Parallel Spatial-Temporal Graph Attention Network for Short Term Multi-Sequence Load Forecasting

Long Zhuo, Xu ZhiYuan, Wu Gongping*, *Member, IEEE,* Deng Feng, *Member, IEEE,* Chen Xiangyuan, Feng Wenshan, Huang Zhiwen

***Abstract*—Short-term load forecasting is a crucial task within the power system. However, existing studies have overlooked the spatial-temporal relationships between multiple series loads. Accounting for this spatial-temporal adjacency can lead to more accurate forecasting in certain scenarios. In this paper, we propose a short-term load forecasting model named Parallel Spatial-Temporal Graph Attention Network (PST-GAT). The method encodes the multi-sequence loads as nodes and constructs the adjacency matrix using the Dynamic Time Warping (DTW) technique to form a fully connected graph of the load data. Combining the sliding window concept, the constructed fully connected graph is partitioned into a series of subgraphs, and the Graph Attention Network (GAT) performs feature extraction on each subgraph individually. To realize learning at multiple scales, PST-GAT adopts a parallel multi-branching approach, where each branch splits the subgraphs with varying lengths. Finally, the feature vectors extracted by each branch are concatenated, and forecasting is accomplished using a fully connected layer. Moreover, the unique structural design of PST-GAT allows for the simultaneous prediction of multiple sequence loadings. Experimental results based on real-world load data validate the superior prediction accuracy of this method compared to existing algorithms.**

***Index Terms*—Short-term load forecasting, multi-sequence loads, spatial-temporal adjacency, graph attention network.**

I. Introduction

POWER system plays a vital role in modern society. Its main purpose is to provide the supply of sufficient and reliable power. The power system operation and planning rely heavily on electric load forecasting [2]. Accurate electric load forecasting can significantly improve the controlling and planning of the operation of electric grid systems [3]. Therefore, electric load forecasting is an important task for achieving the efficient operation of modern power grids [4].

Depending on the time horizons, load forecasting tasks can be divided into three main categories: short-term load forecasting (STLF), medium-term load forecasting (MTLF), and long-term load forecasting (LTLF). If the prediction is for a few hours or a day in the future, it is considered as STLF, and it is considered as LTLF if the prediction is for a year ahead. Among them, SLTF is of great research interest as it can be employed for the day-to-day operations of the utility companies, such as scheduling the generation, transmission of electric energy, and real-time energy dispatching [5]. The study of STLF has been extensively discussed in the existing literature, and significant progress has been made in this field.

Traditional methods for STLF include the autoregressive integrated moving average (ARIMA) model [6], support vector machine (SVM) [7], [8], Dynamic stochastic general equilibrium (DSGE) [9], fuzzy forecasting method [10], Monte Carlo simulation [11], random forest (RF) [12], [13], and various other statistical and machine learning approaches. With the rapid development and widespread application of artificial intelligence and deep learning, many researchers have focused on developing methods for STLF by combining artificial intelligence algorithms. Many algorithms, such as 1-dimensional convolutional network (1-D CNN) [14], residual network (ResNet) [15], Recurrent Neural Network (RNN) [16], Long Short-Term Memory (LSTM) [17], etc, have gained significant attention for their ability to capture complex temporal dependencies and patterns in time series data. Through combining with deep learning algorithms, load forecasting methods demonstrate strong and robust predictive capabilities. Dong et al. [18] propose an electric load forecasting model for systems containing high-level elastic loads, which consists of a 1-D CNN structure, a parallel forecasting structure, and a deep ResNet structure. The experimental results show the good performance and high generalization ability of this model.

Nevertheless, these approaches mainly focus on the temporal sequence of the data, overlooking the spatial relationships between different data sources. For instance, in residential load forecasting tasks, owing to factors such as geographic location and user behavior, load consumption patterns among households may exhibit similar trends. The correlation between households can be used to improve the accuracy of STLF [19]. Graph Neural Network (GNN), with its powerful ability to capture non-Euclidean pair-wise correlations, has garnered the attention of researchers in recent years. GNN is a machine learning model designed for processing graphs, a data structure comprising nodes and edges. Various new paradigms and definitions in graph domains have been rapidly updated to deal with the complicated graph-structured data in the past few years [20]. Models combining GNN have been successfully applied to wind speed forecasting [21], solar power prediction [22], [23] and transportation traffic forecasting [24]. In electric load forecasting, GNN has not been widely studied and applied. Existing studies typically model various objectives as nodes of a graph to express the spatial dependence of the data. Wu et al. [3] propose a self-adaptive Graph WaveNet based framework for residential load

Long Zhuo, Xu Zhiyuan, Wu Gongping, Deng Feng, Chen Xiangyuan, Huang Zhiwen and Feng Wenshan are with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410114, China (Corresponding Author: Wu Gongping, e-mail: longzhuo601@csust.edu.cn, 22205051051@stu.csust.edu.cn, gongping_wu@csust.edu.cn, df_csust@126.com cxyvergas@stu.csust.edu.cn 23205041069@stu.csust.edu.cn, and 23205071162@stu.csust.edu.cn)

forecasting. This framework utilizes historical load data from multiple households to construct graph data, fully exploiting the spatial relationships between different household load.

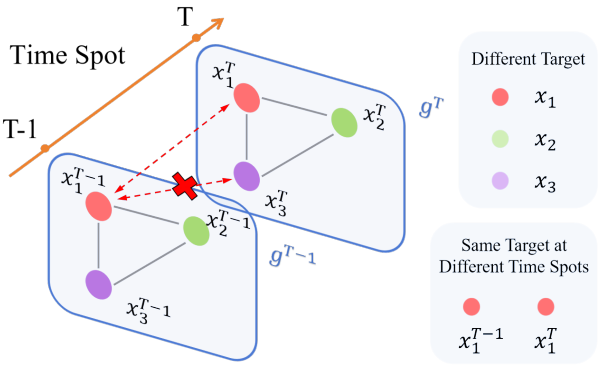

Fig. 1. Graphs at different time spots

However, there are still shortcomings in the existing researches. They process each graph independently, which will lose the temporal sequence between different series. These series can represent various household loads in forecasting tasks or sensors in different industrial load areas. As shown in Fig 1, existing approaches can learn the state evolution of the same series (eg., x_1^{T-1} and x_1^T) but not the temporal adjacencies between different series (e.g., x_1^{T-1} and x_3^T). In some cases, this temporal adjacency is crucial for accurate load forecasting. For instance, Large industrial load centers are typically equipped with multiple transformers serving different load areas. If one transformer fails or requires maintenance, the load distribution will change, leading to a sudden increase in the loads of other transformers. Without considering the temporal adjacency between transformers, these sudden load changes in other transformers are unexpected and unpredictable. To address this limitation and inspired by [26], this paper proposes utilizing fully-connected spatial-temporal (FCST) graph information to achieve multi-source STLF. Specifically, FCST graphs are constructed by using encoding networks and the dynamic time warping (DTW) technique on multiple-source load data to capture temporal correlation and spatial adjacency among them. Considering the complexity of the edge features, the constructed FCST graphs are combined with graph attention network (GAT) and other methods to learn features for effective STFL. The main contribution of this paper are as follow:

(1) A novel FCST graphs method is proposed to fuse the multi-source load data, enabling explicit and comprehensive modeling of temporal and spatial correlations between different data sources. This method captures the complex associations inherent in the data effectively.

(2) A FCST segmentation model is established by pooling and feature concatenation to capture local spatio-temporal relationships between multiple data sources efficiently.

(3) A novel STLF framework is introduced for multi-source load data. Compared to existing STLF methods, this framework facilitates synergistic predictions for multi-source data. The framework's validity and superiority are demonstrated by experiments with real-world data.

The rest of this paper organized as follows: Section 2 introduces technical background. The proposed methodology is detailed in Section 3. Section 4 gives the experimental results presentation and analysis. The paper is concluded in Section 5.

II. Technical Background

A. Dynamic Time Warping

Dynamic Time Warping (DTW) is applied in PST-GAT to construct edge information by calculating the similarity between the two load sequences. DTW is a technique used in the field of time series analysis to measure the similarity between two temporal sequences [27].

Assume that the two time series to be compared are $X = [x_1, x_2, .x_n]$ and $Y = [y_1, y_2, ...y_n]$. The key idea behind DTW is to find an optimal alignment between the elements of two sequences by warping or stretching one or both of them in the time dimension. Specifically, we first define a wrapping path $W = [w_1, w_2, ...w_L]$ with $w_l = (i, j)_l$, where i and j are the horizontal and vertical coordinates of optimal alignment between the elements of two sequences. Wrapping path must satisfy the following three conditions: 1). $w_1 = (1, 1)$ and $w_L = (n, n)$; 2). $i_1 < i_2 < ... < i_{l-1} < i_l$ and $j_1 < j_2 < ... < j_{l-1} < j_l$; 3). The step size must be limited. Then, DTW can be defined in (1).

$$DTW(X, Y) = \min\{\sqrt{\sum_{l=1}^{L} w_l}/L\} \tag{1}$$

Where L denotes the length of the wrapping path. Combined with the dynamic programming algorithm, Detailed calculations for DTW can be obtained in equation (2).

$$\gamma(i, j) = d(x_i, y_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \tag{2}$$

In equation (2), d is the Euclidean distance calculated for both. γ (i, j) indicates the minimum cumulative matching cost between elements x_1 to x_i in sequence X and elements y_1 to y_j in sequence Y. The final result of the DTW calculation is the cumulative distance γ, which represents the similarity between sequences X and Y.

B. Graph Representation and Fundamental Matrices

In PST-GAT, graph data is the core representation of multi-source load data, storing the encoded sequences and the spatio-temporal adjacencies between different load data. Graph is a kind of data structure, which consists of two sets. A graph $g = (v, E)$ includes a node set v and a edge set E, in which $v_i \in v$ represent a node and $e_{ij} \in E$ represent a edge pointing from v_i to v_j. For a clearer illustration of the methodology in this paper, several fundamental matrices on the graph are introduced.

1). Adjacency Matrix (A): The adjacency matrix $A \in R^{N \times N}$ represents the connections between nodes in a graph, where N denotes the numbers of nodes.

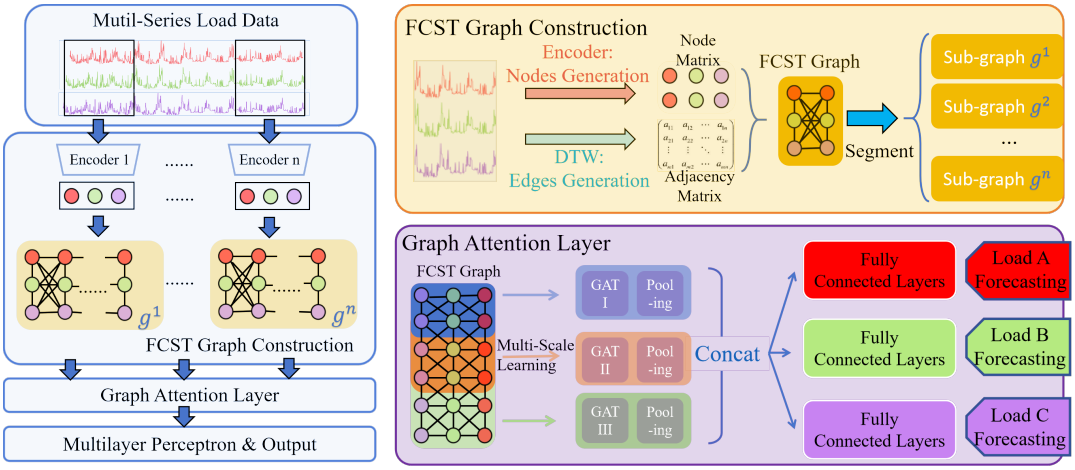

Fig. 2. Overall Structure of PST-GAT

2). Degree Matrix (D): The degree matrix D is a diagonal matrix where D_{ii} is the degree (number of edges connected to) node i.

3). Graph Laplacian Matrix (L): The Laplacian matrix L of a graph is calculated as $L = D - A$, where D is the degree matrix and A is the adjacency matrix. Meanwhile, the Laplacian matrix of a graph is symmetric and positive semi-definite, which makes it eligible for eigen-decomposition.

C. Graph Attention Network

To better learn and capture the spatio-temporal adjacency between load data, GAT is utilized to PST-GAT as the main feature extraction component. GAT [28] is a type of GNN that was introduced to effectively capture the dependencies between nodes in a graph by assigning attention coefficients to neighboring nodes.

Specifically, GAT employs an attention mechanism to assign different importance (attention coefficients) to each neighbor of a node during the aggregation step. For each node, GAT first computes the weights, which are determined by the attention coefficients. Attention coefficients are calculated using a shared self-attention mechanism that considers both the source node and its neighbors. Mathematically, this step can be expressed in equation (3).

$$e_{ij} = \delta([Wh_i \| Wh_j]), j \in N_i \tag{3}$$

Equation (3) represents the calculation of the attention coefficients e_{ij} for node i, where h_i denotes the feature of node i. N_i represent its neighbors, and h_j stands for the feature of node j. W is the weight of linear map, here, the weights-shared linear mapping is applied to the features of nodes i and j. $[.\| .]$ is a splice of two vectors. $\delta(.)$ is a single-layer feed-forward neural network. The attention coefficients e_{ij} obtained also need to be normalized, which can be found in equation (4).

$$a_{ij} = \frac{\exp(Leaky\,\mathrm{Re}\,LU(e_{ij}))}{\sum_{k\in N_i}\exp(Leaky\,\mathrm{Re}\,LU(e_{ik}))} \tag{4}$$

In equation (4), $exp(.)$ denotes exponentiation, and $LeakyReLU$ denotes a kind of activate function. a_{ij} is the finalized attention coefficient.

Then, GAT computes a weighted sum of the feature vectors of its neighbors, transforming the result through a shared learnable linear transformation and activation function. Additionally, GAT employs multiple attention heads in parallel, each with its own set of attention parameters. The outputs from different heads are then either concatenated or averaged. The application of multiple attention heads allow the model to capture diverse relationships or patterns within the graph. This step is defined in (5).

$$h_i'(K) = mean[\delta(\sum\nolimits_{j\in N_i} \alpha_{ij}^k W^k h_j)] \tag{5}$$

Where $h_i'(K)$ is the output signal. ${a_{ij}}^k$ represents the attention coefficients in k-th layer, which can be calculated using (4).

III. Methodology

A. Overall Structure Description

The overall architecture of the proposed short-term forecasting method is illustrated in Fig 2. In summary, the method can mainly be divided into two parts: FCST Graph Construction and Graph Attention. FCST Graph Construction constructs load data from different series and different time points into nodes by encoders. The encoder consists of a 1-D convolutional neural network with fully connected layers. Meanwhile, the DTW technique is applied to create the adjacency matrix of the FCST graph, establishing temporal adjacency and spatial correlation between multi-series load data. Graph Attention focuses on the feature extraction of the constructed FCST graph, learning the local structure of the graph and reducing the feature dimension. Subsequent sections provide more detailed descriptions.

B. FCST Graph Construction

Constructing an FCST graph involves constructing its node features matrix and adjacency matrix. In our method, 1-D convolutional neural network (1-D CNN), which is the

encoder, is utilized to convert load data into node features. The DTW technique is applied to analyze the similarity between two segments of load data, constructing the adjacency matrix.

Specifically, for a section of multi-series load data $x \in R^{L\times D}$, this method first segments it into equal-length intervals in time dimension. The load data is segmented into several sub-sequences $x_{sub} \in R^{L\times M}$. Each segment $x_i \in R^M$ is considered as a node and be encoded individually. The encoder f_e maps the raw data to the feature space: $f_e : x_i \in R^M \rightarrow x_i \in R^N$. Concretely, the encoder consists of two 1-d convolutional layers, two linear transformation layers and nonlinear functions. 1-d convolutional layers are responsible for extracting local patterns and correlations from the load data, and linear transformation layers complete the integration of local features and the mapping of features. The nonlinear function utilizes the ReLU function. The encoder outputs feature vectors $x_i \in R^N$, and all feature vectors collectively form the node feature matrix M_{node}. This step is shown in Fig 3.

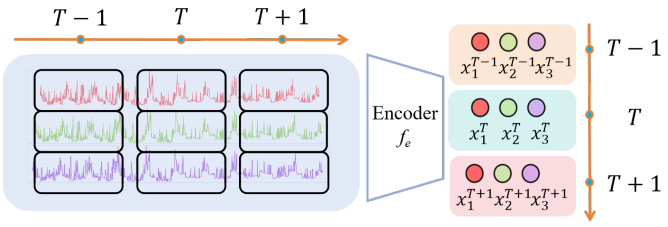

Fig. 3. The Acquisition of Node Features

On the other hand, the DTW method is applied to calculate the correlation coefficient between each segment load data and take it as an element of adjacency matrix M_{adj}. As an example, assume that the 9 nodes in Fig 3 are segmented as a subgraph g_l. Thus, the number of rows or columns of M_{adj} in g_l is 9. Each elements are correlation coefficient between 2 segment load data calculated by DTW. Noticeable, the calculation of correlation is symmetric, i.e. $f(x_1^{T-1}, x_2^{T-1}) = f(x_2^{T-1}, x_1^{T-1}) = a_{12} = a_{21}$. Hence, the M_{adj} is a symmetric matrix, and the FCST graph is an undirected graph.

The FCST graph G_{FCST} of the load data is constructed by combining the node feature matrix M_{node} and the adjacency matrix M_{adj}.In M_{adj}, elements between nodes at different times within the same load data manifest temporal correlations (e.g. x_1^{T-1} and x_1^T), while elements between nodes from different load data at the same time reflect spatial correlations (e.g. x_1^{T-1} and x_2^{T-1}). Spatio-temporal adjacency is represented by elements connecting nodes at different times and from different load data (e.g. x_1^{T-1} and x_2^T). Therefore, G comprehensively links the data across both temporal and spatial dimensions, integrating their impacts and interconnections. The constructed FCST graphs are processed and learned by GAT, and load prediction of multiple sequences is accomplished in the final multi-layer perceptron (MLP).

C. Subgraph Segmentation and Graph Attention

The self-attention mechanism of GAT can effectively learn the FCST graph and capture the relationships between the represented nodes. Furthermore, its powerful expressive ability enables it to handle large-scale graph structures, making it suitable for FCST graphs. For an FCST graph, it first be divided into different subgraphs and learned by GAT graph by graph, then pooling is utilized to downscale the learned features.

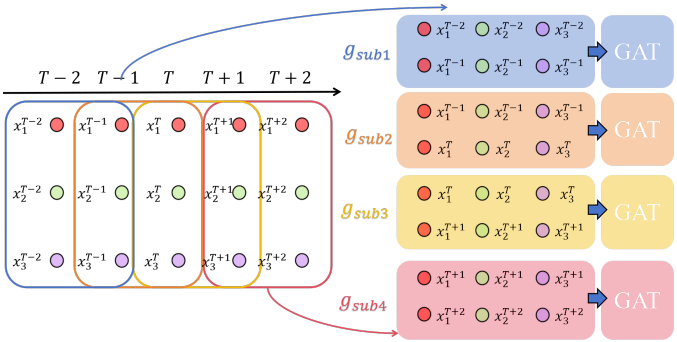

Fig. 4. Subgraph Segmentation of The FCST Graph

Specifically, the FCST graph is first segmented in the time dimension by a fixed length (e.g., $s = 2$). Combined with the concept of sliding windows in convolutional neural networks (CNNs), the segmented subgraphs slide with a stride of length=1 to generate several subgraphs. All segmented subgraphs then be learned by GAT. A visual example is provided in Fig 4. Featuring an FC graph with five time points, each containing three series. A size-two window moves with stride one, leading to four subgraphs obtained. Here, each subgraph contains two time point, each containing two series. Then, GAT is adopted within each subgraph. By segmenting the original graph into subgraphs and processing them individually, it can help the GAT to better capture the relationships between nodes and learn the local structure of the graph.

Meanwhile, a multi-scale parallel processing strategy is adopted in PST-GAT, as shown in Fig 5. In Fig 5, the "Subgraph SL" stands for subgraph segmentation length.

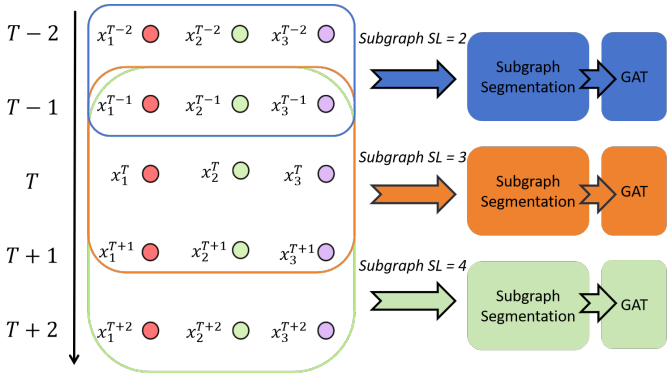

Fig. 5. Multi-Scale Parallel Process

Based on different segmentation lengths, each branch performs subgraph segmentation and GAT feature learning individually. The segmentation lengths determine the size of subgraphs in each branch, representing graph information over different time spans. This enables GAT to learn the FCST graph from different perspectives, enhancing its ability

to capture the spatio-temporal correlation between the load data. GAT can aggregate domain node information of a given node to learn features from different sequences and different time stops. The features extracted by the parallel GAT are concatenated into a vector, which serves as the output feature vector. The output vectors are individually processed and learned by different MLPs to achieve prediction for multiple load sequences.

IV. Experiment

A. Description of Experimental Data

To more objectively demonstrate the validity of the method proposed in this paper for real production operating data, we use historical load data from endpoint equipment at a 10kV production site for forecasting and verification. The load data is provided by China Southern Power Grid and sourced from their database.Specifically, there are three transformers located at the production site, each situated in a different area and supplying power to that respective area. The historical load data includes total active power, three-phase active power, total reactive power, and three-phase reactive power on the secondary side of each transformer from May 2023 through October 2023. It has a time resolution of 15 minutes, meaning that data are recorded every 15 minutes, resulting in a total of 4 data points per hour and 96 data points per day.

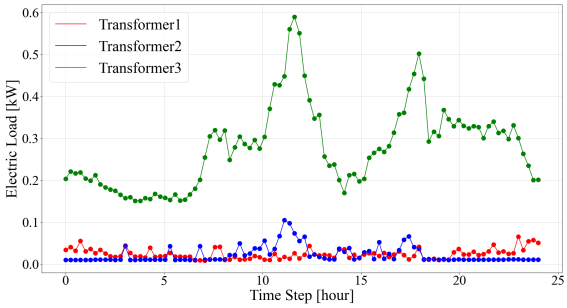

Fig. 6. Active Power of Three Transformers in One Day

In this experiment, the total active power value is used as the indicator of load demand as it directly reflects the power consumption at the output of the transformer. An example is shown in Fig 6. From Fig 6, it can be observed that there is a connection between the change patterns of the secondary side loads of different transformers, especially the peak change trends of transformers 2 and 3 are similar.

We conduct a 1-day ahead load forecasting experiment using PST-GAT. Specifically, we used four days of data from three transformers as a group, with the first three days as inputs to the model and the fourth day as the prediction target of the model. The data for all five months are grouped at an interval of 4 time points (i.e., 1 hour). Each group is considered as a sample, and all samples constitute the total data set. The obtained total data set is divided into training, validation and test sets in the ratio of 0.7:0.2:0.1. The training set is the region where the model learned and trained on the data. The validation set is used to evaluate the model's ability to generate on unseen samples during the training process. The test set was used to evaluate the final accuracy of the model.

B. Experimental Settings

Firstly, all model training and testing in this experiment, including subsequent comparative experimental models, are conducted on a desktop computer with an 13th Gen Intel (R) Core (TM) i9-13900KF (3.00 GHz,16 cores) and an NVIDIA GeForce RTX 4090 (24 GB memory). The operating system is Windows 10.

To compare the predictive ability of PST-GAT alongside other methods, several popular forecasting algorithms were applied to this dataset and compared with PST-GAT. These algorithms include feedforward neural network (FNN), 1-D convolutional neural network (1-D CNN), convolutional gated recurrent unit (CNN-GRU) [30],Self-Attention long short-term memory (Attention-LSTM), and ST-GAT [31]. FNN has 3 hidden layers with hidden units 256, 128 and 96. 1-D CNN consists of 1 convolutional layer and 2 fully connected layers. For the convolutional layer, the parameters are as follows: kernel size is 3, output channels is 8, and padding is 1. The Attention-LSTM model has 2 stacked LSTM layers, 1 layer for attention mechanism and 1 fully connected layer. For the LSTM layer, its hidden size is 128. CNN-GRU has 1 convolutional layer and 2 stacked GRU, and the hidden size of GRU is 128. The parameters of PST-GAT are explained in 4.3.

Besides, to explore the optimal structure of PST-GAT, several comparative experiments for different structures of PST-GAT have also been conducted. Primarily focusing on the number of parallel branches (as mentioned in Chapter 3.3) and other feature extraction architectures (e.g., GNN, GCN, etc.).

C. Model Parameters and Evaluation Metrics

The main parameters for training the PST-GAT model are as follow: the optimizer is Adam; The number of branches and the length of the subgraphs (as mentioned in 3.3) are 3 and [2, 3, 4], respectively (There are 3 parallel GAT feature learning branches, each of which segments the subgraphs of lengths 2, 3, and 4); the training epoch is 50 and the initial learning rate is 0.001. Table I lists detailed parameters applied in PST-GAT.

Mean square error (MSE) and mean absolute percentage error (MAPE) are applied to evaluate the model in test stage. Due to the data values of the samples used in this experiment are all relatively small, the calculated MSE are relatively small, the differences in MSE between different algorithms are also small. Even with small MSE differences, they can still represent the performance of different algorithms. Additionally, MAPE is introduced to further evaluate our comparative experiments, ensuring robust and convincing results. MSE and MAPE are calculated as shown in Equation (6) and (7):

$$MSE = \frac{1}{n} \cdot \frac{1}{l} \sum_{i \le n} \sum_{j \le l} (y_{ij}^{\text{pre}} - y_{ij}^{\text{true}})^2 \tag{6}$$

TABLE II
COMPARATIVE EXPERIMENT BETWEEN PST-GAT AND OTHER ALGORITHMS

	Experiment NO.	1		2		3		4		5	
	Evaluation Metrics	MSE[$\times 10^{-3}$]	MAPE [%]	MSE[$\times 10^{-3}$]	MAPE [%]	MSE[$\times 10^{-3}$]	MAPE [%]	MSE[$\times 10^{-3}$]	MAPE [%]	MSE[$\times 10^{-3}$]	MAPE [%]
Transformer 1	FNN	2.8376	20.47	2.6896	18.83	2.6100	17.18	2.6627	18.02	2.6282	17.59
	1-D CNN	2.3537	16.71	2.5639	17.94	2.8406	18.69	2.6582	17.98	2.3287	15.58
	Attention-LSTM	1.8128	13.07	1.7631	12.71	1.8026	13.01	1.7874	12.89	1.9057	12.75
	CNN-GRU	1.9311	13.93	1.7982	12.97	1.8102	13.05	1.8973	13.68	1.8556	13.38
	ST-GAT	1.4237	10.27	1.5356	11.08	1.2964	9.35	1.4265	10.22	1.4571	10.51
	PST-GAT	**0.7836**	**5.65**	**1.4973**	**10.48**	**1.5002**	**9.87**	**1.1628**	**7.86**	**0.8946**	**5.98**
Transformer 2	FNN	8.2633	36.34	6.9039	29.34	6.9039	27.64	6.2615	21.43	6.3835	29.32
	1-D CNN	10.907	47.96	9.8419	41.82	9.8419	38.64	10.4656	36.84	9.0511	41.52
	Attention-LSTM	4.0268	17.71	4.2971	18.89	3.9618	17.42	4.1361	18.19	4.1073	18.06
	CNN-GRU	5.8824	25.86	4.7343	20.11	4.7343	19.51	3.6791	12.59	4.5711	20.99
	ST-GAT	3.1126	13.69	3.3279	14.64	3.4092	14.97	3.0485	13.41	3.3684	14.81
	PST-GAT	**3.0524**	**13.42**	**2.2066**	**9.37**	**2.1206**	**8.73**	**2.7844**	**9.52**	**3.4337**	**15.77**
Transformer 3	FNN	3.0247	15.58	3.6334	20.09	3.2206	15.18	2.8233	15.41	3.3203	16.48
	1-D CNN	3.5467	18.26	3.2307	17.86	3.4154	16.09	3.4166	18.64	4.2158	20.92
	Attention-LSTM	2.1419	11.03	2.2702	11.69	2.2936	11.81	2.1921	11.29	2.1073	10.85
	CNN-GRU	2.4572	12.65	2.3986	13.26	2.1362	10.06	2.0507	11.19	2.1864	10.85
	ST-GAT	2.0917	10.77	1.8924	9.75	1.9573	10.08	1.8218	9.38	2.2397	11.54
	PST-GAT	**1.2706**	**6.47**	**1.0311**	**5.71**	**1.5275**	**7.20**	**1.5552**	**8.48**	**0.9854**	**4.89**

TABLE I
MAIN PARAMETERS FOR TRAINING PST-GAT

Parameters	Value
Optimizer	Adam
Epoch	50
Learning Rate	0.001
Batch Size	128
Number of Branches	3
Subgraph Length	[2, 3, 4]
MLP Layers	4
Activate Function	ReLU
Drop-out Rate	0.5

$$MAPE = \frac{1}{n} \cdot \frac{1}{l} \sum_{i \le n} \sum_{j \le l} \left| \frac{y_{ij}^{pre} - y_{ij}^{true}}{y_{ij}^{true}} \right| \times 100\% \quad (7)$$

In the equation above, n represents the number of sequences to be predicted; l is the number of time points to be predicted; y^{pre} denotes the predicted load value at a time point, and y^{true} denotes the true load value at a time point.

D. Experimental Results

In this section, firstly, the comparison of performance results for load forecasting between PST-GAT and classical algorithms is demonstrated first. Secondly, several sets of experimental results are illustrated and analyzed for the structure of PST-GAT itself.

(1) Comparative experiments between PST-GAT and classical algorithms

The comparative experiment is conducted on the same dataset and under identical conditions. Optimal hyperparameters are utilized for all models. Fig 7 provides an example of the visualization of the forecasting results of each algorithm in this comparative experiment. Additionally, five independent experiments are conducted to ensure the reliability of the result. Fig 8 provides a histogram of the results of each experiment. Table II provides detailed experimental results. Notably, Fig 7 shows that Transformer 1 fluctuates frequently, making accurate predictions challenging. However, because of its load data values are low and the overall load trends are forecasted well, the resulting prediction errors are small, as reflected in Table II. Additionally, the load data for Transformer 2 exhibits higher frequency fluctuations compared to Transformer 3 and larger magnitude fluctuations compared to Transformer 1, leading to greater prediction errors and making it more unpredictable.

Fig 8 visualize that the prediction result metrics MSE and MAPE of PST-GAT are lower than those of other algorithms for each experiment and each transformer. The forecasting performance gap between the different algorithms can be quantitatively compared from Table II.

It can be observed from Table II that PST-GAT has a significant advantage over the simpler forecasting algorithms, FNN and 1-D CNN, and also offers some benefits over the more advanced CNN-GRU. Additionally, since PST-GAT incorporates an attention mechanism, Attention-LSTM is also used as a comparison model. The results show that although Attention-LSTM also includes an attention mechanism, it fails to capture the spatio-temporal correlations between multi-source load data, leading to lower prediction accuracy compared to PST-GAT. This suggests that the attention mechanism alone is not the sole factor contributing to the superior performance of PST-GAT.

More notably, to validate the effectiveness of temporal adjacency in the multi-source load forecasting task, the ST-GAT model is introduced for comparison with PST-GAT. The primary difference between PST-GAT and ST-GAT is the ability to capture and learn temporal adjacency. While ST-GAT can learn both temporal and spatial correlations between multiple data sources, PST-GAT goes a step further by also capturing temporal adjacency. As summarized in Table II, the experimental results show that PST-GAT achieves higher prediction accuracy than ST-GAT. This indicates that the temporal

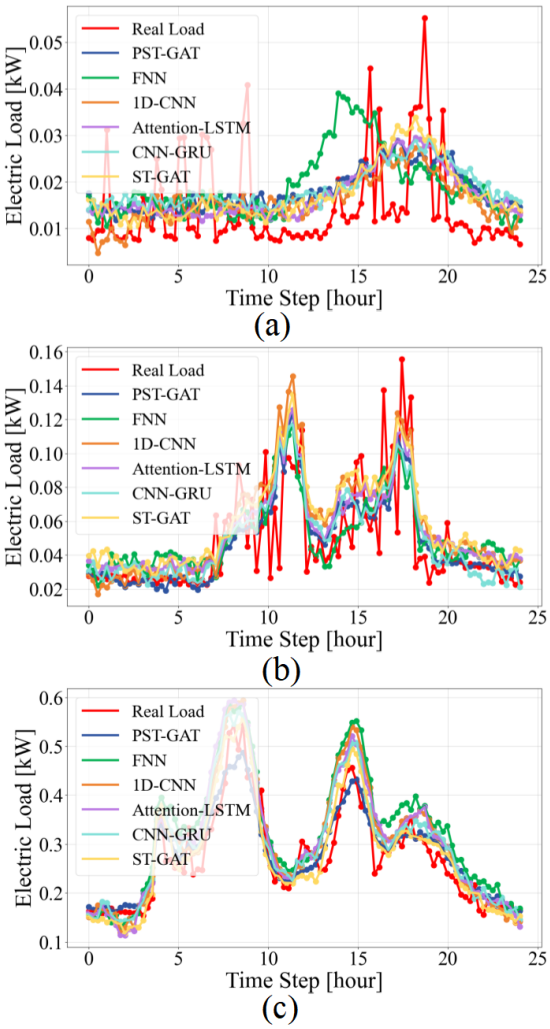

Fig. 7. Visualization of The Forecasting Results of Each Algorithm: (a) Forecast Results of Transformer 1, (b) Forecast Results of Transformer 2, (c) Forecast Results of Transformer 3

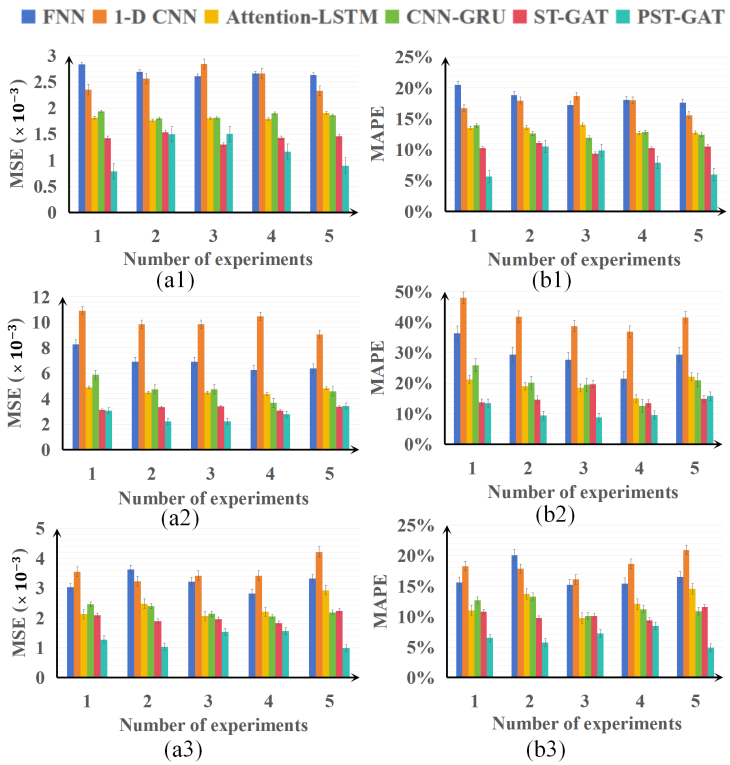

Fig. 8. MSE and MAPE Experiment Results of Each Transformer (a1) MSE Experiment Results of Transformer 1, (a2) MSE Experiment Results of Transformer 2, (a3) MSE Experiment Results of Transformer 3, (b1)MAPE Experiment Results of Transformer 1, (b2) MAPE Experiment Results of Transformer 2, (b3)MAPE Experiment Results of Transformer 3.

adjacency captured by PST-GAT significantly contributes to the precision of multi-source load forecasting. From a different perspective, PST-GAT's ability to capture temporal adjacency indicates that it can learn the correlation between multi-source load data more comprehensively.

The results of the comparison test validate the effectiveness of PST-GAT for short-term load forecasting tasks. This outcome aligns with our expectations, as PST-GAT excels in mining and capturing non-Euclidean correlations among load data from different transformers. Additionally, PST-GAT's architecture enables it to learn and predict load data from multiple sources simultaneously. These advantages highlight the superiority of PST-GAT over existing algorithms.

To better illustrate the relationship between prediction error and training epochs, a line graph is provided in Fig 9. As shown in Fig 9, the validation MSE gradually decreases as the number of epochs increases, reaching a convergence around epoch 40 (Here, the validation MSE is the summation of validation MSEs for the three transformers). However, after epoch 40, the validation error starts to rise, indicating the onset of overfitting, which suggests that training should be halted at this point to prevent further degradation in model performance.

Furthermore, compared to existing methods, PST-GAT requires a higher computational cost, primarily because of its more complex structure designed to capture spatio-temporal adjacency and achieve more accurate forecasting. Reducing the computational cost while maintaining or improving accuracy will be a key focus for future optimizations of PST-GAT.

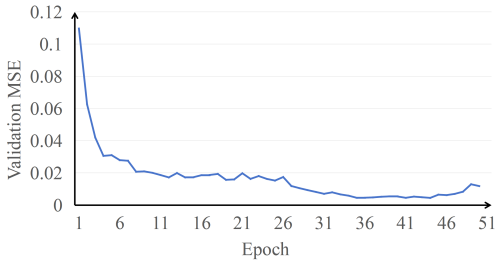

Fig. 9. Validation of MSE in Training Stage

(2) Comparative experiments of different structures of PST-GAT

To explore the optimal structure of PST-GAT, comparative experiments were designed for different structural models. We conducted several sets of experiments focusing on determining the optimal number of branches for subgraph segmentation (as

illustrated in Fig. 5) and the optimal feature extraction module (GAT).

Firstly, multiple sets of experiments with different subgraph segmentation branches were conducted. Multiple subgraph segmentation branches can learn and extract information from different perspectives of load data. However, having too many branches may lead to model overfitting, thereby reducing performance. Additionally, the number of branches influences computational cost. Hence, it is crucial to select an appropriate number of branches. In this set of experiments, PST-GAT is trianed with the number of branches set to 1, 2, 3, and 4, respectively. The lengths of the subgraph segmentations are [2], [2, 3], [2, 3, 4], and [2, 3, 4, 5]. The hyperparameters are set as described in 4.3. The experimental results, assessed using MSE as a metric to evaluate prediction performance, are shown in Table III (The MSE value in Table III represents the sum of the forecast error MSE for the 3 transformers) and Fig 10.

TABLE III
COMPARATIVE EXPERIMENTS RESULTS OF PST-GAT WITH DIFFERENT STRUCTURE

Number of Branches	1	2	3	4
#1	0.008706	0.005285	0.005106	0.005532
#2	0.009576	0.005489	0.004735	0.005886
#3	0.008865	0.005460	0.005148	0.005621
#4	0.008189	0.004891	0.005502	0.005547
#5	0.009481	0.005684	0.005313	0.005262
Mean	0.008963	0.005362	0.005161	0.005570

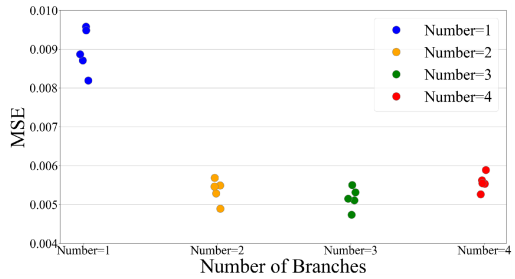

Fig. 10. Scatter Plot of Experimental Results between Different Structure PST-GAT

In Fig 10, the five data points for each number of branches represent the results of five independent experiments respectively, corresponds to table III. Fig 10 presents a visual comparison more intuitively. It is evident that PST-GAT exhibits greater effectiveness when the number of branches is 2 or 3. From the more detailed data in Table III, the average MSE loss value for branch number 3 is lower than 2 $\times 10^{-3}$ indicating slightly higher prediction accuracy compared to branch number 2. It is hypothesized that the 1-branch PST-GAT is unable to capture information from load sequences at multiple scales, whereas the 4-branch PST-GAT may struggle to complete the training process efficiently due to the excessive number of branches. Additionally, the detailed training times are provided in Table IV. It can be observed that as the number of branches increases, the training time of the model also rises, indicating an increase in computational cost. The training time of the model with branch number 3 is approximately 28 minutes higher than that with branch number 2. Therefore, the number of branches of PST-GAT is set to 3 to achieve the highest prediction accuracy. However, the 2-branch PST-GAT is also a viable option for reducing computational cost.

TABLE IV
TRAINING TIMES OF MODEL WITH DIFFERENT STRUCTURE

Number of Branches	1	2	3	4
Training Times/Epoch	9 min	26min	54 min	80 min

Secondly, for the selection of the feature extraction module, we compared GAT with other graph neural networks, which are GCN and ChebConv respectively. This experiment was configured using a 3-branch PST-GAT setup, to simultaneously evaluate the three different architectures under the same conditions. The resulting performance metrics are illustrated in Fig 11.

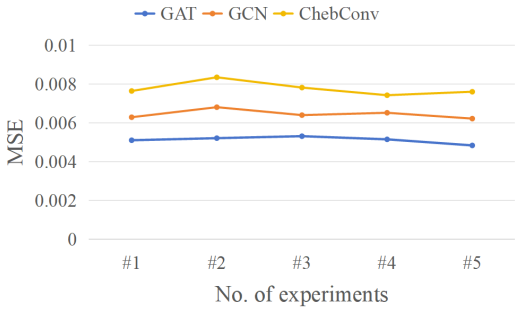

Fig. 11. Experimental Results between Different Structure PST-GAT

The Fig 11 illustrates the comparative performance of the three architectures. As depicted, the GAT-based model outperformed both the GCN-based and ChebConv-based models in terms of prediction accuracy. Specifically, the Mean Squared Error (MSE) loss of the GAT-based model was found to be lower by 1.382 $\times 10^{-3}$ compared to the GCN-based model and by 2.767 $\times 10^{-3}$ compared to the ChebConv-based model. These results indicate that GAT is a highly suitable module for feature extraction in this context. Its ability to capture intricate relationships within the graph data and prioritize relevant features leads to superior prediction performance compared to alternative architectures such as GCN and ChebConv.

E. Evaluation on Extended Dataset

To fully assess the long-term performance of PST-GAT, additional historical load data is acquired and utilized. Specifically, the extra load data spans from October 2022 to May 2023, bringing the total dataset to cover the period from October 2022 to October 2023, a full year. Different from previous experiments, this experiment uses 7 days of data to predict the next 3 days' load. This mainly because load schedules often exhibit weekly patterns, and using 7 days of input data helps the model capture this pattern. Additionally, predicting the next 3 days' load is more challenging and

provides a better evaluation of the performance of PST-GAT. The detailed experiment result can be found in Table V.

TABLE V
FORECASTING EFFECTIVENESS OF PST-GAT ON EXPANDED DATASETS

	Transformer 1	Transformer 2	Transformer 3
MSE($\times 10^{-3}$)	0.8045	2.1752	1.0964
MAPE(%)	5.81	9.57	5.64

From the Table V, it can be oberved that PST-GAT performance well on extended dataset. What is more, through comparing the evaluation metrics of PST-GAT on raw and extended datasets, it can be found that PST-GAT performs better on extended datasets. This might be because load schedules often exhibit weekly patterns, and using 7 days of input data helps the model capture this pattern.

F. Experimental Conclusion

Through the above experiments and analysis, the following conclusion can be obtained: (1) In the task of short-term load forecasting with multiple sequences, PST-GAT demonstrates higher prediction accuracy compared to existing algorithms. This superiority stems from its ability to capture non-Euclidean and spatio-temporal relationships among multiple loads more comprehensively. Additionally, PST-GAT can predict multiple loads sequence simultaneously. (2) The experimental results indicate that the 3-branch parallel GAT-based model exhibited the best prediction performance, which is suggested as the optimal structure of PST-GAT. Specifically, the comparative analysis revealed that the 3-branch parallel model exhibit the best predict performance, and the GAT-based model outperformed both GCN- and ChebConv-based architectures in terms of prediction accuracy.

V. CONCLUSION

A model called PST-GAT for short-term load forecasting is proposed in this paper. The method utilizes 1-D CNN as an encoder and DTW techniques to construct multi-series load data into fully connected graphs. This is done to establish interconnections among load data from different series and time points, forming the temporal adjacencies between different series. Subsequently, the FCST graph is segmented into a series of subgraphs using the sliding window concept, and feature extraction is performed on each subgraph using the GAT algorithm. This process employs a multi-branch parallel processing approach with varying subgraph segmentation lengths in each branch to achieve multi-scale learning and mining of the constructed FCST graph. Finally, the MLP performs the final learning of the feature vectors extracted from multiple branches to complete the forecasting process. Furthermore, due to its unique structure, PST-GAT can forecast multiple series of load data simultaneously. The experimental results show that our method outperforms existing STLF algorithms and demonstrates the potential of PST-GAT for STLF. In future work, we intend to adapt the architecture of PST-GAT for medium-term and long-term load forecasting, enabling it to tackle forecasting tasks across various time horizons. Moreover, we envision applying PST-GAT to additional scenarios within the power system. Specifically, we are interested in exploring its application in photovoltaic forecasting and wind power forecasting tasks.

ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China under Grant (52407037, 62403077, 52377073), the Changsha Natural Science Foundation Youth Project (kq2402022), the Natural Science Excellent Youth Fund of Hunan Provincial under Grant (2024JJ4001), the General Projects of the Hunan Provincial Education Department (22C0138), the Key RD Program Project of Hunan Province under Grant (2024JK2106), and the Changsha University of Science and Technology University-level research projects (CLSJCX23064).

REFERENCES

[1] Sharma, Abhishek, and Sachin Kumar Jain. "A Novel Two-Stage Framework for Mid-Term Electric Load Forecasting." *IEEE Transactions on Industrial Informatics*(2023).

[2] O. Rubasinghe, X. Zhang, T. K. Chau, Y. Chow, T. Fernando and H. H. Iu, "A novel sequence to sequence data modelling based CNN-LSTM algorithm for three years ahead monthly peak load forecasting", *IEEE Trans. Power Syst.*, Apr. 2023.

[3] W. Lin, D. Wu, and B. Boulet, "Spatial-temporal residential short-term load forecasting via graph neural networks," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5373–5384, Nov. 2021.

[4] A. Ahmad, N. Javaid, M. Guizani, N. Alrajeh, and Z. A. Khan, "An accurate and fast converging short-term load forecasting model for industrial applications in a smart grid," *IEEE Trans. Ind. Informat.*, vol. 13, no. 5, pp. 2587–2596, Oct. 2017.

[5] I. K. Nti, M. Teimeh, O. Nyarko-Boateng, and A. F. Adekoya, "Electricity load forecasting: A systematic review," J. *Electr. Syst. Inf. Technol.*, vol. 7, no. 1, pp. 1–19, 2020.

[6] Fan, Fulin, Keith Bell, and David Infield. "Probabilistic real-time thermal rating forecasting for overhead lines by conditionally heteroscedastic auto-regressive models." *IEEE Transactions on Power Delivery* 32.4 (2016): 1881-1890.

[7] P. Pawar, M. TarunKumar, and P. Vittal K., "An IoT based intelligent smart energy management system with accurate forecasting and load strategy for renewable generation," *Measurement*, vol. 152, Feb. 2020, Art. no. 107187.

[8] Y. Dai and P. Zhao, "A hybrid load forecasting model based on support vector machine with intelligent methods for feature selection and parameter optimization," *Appl. Energy*, vol. 279, Dec. 2020, Art. no. 115332.

[9] Madadi, Sajad, Behnam Mohammadi-Ivatloo, and Sajjad Tohidi. "Probabilistic real-time dynamic line rating forecasting based on dynamic stochastic general equilibrium with stochastic volatility." *IEEE Transactions on Power Delivery* 36.3 (2020): 1631-1639.

[10] S.-M. Chen, X.-Y. Zou, and G. C. Gunawan, "Fuzzy time series forecasting based on proportions of intervals and particle swarm optimization techniques," *Inf. Sci.*, vol. 500, pp. 127–139, Oct. 2019.

[11] Dai, Qian, et al. "Stochastic modeling and forecasting of load demand for electric bus battery-swap station." *IEEE Transactions on Power Delivery* 29.4 (2014): 1909-1917.

[12] H. Aprillia, H.-T. Yang, and C.-M. Huang, "Statistical load forecasting using optimal quantile regression random forest and risk assessment index," *IEEE Trans. Smart Grid*, vol. 12, no. 2, pp. 1467–1480, Mar. 2021.

[13] G.-F. Fan, L.-Z. Zhang, M. Yu, W.-C. Hong, and S.-Q. Dong, "Applications of random forest in multivariable response surface for short-term load forecasting," *Int. J. Electr. Power Energy Syst.*, vol. 139, Jul. 2022, Art. no. 108073.

[14] S. M. J. Jalali, S. Ahmadian, A. Khosravi, M. Shafie-Khah, S. Nahavandi, and J. P. S. Catalão, "A novel evolutionary-based deep convolutional neural network model for intelligent load forecasting," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8243–8253, Dec. 2021.

[15] Z. Zhang, P. Zhao, P. Wang, and W.-J. Lee, "Transfer learning featured short-term combining forecasting model for residential loads with small sample sets," *IEEE Trans. Ind. Appl.*, vol. 58, no. 4, pp. 4279–4288, Aug. 2022.

[16] B. Jiang, Y. Liu, H. Geng, Y. Wang, H. Zeng, and J. Ding, "A holistic feature selection method for enhanced short-term load forecasting of power system," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–11, 2023.
[17] Lin, **n, et al. "A hybrid short-term load forecasting approach for individual residential customer." *IEEE Transactions on Power Delivery* 38.1 (2022): 26-37.
[18] C. Li et al., "Interpretable memristive LSTM network design for probabilistic residential load forecasting," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 6, pp. 2297–2310, Jun. 2022.
[19] A. Tascikaraoglu, "Evaluation of spatio-temporal forecasting methods in various smart city applications," *Renew. Sustain. Energy Rev.*, vol. 82, pp. 424–435, Feb. 2018.
[20] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang, "A review of graph neural networks and their applications in power systems," *J. Modern Power Syst. Clean Energy*, vol. 10, no. 2, pp. 345–360, Mar. 2022, doi: 10.35833/MPCE.2021.000058.
[21] M. Yu, Z. Zhang, X. Li et al., "Superposition graph neural network for offshore wind power prediction," *Future Generation Computer Systems*, vol. 113, pp. 145-157, Dec. 2020.
[22] M. Khodayar, S. Mohammadi, M. Khodayar et al., "Convolutional graph autoencoder: a generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 571-583, Apr. 2020.
[23] C. Qin, A. K. Srivastava, A. Y. Saber, D. Matthews, and K. Davies, "Geometric deep-learning-based spatiotemporal forecasting for inverterbased solar power," *IEEE Syst. J.*, vol. 17, no. 3, pp. 3425–3435, Sep. 2023.
[24] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network scale traffic learning and forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.
[25] C. Dinesh, S. Makonin, and I. Bajic, "Residential power forecasting using load identification and graph spectral clustering," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 11, pp.1900-1904, Nov. 2019.
[26] Wang, Y.; Xu, Y.; Yang, J.; Wu, M.; Li, X.; Xie, L.; Chen, Z. Fully-Connected Spatial-Temporal Graph for Multivariate Time Series Data. *arXiv* 2023, arXiv:2309.05305.
[27] T. Teeraratkul, D. O'Neill, and S. Lall, "Shape-based approach to household electric load curve clustering and prediction," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5196–5206, Mar. 2017.
[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, "Graph attention networks", *Proc. Int. Conf. Learn. Representations*, 2018.
[29] M. -S. Ko, K. Lee and K. Hur, "Feedforward Error Learning Deep Neural Networks for Multivariate Deterministic Power Forecasting," in *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6214-6223, Sept. 2022.
[30] Lin, Jun, et al. "Short-term load forecasting based on LSTM networks considering attention mechanism." *International Journal of Electrical Power & Energy Systems* 137 (2022): 107818.
[31] Y. Huang, H. Bi, Z. Li, T. Mao and Z. Wang, "STGAT: Modeling spatial-temporal interactions for human trajectory prediction", *Proc. Int. Conf. Comput. Vis.*, pp. 6272-6281, 2019.

Zhuo Long received the Ph.D. degree in Electrical Engineering from College of Electrical and Information Engineering at Hunan University, Changsha, P. R. China, in 2021. And he received the B.S. degree and the M.S. degree in Electrical Engineering and Automation from School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan, P. R. China, in 2012 and 2015, respectively. He is currently a lecturer in College of Electrical and Information Engineering at Changsha University of Science and Technology, Changsha, P. R. China. His current research interests include fault diagnosis, intelligent prognosis, fault information visualization and information fusion.

Zhiyuan Xu received the B.S. degree in electrical engineering and automation from Hunan University of Technology, Hunan, China, in 2022. He is currently working toward the M.S. degree in electrical engineering with Changsha University of Science & Technology, Changsha, China. His current research interests include intelligent artificial algorithms, motor fault diagnosis, and short-term load forecasting.

Gongping Wu (Member, IEEE) was born in Jiangxi, China, in 1992. He received the B.S. degree in electrical engineering and automation from Jinggangshan University, Jiangxi, China, in 2014, the M.S. degree in electrical engineering and automation from the Hunan University of Technology, Hunan, China, in 2017, and the Ph.D. degree in electrical engineering from the College of Electrical and Information Engineering, Hunan University, Changsha, China, in 2021. He studied with Tongji University, Shanghai, China, from September 2011 to September 2013. From 2019 to 2021, he was a Visiting student with the Department of Electrical Engineering, Technical University of Denmark, Lyngby, Denmark.

Since September 2021, he has been an Assistant Professor with the College of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha. His current research interests include motor fault diagnosis, model predictive control, and permanent magnet synchronous motor systems.

Feng Deng (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Changsha University of Science and Technology, Changsha, China, 2006 and 2009, respectively, and the Ph.D. degree from Hunan University, Changsha, China, 2018. She is currently a Professor with the School of Electrical and Information Engineering, Changsha University of Science and Technology. Her research interests include power system microcomputer protection, and traveling wave fault protection and location.

Xiangyuan Chen was born in 2000 in China.He received the B.S. degree in electrical engineering and automation from Changsha University of Science Technology, Changsha, China, in 2022. He is currently working toward the M.S. degree in electrical engineering with Changsha University of Science & Technology, Changsha, China. His current research interests include power system operation and control, optimal operation of integrated energy systems.

Wenshan Feng received the B.S. degree from North China Institute of Science and Technology, Langfang, China, in 2022. She is currently pursuing the M.S. degree with Changsha University of Science and Technology, Changsha, China. Her current research interests include fault diagnosis, intelligent artificial algorithm, fault information visualization,and information fusion.

Zhiwen Huang received the B.S. degree in electrical engineering and automation from East China Jiaotong University, Nanchang, China, in 2023. He is currently working toward the M.S. degree in electrical engineering with Changsha University of Science & Technology, Changsha, China. His current research interests include fault diagnosis, intelligent artificial algorithms, fault information visualization, and information fusion.