
	EnGAT-BiLSTM	**0.0423**	**0.9247**	**0.9421**
RData5	ARIMA	0.3165	0.5745	0.6589
	SVM	0.2952	0.7515	0.7425
	MSD-LSTM	0.2746	0.8086	0.7965
	GAN	0.1803	0.8548	0.8783
	GCN	0.1416	0.8814	0.9045
	EnGAT-BiLSTM	**0.0519**	**0.9412**	**0.9308**
RData6	ARIMA	0.3724	0.5397	0.6321
	SVM	0.2456	0.7704	0.7672
	MSD-LSTM	0.2109	0.7934	0.8545
	GAN	0.1665	0.8428	0.8632
	GCN	0.1292	0.8781	0.8976
	EnGAT-BiLSTM	**0.0451**	**0.9287**	**0.9452**

AUC area is one of the most important indexes to evaluate the model performance. Generally, we by the closing degree of the ROC curve to the upper left corner to judge the accuracy of the model. The maximum AUC area is 1, AUC area of each model is shown in Fig. 13. From Fig. 13, the AUC area of the EnGAT-BiLSTM model is 0.9468, and this value is closest to the top left corner, which is 0.0796 higher than the second-rank GCN. Introducing the attention module into the GNN structure can extract the load feature effectively. The AUC area of the SVM and ARIMA models is 0.5578 and 0.4512, respectively, which proves their model performance is inferior to that of the graph structure model for realizing the load prediction.

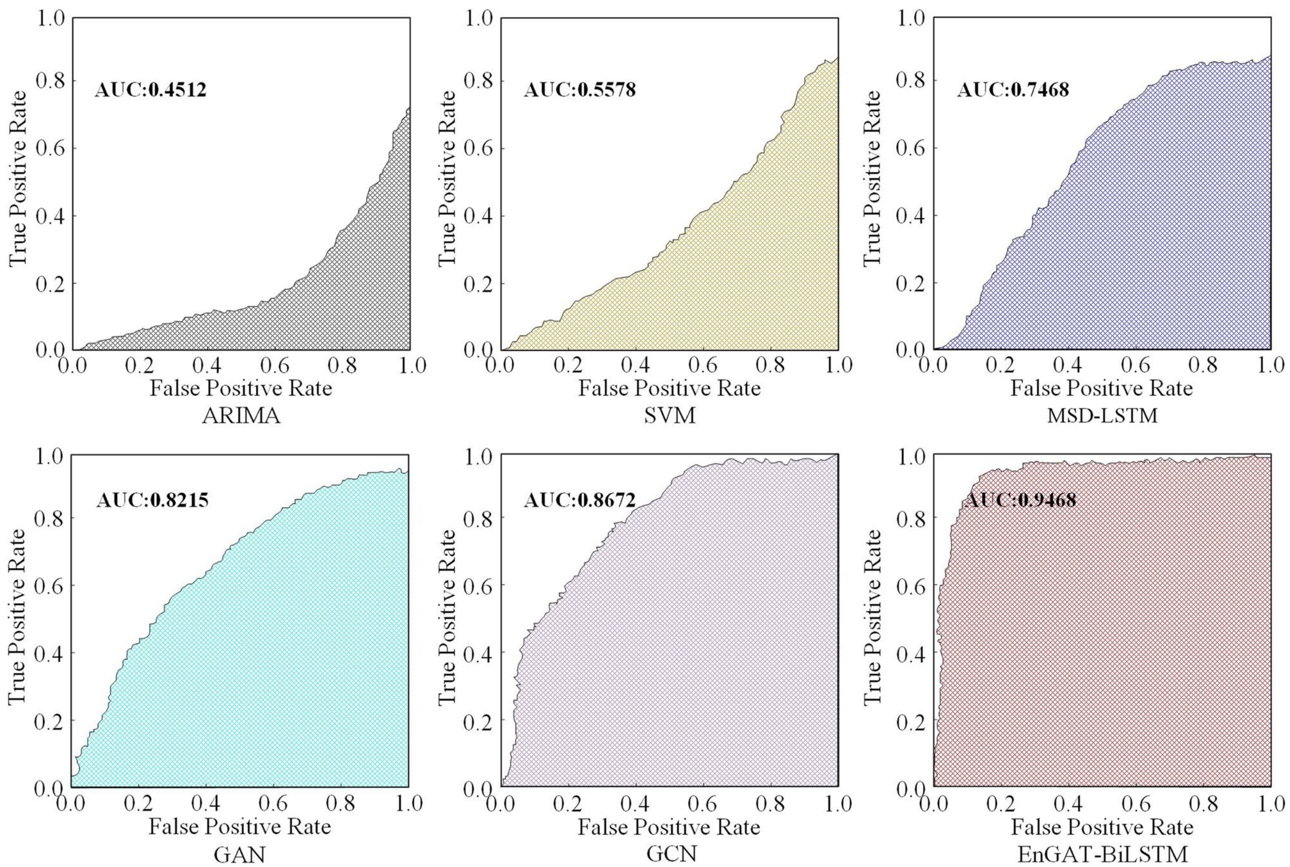


Fig. 13 Compare the results of AUC area in six methods

5 Conclusion

In this paper, we present a novel load forecasting method, EnGAT-BiLSTM, to improve the accuracy and the instability of short-term load forecasting. In this method, utilizing Box-Cox transformation technology to achieve the preprocessing of raw load dataset and to improve the effectiveness of the experimental data; EnGAT is used to enhance the learning ability of the load features representational; the BiLSTM framework is utilized to implement the load lifelong learning and forecasting. Lots of experiments have shown that our method has significant performance improvements for load forecasting. This method can be spread to the IoT system of the smart grid. The next stage of our work will use the more complex data sets training model and extend it to applications in other related fields.

The authors of this paper were supported by S &T Program of Hebei through Grant 20310101D. The authors of this paper were also supported by Humanities and Social Science research project of Hebei Province SQ2022071.

References

1. Tang FX, Fadlullah ZM, Mao BM, Kato N (2019) An Intelligent Traffic Load Prediction-Based Adaptive Channel Assignment Algorithm in SDN-IoT: A Deep Learning Approach. IEEE Internet Things J 5(6):5141–5154. https://doi.org/10.1109/JIOT.2018.2838574
2. Rabie AH, Saleh AI, Ali HA (2021) Smart electrical grids based on cloud. IoT, and big data technologies: state of the art, ournal of ambient intelligence and humanized computing. 12(10):9449–9480. https://doi.org/10.1007/s12652-020-02685-6
3. Maity I, Misra S, Mandal C (2021) CORE: Prediction-Based Control Plane Load Reduction in Software-Defined IoT Networks. IEEE Trans Commun 69(3):1835–1844. https://doi.org/10.1109/TCOMM.2020.3043760
4. Bedi G, Venayagamoorthy GK, Singh R (2020) Development of an IoT-Driven Building Environment for Prediction of Electric Energy Consumption. IEEE Internet Things J 7(6):4912–4921. https://doi.org/10.1109/JIOT.2020.2975847
5. Randall L, Agrawal P, Mohapatra A (2022) IoT Based Load Forecasting for Reliable Integration of Renewable Energy Sources. JOURNAL OF SIGNAL PROCESSING SYSTEMS FOR SIGNAL IMAGE AND VIDEO TECHNOLOGY. https://doi.org/10.1007/s11265-022-01785-0
6. Bui V, Le NT, Nguyen VH, Kim J, Jang YM (2022) Multi-Behavior with Bottleneck Features LSTM for Load Forecasting

in Building Energy Management System. Electronics 10(9):209–236. https://doi.org/10.3390/electronics10091026

7. Zhang CG, Chen ZC, Wu LJ, Cheng, SY, Lin PJ (2019) A NB-IoT based intelligent combiner box for PV arrays integrated with short-term power prediction using extreme learning machine and similar days, IOP Conference Series-Earth and Environmental Science. Paper presented at 4th International Conference on Energy Engineering and Environmental Protection (EEEP), 2021, Xiamen, PEOPLES R CHINA. 467. (2019). https://doi.org/10.1088/1755-1315/467/1/012081
8. Zhang XY, Cordoba-Pachon JR, Guo PQ, Watkins C, Kuenzel S (2022) Privacy-Preserving Federated Learning for Value-Added Service Model in Advanced Metering Infrastructure. IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS. https://doi.org/10.1109/TCSS.2022.3204361
9. Liang F, Yu, A, Hatcher WG, Yu W, Lu C (2019) Deep Learning-Based Power Usage Forecast Modeling and Evaluation, Procedia Computer Science. Paper presented at 9th International Conference of Information and Communication Technology [ICICT], Nanning, PEOPLES R CHINA. https://doi.org/10.1016/j.procs.2019.06.016
10. Xiao LY, Shao W, Yu MX, Ma J, Jin CJ (2017) Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting. Appl Energy 198(9):203–222. https://doi.org/10.1016/j.apenergy.2017.04.039
11. Chandrarathna Kasun, Edalati Arman, Tabar Ahmad Reza Fourozan (2020) Forecasting Short-term load using Econometrics time series model with T-student Distribution. Preprint at https://arxiv.org/Statistical Finance
12. Ozturk Ali, Tosun Salih, Celik Hasan (2016) Forecasting Short-term load using Econometrics time series model with T-student Distribution. International Symposium Innovative Technologies Engineering and Science
13. Wu F, Cattani C, Song W, Zio E (2020) Fractional ARIMA with an improved cuckoo search optimization for the efficient Short-term power load forecasting. Academie PressAEJ - Alexandria Engineering Journal. 59(5). https://doi.org/10.1016/j.aej.2020.06.049
14. Wang Huiping, Wang Yi (2022) Forecasting solar energy consumption using a fractional discrete grey model with time power term. Clean Technologies and Environmental Policy. Preprint at https://doi.org/10.1007/s10098-022-02320-2
15. Zhu Fuyun, Wu Guoqing (2021) Load Forecasting of the Power System: An Investigation Based on the Method of Random Forest Regression. Energy Engineering. Preprint at https://doi.org/10.32604/EE.2021.015602
16. Aasim Singh SN, Mohapatra A (2021) Data driven day-ahead electrical load forecasting through repeated wavelet transform assisted SVM model. Applied Soft Computing. 111(16):107730. https://doi.org/10.1016/j.asoc.2021.107730
17. Lv Da, Xue Jianjie, Zhang Yan, Tang Mengcong, Chen Qian (2020) Research on Power User Intelligent Load Forecasting Method Based on Data State Drive. Journal of Physics Conference Series. 1449:012073. https://doi.org/10.1088/1742-6596/1449/1/012073
18. Ma H, Tang JM (2020) Short-Term Load Forecasting of Microgrid Based on Chaotic Particle Swarm Optimization. Procedia Computer Science. 166:546–550. https://doi.org/10.1016/j.procs.2020.02.026
19. Fan GF, Yu M, Dong SQ, Yeh YH, Hong WC (2021) Forecasting short-term electricity load using hybrid support vector regression with grey catastrophe and random forest modeling. Utilities Policy. 73(2):173–196. https://doi.org/10.1016/j.jup.2021.101294
20. Hafeez G, Khan I, Usman M, Aurangzeb K, Ullah A (2020) Fast and Accurate Hybrid Electric Load Forecasting with Novel Feature Engineering and Optimization Framework in Smart Grid. Paper presented at the 6th Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia. https://doi.org/10.1016/j.jup.2021.101294
21. Carcangiu S, Fanni A, Pegoraro PA, Sias G, Sulis S (2020) Forecasting-Aided Monitoring for the Distribution System State Estimation. Complexity 2020(4281219):4281219. https://doi.org/10.1155/2020/4281219
22. Peng H, Li J, Song Y, Yang R, He L (2021) Streaming Social Event Detection and Evolution Discovery in Heterogeneous Information Networks, ACM TRANSACTIONS ON KNOWLEDGE DISCOVERY FROM DATA. 15. https://doi.org/10.1145/3447585
23. Peng Hao, Li Jianxin, Gong Qiran, Ning Yuanxin, He Lifang (2020) Motif-Matching Based Subgraph-Level Attentional Convolutional Network for Graph Classification. Paper presented at 34th AAAI Conference on Artificial Intelligence / 32nd Innovative Applications of Artificial Intelligence Conference / 10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY. 34:5387-5394
24. Silva Mad, Abreu T, CR Santos-Júnior, Minussi CR (2021) Load forecasting for smart grid based on continuous-learning neural network. Electric Power Systems Research. 201(4281219):107545-107545. https://doi.org/10.1016/j.epsr.2021.107545
25. Soares LD, Franco, Emc (2021) BiGRU-CNN neural network applied to short-term electric load forecasting. Electric Power Systems Research. 32. https://doi.org/10.1590/0103-6513.20210087
26. Miao Kai, Hua Qiang, Shi Huifeng (2021) Short-Term Load Forecasting Based on CNN-BiLSTM with Bayesian Optimization and Attention Mechanism. CONCURRENCY AND COMPUTATION-PRACTICE & EXPERIENCE 32. https://doi.org/10.1002/cpe.6676
27. Zhao H, Yao Q, Weiwei TU (2021) Search to aggregate neighborhood for graph neural network, IEEE International Conference on Data Engineering. Paper presented at 2021 IEEE 37TH INTERNATIONAL CONFERENCE ON DATA ENGINEERING (ICDE 2021), ELECTR NETWORKc. https://doi.org/10.1109/ICDE51399.2021.00054
28. Dudek G, Peka P, Smyl S (2020) A Hybrid Residual Dilated LSTM end Exponential Smoothing Model for Mid-Term Electric Load Forecasting. Preprint at https://arxiv.org/2004.00508
29. Xiao Y, Zheng KH, Zheng ZJ, Qian B, Li S, Ma QL (2021) Multi-scale skip deep long short-term memory network for short-term multivariate load forecasting. Journal of Computer Applications. 41(1):231–236
30. Fu L (2020) Time Series-oriented Load Prediction Using Deep Peephole LSTM. Paper presented at 12th International Conference on Advanced Computational Intelligence (ICACI), Dali, PEOPLES R CHINA
31. Xu XZ, Meng ZR (2020) A hybrid transfer learning model for short-term electric load forecasting. Electr Eng 10(3):1371–1381. https://doi.org/10.1007/s00202-020-00930-x
32. Lu YK, Tian Z, Zhou RY, Liu WJ (2020) Multi-step-ahead prediction of thermal load in regional energy system using deep learning method. ENERGY AND BUILDINGS. 233:110658. https://doi.org/10.1016/j.enbuild.2020.110658
33. Wang B, Zhang LM, Ma HR, Wang HX, Wan SH (2019) Parallel LSTM-Based Regional Integrated Energy System Multienergy Source-Load Information Interactive Energy Prediction. Complexity 2019:7414318. https://doi.org/10.1155/2019/7414318
34. Wang B, Zhang LM, Ma HR, Wang HX, Wan SH (2020) Smart load prediction analysis for distributed power network of Holiday Cabins in Norwegian rural area. J Clean Prod 266:121423. https://doi.org/10.1016/j.jclepro.2020.121423

35. Wang B, Zhang LM, Ma HR, Wang HX, Wan SH (2020) Edge sensing data-imaging conversion scheme of load forecasting in smart grid. SUSTAINABLE CITIES AND SOCIETY. 62:102363. https://doi.org/10.1016/j.scs.2020.102363
36. Yong Zhao, Jiangxuehui Zhao (2020) Residual resistance prediction of sailing yacht based on random forest model, Journal of Huazhong University of Science and Technology. Nature Science. 10(1):6332. https://doi.org/10.1038/s41598-020-63331-x
37. Fang L, Zhou ZY, Hong YP (2021) Symmetry Analysis of the Uncertain Alternative Box-Cox Regression Model. SYMMETRY-BASEL. 49(11):118–122
38. Zhao Z, Zhao YG, Li PP (2022) Efficient approach for dynamic reliability analysis based on uniform design method and Box-Cox transformation. MECHANICAL SYSTEMS AND SIGNAL PROCESSING. 172:108967. https://doi.org/10.1016/j.ymssp.2022.108967
39. Zhu KD, Geng J, Wang K (2021) A hybrid prediction model based on pattern sequence-based matching method and extreme gradient boosting for holiday load forecasting. ELECTRIC POWER SYSTEMS RESEARCH. 190(106841). https://doi.org/10.1016/j.epsr.2020.106841
40. Ge L, Li Y, Yan J, Wang Y, Zhang N (2021) Short-term Load Prediction of Integrated Energy System with Wavelet Neural Network Model Based on Improved Particle Swarm Optimization and Chaos Optimization Algorithm. JOURNAL OF MODERN POWER SYSTEMS AND CLEAN ENERGY. 9(6):1490-1499. https://doi.org/10.35833/MPCE.2020.000647
41. Chen Jinpeng, Hu Zhijian, Chen Weinan, Gao Mingxin, Du Yixing, Lin Mingrong (2021) Load Prediction of Integrated Energy System Based on Combination of Quadratic Modal Decomposition and Deep Bidirectional Long Short-term Memory and Multiple Linear Regression. Automation of Electric Power Systems. 45(1000-1026):85-94. https://doi.org/10.35833/MPCE.2020.000647
42. Guo Wei, Zhang Kai, Wei Xinjie, Liu Mei (2021) Short-Term Load Forecasting Method Based on Deep Reinforcement Learning for Smart Grid. Hindawi Limited. 2021(8453896). https://doi.org/10.1155/2021/8453896
43. Peng Hao, Zhang Ruitong, Li Shaoning, Cao Yuwei, Pan Shirui, Yu Philip (2022) Reinforced, Incremental and Cross-lingual Event Detection From Social Messages. IEEE transactions on pattern analysis and machine intelligence. PP. (2022). https://doi.org/10.1109/TPAMI.2022.3144993
44. Liu Chao, Li Xinchuan, Zhao Dongyang, Guo Shaolong, Yao Hong (2020) A-GNN: Anchors-Aware Graph Neural Networks for Node Embedding, Lecture Notes of the Institute for Computer Sciences Social Informatics and Telecommunications Engineering. Paper presented at 15th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine), Shenzhen, PEOPLES R CHINA. 300:141-153. https://doi.org/10.1007/978-3-030-38819-5_9
45. Dai Jianhua, Chen Yuanmeng, Xiao Lin, Jia Lei, He Yongjun (2022) Design and Analysis of a Hybrid GNN-ZNN Model with a Fuzzy Adaptive Factor for Matrix Inversion. IEEE Trans Industr Inf 2434–2442(4):2434–2442. https://doi.org/10.1109/TII.2021.3093115
46. Zhao Xusheng, Dai Qiong, Wu Jia, Peng Hao, Liu Mingsheng, Bai Xu, Tan Jianlong, Wang Senzhang, Yu Philip (2022) TinyGNN: Multi-view Tensor Graph Neural Networks Through Reinforced Aggregation, IEEE Transactions on Knowledge and Data Engineering
47. Zhang Z, Leng J, Ma L, Miao Y, Guo M (2020) Architectural Implications of Graph Neural Networks. IEEE Comput Archit Lett 19(1):59–62. https://doi.org/10.1109/LCA.2020.2988991
48. Peng H, Yang R, Wang Z, Li J, He L, Yu P, Zomaya A, Ranjan R (2022) LIME: Low-Cost Incremental Learning for Dynamic Heterogeneous Information Networks. IEEE Trans Comput 71(3):628–642. https://doi.org/10.1109/TC.2021.3057082
49. Bai Youhui, Li Cheng, Lin Zhiqi, Wu Yufei, Xu Yinlong (2020) Efficient Data Loader for Fast Sampling-based GNN Training on Large Graphs. IEEE Trans Parallel Distrib Syst 32(10):2541–2556. https://doi.org/10.1109/TPDS.2021.3065737
50. Guirado R, Jain A, Abadal S, Alarcón E (2019) Characterizing the Communication Requirements of GNN Accelerators: A Model-Based Approach, IEEE International Symposium on Circuits and Systems. Paper presented at2021 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS (ISCAS), Daegu, SOUTH KOREA. https://doi.org/10.1109/ISCAS51556.2021.9401612
51. Yin JB, Wang YY, Chen KY (2021) A Novel Graph Based Sequence Forecasting Model for Electric Load of Campus. Paper presented at 2nd International Conference on Artificial Intelligence and Information Systems (ICAIIS), Chongqing, PEOPLES R CHINA, 28-30 MAY. https://doi.org/10.1155/2021/8453896
52. Liao W, Bak-Jensen B, Pillai JR, Wang Y, Wang Y (2022) A Review of Graph Neural Networks and Their Applications in Power Systems. JOURNAL OF MODERN POWER SYSTEMS AND CLEAN ENERGY. 10:345-360. https://doi.org/10.35833/MPCE.2021.000058
53. Li HJ (2022) Short-Term Wind Power Prediction via Spatial Temporal Analysis and Deep Residual Networks. FRONTIERS IN ENERGY RESEARCH. 10(920407). https://doi.org/10.3389/fenrg.2022.920407
54. Peng H, Du B, Liu M, Liu M, He L (2021) Dynamic Graph Convolutional Network for Long-Term Traffic Flow Prediction with Reinforcement Learning. Inf Sci 578:401–416. https://doi.org/10.1016/j.ins.2021.07.007
55. Peng H, Zhang R, Dou Y, Yang R, Yu PS (2022) Reinforced Neighborhood Selection Guided Multi-Relational Graph Neural Networks. ACM TRANSACTIONS ON INFORMATION SYSTEMS. 40(4):69. https://doi.org/10.1016/10.1145/3490181
56. Zhao X, Wu J, Peng H, Beheshti A, Monaghan J, Mcalpine D, Hernandez-Perez H, Dras M, Dai Q, Li Y (2022) Deep Reinforcement Learning Guided Graph Neural Networks for Brain Network Analysis. Neural networks : the official journal of the International Neural Network Society. 154:56–67. https://doi.org/10.1016/j.neunet.2022.06.035
57. Zhang Y, Xiong R, He H, Pecht M (2018) Lithium-ion battery remaining useful life prediction with Box-Cox transformation and Monte Carlo simulation. IEEE Trans Industr Electron 66:1585–1597. https://doi.org/10.1109/TIE.2018.2808918
58. Johannesen NJ, Kolhe ML, Goodwin M (2020) Smart load prediction analysis for distributed power network of Holiday Cabins in Norwegian rural area-ScienceDirect. Journal of Cleaner Production. 266(121423). https://doi.org/10.1016/j.jclepro.2020.121423
59. Jiang YM, Liu MS, Peng H, Bhuiyan MZA (2021) A reliable deep learning-based algorithm design for IoT load identification in smart grid. Ad Hoc Networks. 123:102643. https://doi.org/10.1016/j.adhoc.2021.102643
60. Wang HX, Li YF, Li YF, Dang LM, Ko J, Han D, Moon H (2020) Smartphone-based bulky waste classification using convolutional neural networks, MULTIMEDIA TOOLS AND APPLICATIONS. 79(39-40):29411-29431. https://doi.org/29411-29431
61. Unlu R, Kiris R (2020) Detection of damaged buildings after an earthquake with convolutional neural networks in conjunction with image segmentation. VISUAL COMPUTER. 38(2):685–694. https://doi.org/10.1007/s00371-020-02043-9

62. Unlu R, Kiris R (2020) Neural Network Compression and Acceleration by Federated Pruning. Lecture Notes Artificial Intelligence 12453:173–183. https://doi.org/10.1007/978-3-030-60239-0_12
63. Shan Lin, Hong Wang, Linhai Qi, Hanyu Feng, Su Ying (2021) Short-term Load Forecasting Based on Conditional Generative Adversarial Network. Automation of Electric Power Systems. 45(1000–1026):52–60
64. Han Kai, Wang Yunhe, Guo Jianyun, Tang Yehui, Wu Enhua (2022) Vision GNN: An Image is Worth Graph of Nodes. arXiv preprint arXiv:2206.00272
65. Zhang Ruitong, Peng Hao, Dou Yingtong, Wu Jia, Sun Qingyun, Zhang Jingyi, Yu Philip S (2022) Automating DBSCAN via Deep Reinforcement Learning. In Proceedings of The 31th ACM International Conference on Information and Knowledge Management, CIKM 2022, Atlanta, Georgia, USA, 17-22 Oct
66. Peng Hao, Li Jianxin, Wang Zheng, Yang Renyu, Liu Mingsheng, Zhang Mingming, Yu Philip, He Lifang (2021) Lifelong Property Price Prediction: A Case Study for the Toronto Real Estate Market. IEEE Transactions on Knowledge and Data Engineering
67. Hai W, Wang, Y.: Load Forecast of Gas Region Based on ARIMA Algorithm, Chinese Control and Decision Conference. 1960–1965. (2020) Paper presented at 32nd Chinese Control And Decision Conference (CCDC). PEOPLES R CHINA, Hefei, p 2020
68. Dai XQ, Sheng KC, Shu FZ (2022) Ship power load forecasting based on PSO-SVM. MATHEMATICAL BIOSCIENCES AND ENGINEERING. 19:4547–4567. https://doi.org/10.3934/mbe.2022210
69. Yong Xiao, Kaihong Zheng, Zhenjing Zheng, Bin Qian, Sen Li, Qianli Ma (2021) Multi-scale skip deep long short-term memory network for short-term multivariate load forecasting. MATHEMATICAL BIOSCIENCES AND ENGINEERING. 41(1001–9081):231–236
70. Lin Shan, Wang Hong, Qi Linhai, Feng Hanyu, Su Ying (2021) Short-term Load Forecasting Based on Conditional Generative Adversarial Network. Automation of Electric Power Systems. 45(1000–1026):52–60
71. Liu RW, Liang MH, Nie JT, Yuan YL, Xiong ZH, Yu H, Guizani N (2022) STMGCN: Mobile Edge Computing-Empowered Vessel Trajectory Prediction Using Spatio-Temporal Multigraph Convolutional Network. Automation of Electric Power Systems. 18(11):7977-7987. https://doi.org/10.1109/TII.2022.3165886