IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 March 2024, accepted 27 March 2024, date of publication 1 April 2024, date of current version 26 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3383912

RESEARCH ARTICLE

Enhancing Short-Term Power Load Forecasting With a TimesNet-Crossformer-LSTM Approach

JUN HE, KUIDONG YUAN, ZIJIE ZHONG, AND YIFAN SUN
School of Information Engineering, Nanchang University, Nanchang 330031, China

Corresponding author: Jun He (416100220279@email.ncu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62066025.

ABSTRACT Efficient and accurate short-term electric load forecasting plays a significant role in energy conservation and reducing carbon emissions. Recurrent neural networks (RNN) and their derived deep learning models have continuously improved the accuracy of short-term load predictions. However, traditional deep learning models, constrained by the one-dimensional structure of time series data, struggle to capture the relationships within and between periods. And when performing load forecasting tasks, these models tend to establish temporal relationships in the time dimension while overlooking the relationships between different feature variable dimensions. In order to address both, this paper proposes a Crossformer-based TimesNet-LSTM method for short-term electric load forecasting. The proposed approach takes historical load data as input and leverages the unique structure of TimesNet to convert the one-dimensional time series into a two-dimensional space for information extraction. The Crossformer model with double attention mechanisms is then employed to capture the relationships between sequences, time, and feature variables in different dimensions. Finally, the LSTM computes the output results. Experimental calculations on publicly available datasets from Australia and the United States demonstrate the superior performance of the proposed model compared to traditional single models and other hybrid models in short-term forecasting of multidimensional electricity load data. The Mean Absolute Percentage Error (MAPE) achieved on the Australian dataset is 0.52%, while on the U.S. dataset it is 0.53%. These outstanding results highlight the universality and robustness of the model. The proposed Crossformer-based TimesNet-LSTM method not only overcomes the limitations of traditional one-dimensional deep learning models but also enhances the accuracy of short-term electric load forecasting. Its application has significant implications for energy saving and carbon emission reduction.

INDEX TERMS Time series, TimesNet, crossformer, two stage attention, long and short-term memory neural networks.

I. INTRODUCTION

With the goal of ‘‘carbon peak and carbon neutrality’’ being proposed [1], the electricity market, which is mainly composed of thermal power generation, needs to take action to reduce carbon emissions. Efficient and accurate short-term load forecasting plays a significant role in energy conservation and carbon emissions reduction [2]. Therefore, it is particularly important to establish an efficient, accurate, and stable load forecasting model [3].

In the past, many single models have been applied to short-term load forecasting. Among them, multivariate linear regression [4] is a method that fits the load curve to predict future variables by continuously fitting the input variables and output variables. Time series analysis [5] is a classical method for load forecasting. It analyzes the stochastic process of historical load sequences, sets corresponding parameters, and builds models, followed by estimating the parameters in the model. With the development of machine learning technology, it has been widely used in load forecasting, including support vector machine (SVM) models [6], random forests [7]. and the emergence of recurrent neural

The associate editor coordinating the review of this manuscript and approving it for publication was Emilio Barocio.

networks (RNNs) which have spawned a series of models for time series prediction. RNN has good memory for short sequence data changes, but it faces problems such as vanishing gradients and exploding gradients when dealing with long sequences [8]. And unlike RNN which overlays all input information repeatedly, LSTM can forget the unimportant temporal information, which shows the advantage in dealing with long sequence memory task while solving the gradient explosion of RNN. Therefore, LSTM has been widely used in time series load forecasting [9], [10]. Gated recurrent unit (GRU) combines the forget gate and input gate of the LSTM model, which simplifies the structure compared to LSTM [11]. Bi-directional long short-term memory neural networks (BiLSTM) further improve the global and comprehensive feature extraction by training LSTM in both forward and backward directions for time series [12], [13]. Convolutional neural networks (CNN) have the advantage of sharing convolution kernels and are effective in extracting hidden information from historical data when dealing with multidimensional feature vectors [14]. TimesNet, proposed in the literature [15], extends one-dimensional time series data to a two-dimensional space for analysis. It achieves good performance in long and short-term time prediction by extracting features within and between periods. With the introduction of Transformers, the self-attention mechanism has been widely applied to time series prediction, achieving remarkable results [16], [17]. Crossformer, proposed in the literature [18], applies TSA (Two-Stage Attention) to extract the relationships between features and time, as well as between different features, achieving good results in high-dimensional time series prediction.

However, a single model is not sufficient to meet the requirements of prediction accuracy. Therefore, combining the advantages of various models is the purpose of constructing ensemble models. The literature [19] proposes a hybrid model based on Informer and VMD (Variational Mode Decomposition)-LSTM for time series prediction. It improves the accuracy of long-term time series prediction by capturing the individual long-term correlations between time series outputs and inputs. The literature [20] combines CNN-BiLSTM and improves the accuracy of time series prediction through feature selection. The literature [21] combines CNN-GRU and adds an attention mechanism to reduce the loss of historical information and enhance the attention weight of important information. The literature [22] combines CNN-ResNet by first using CNN to extract the correlation and features of time series and then using ResNet to extract features of other related data to address the degradation problem and improve the accuracy of prediction. The literature [23] proposes a combination model based on double-layer XGBoost (extreme gradient boosting) algorithm, which avoids data feature standardization and reduces the impact of missing data fields, showing good learning performance in ultra-short-term load prediction tasks. The literature [24] proposes a short-term load forecasting model based on VMD (Variational Mode Decomposition). First, the original load is decomposed into several regular and random subsequences using the VMD-CSA method. Then, two different time series models are used to predict the regular and random subsequences, and the final prediction result is the sum of the predicted results of each subsequence. The literature [25] proposes an electric power prediction method based on Empirical Mode Decomposition (EMD) and BiLSTM. Firstly, EMD separates the time series into components with different resolutions. Then BiLSTM is used for accurate prediction of each intrinsic mode function, and finally, the predictions of each component are combined to obtain the overall prediction. The literature [26] predicts the load using different artificial neural networks. Then, the multi-objective pollination algorithm is used to optimize the weight coefficients of each individual model. By integrating the strengths of various models, ensemble models can achieve higher prediction accuracy. In order to overcome the dimension limitation of traditional prediction models in time series, this paper transforms the one-dimensional sequence into a two-dimensional space for information extraction. This greatly discovers the relationships at various time points and time periods. By employing a two-stage attention mechanism, the model is able to enhance robustness and accuracy in mining long sequence information.

In conclusion, to improve the accuracy of short-term load forecasting, this paper proposes a model for short-term load forecasting based on TimesNet-Crossformer-LSTM. The proposed method segments the input features by equally dividing them using TimesNet and extracts features through the internal convolution of the model to explore the relationships between segments. Then, the Crossformer model uses two-stage attention to extract the relationship between input vectors and time, as well as the relationships between each input feature and other features. Finally, LSTM is used to learn the time series, perform error correction, and output the prediction results. The model is evaluated using actual load data from a region in Australia and actual weather data from a region in the United States. The experimental results comparing with various common prediction models show that the proposed TimesNet-LSTM model based on Crossformer achieves higher accuracy and robustness in multi-dimensional input feature data. The main contributions of this paper include:

1) Analyzing and selecting feature parameters with higher correlation to power demand using Pearson correlation coefficient, providing the prediction model with feature data that is more closely related to load.
2) Comparing experiments with different sliding window widths and selecting the fixed sliding window width that best suits the model to capture the periodicity embedded in the time series.
3) Utilizing TimesNet to decompose the complex time series into different periods through a modular structure, transforming the original one-dimensional time

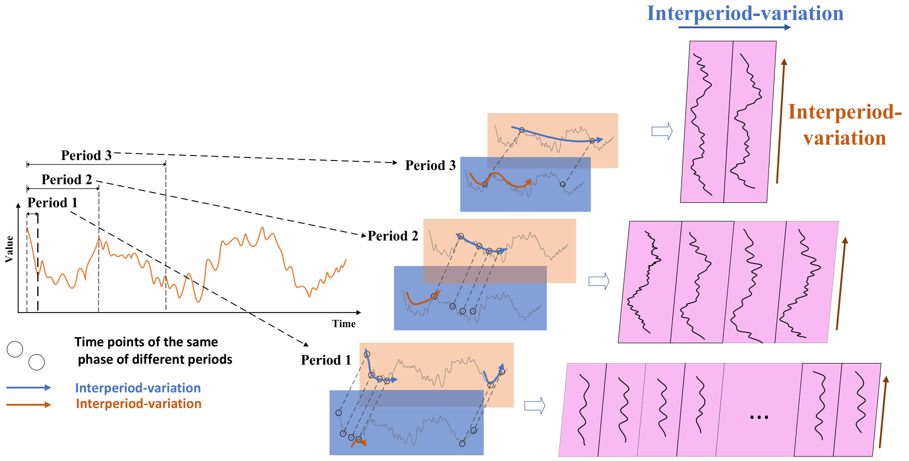


FIGURE 1. Multi-periodicity and time two-dimensional variation diagram of time series.

series into a two-dimensional space, and better extracting the relationship between periods within and between the time series.

4) Employing two-stage attention to perform attention not only in the time dimension but also between feature dimensions, which yields better results in load prediction tasks with multi-dimensional features.
5) A load forecasting model based on TimesNet-Crossformer-LSTM is proposed in this study. The model is evaluated using two public datasets, one from Australia and the other from United States. Experimental results demonstrate that the proposed model outperforms other single models and hybrid models in terms of prediction.

II. METHODOLOGY

A. TIMESNET

To overcome the limited representational capacity of one dimensional time series, TimesNet introduces a novel multi-cycle perspective for analyzing temporal changes. As depicted in Figure 1, it is evident that when predicting time series, there exists a closely interrelated nature between each input time point and its adjacent time points, as well as time points with similar phases but in different periods. By exploring the relationships within and between time segments, improved results can be achieved in sequence prediction tasks, thereby significantly advancing machine learning processes. Nevertheless, constrained by the inherent one-dimensional structure of time series, the original sequence struggles to simultaneously capture two distinct temporal variations: within-cycle and between-cycle changes. Innovatively, TimesNet extends one-dimensional time series data into a two-dimensional space for analysis. As illustrated in the figure, by folding the one-dimensional time series based on multiple cycles, multiple two-dimensional tensors can be obtained. The columns and rows of each tensor reflect within-cycle and between-cycle temporal variations, effectively incorporating two-dimensional temporal changes into the analysis.

TimesNet employs a modular structure to decompose complex time series into different periods, and their periodicity can be computed using the fast Fourier transform (FFT) in the time dimension, the process is shown below:

$$\begin{cases} A = Avg(Amp(FFT(X_{1D}))) \\ \{f_1, f_2 \cdots, f_k\} = \underset{f_* \in \{1,2,\cdots,[T/2]\}}{\arg Top(k)} \\ p_i = [T/f_i], \, i \in \{1, 2, \cdots, k\} \end{cases} \tag{1}$$

where FFT($\cdot$) denotes the Fast Fourier Transform, Amp($\cdot$) denotes the computation of the amplitude, Avg($\cdot$) denotes the computation of the mean value of the C-dimensional features, and $A \in \mathbb{R}^T$ represents the intensity of each of the frequency components in X_{1D}.

After obtaining the various periodic components and their corresponding frequencies from the Fast Fourier Transform, they are converted into two-dimensional tensors to represent the two-dimensional temporal variations in the following process:

$$X_{2D}^i = Reshape_{p_i, f_i}(Padding(X_{1D})) \tag{2}$$

where $X_{2D}^i \in R^{p_i \times f_i \times C}$ denotes the 2D tensor corresponding to the frequency f_i. The process is clearly demonstrated by the lower left part of Figure 2: the 2D time series tensor obtained in the case of the selected period and frequency, when the selected period or frequency is different, the resulting 2D time series tensor is also very different, so it is necessary to further mine for deeper information.

Since the 2D tensor has 2D localization, in order to extract the 2D temporal variations using 2D convolution to extract the information, the classical Inception model is selected and the extraction process is shown as follows:

$$X_{2D}^i = Inception\left(X_{2D}^{l,i}\right), \quad i \in \{1, \cdots, k\} \tag{3}$$

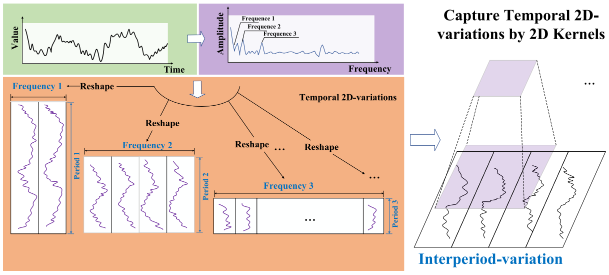


FIGURE 2. Time series of two-dimensional structures and changes in two-dimensional nuclear capture time.

For the extracted temporal features, TimesNet transforms them back into one-dimensional space for information aggregation. The process is as follows:

$$X_{1D}^{i} = Trunc\left(Reshape_{1,((p_i, f_i)}\left(X_{2D}^{i}\right)\right),\ i \in \{1, \cdots, k\} \quad (4)$$

where Trunc(·) denotes the removal of the 0 supplemented by the operation of step Padding (·).

In order to fuse the multi-period information, we perform a weighted summation of the extracted 2D temporal representations, and the chosen summation weights are the corresponding frequency intensities obtained in the previous steps, as shown in the following procedure:

$$A_{f_1}, \cdots, A_{f_k} = \text{Softmax}(A_{f_1}, \cdots, A_{f_k}) \quad (5)$$

$$X_{1D}^{l} = \sum_{i=1}^{k} A_{f_1}^{l-1} \times X_{1D}^{l,i} \quad (6)$$

Through the design of transforming 1D time series into 2D space, TimesNet implements the process of modeling time series changes by extracting 2D time series changes from multiple cycles separately and then fusing them adaptively.

B. CROSSFORMER

The Crossformer model mainly consists of three major modules. The Dimension-Segment-Wise Embedding module primarily divides the time series output of the previous model into patches, obtaining patch embedding, which serves as the input for the subsequent model. It also divide the time series of each variable into blocks according to a certain window, and maps each block through full connections. The Two-Stage Attention Layer focuses on two stages of the time series, namely the time dimension attention and the variable dimension attention. The tiered encoder-decoder uses different patch sizes and partitioning methods to form a tiered codingdecoding structure. In this paper, only the first two modules are selected in order to reduce the complexity of the combined model.

1) DIMENSION-SEGMENT-WISE EMBEDDING (DSW)

The main goal of this module is to divide the time series output of the previous model into patches, obtaining patch embeddings that serve as inputs to the subsequent module. It also divides the time series of each variable into blocks based on a specific window size, and maps each block through fully connected layers. For instance, for the input sequence $X_{1D} \in \mathbb{R}^{T \times C}$ with a length of T and a feature variable dimension of D, the entire input consists of $T/L \times D$ segments. Therefore, the size of the i-th segment in the d dimensional space is denoted as $x_{i,d}^{(s)} \in \mathbb{R}^{1 \times L}$ and then each segment is embedded into a vector using positional embedding and linear projection, resulting in a two-dimensional vector $W = \{h_{i,d} | 1 \leq d \leq D\} \in \mathbb{R}^{T/L \times D \times d_{\text{mod}\,el}}$

2) TWO-STAGE ATTENTION (TSA)

The two-stage attention refers to the time dimension and the spatial dimension. The input first undergoes a layer of time attention, independently modeling the time for each sequence. Subsequently, it proceeds through a layer of spatial attention, aligning and encoding different variables at each time step, which is also a core aspect of Crossformer. The attention in the time dimension is similar to the standard Transformer model, where each variable's time series is transformed into a patch and then subjected to attention mechanism for computing correlations. Since attention is applied in the time dimension, multi-head self-attention (MSA) can be directly applied to each vector dimension, which is represented as follows:

$$\begin{cases} Z_d^{time} = \text{LayerNorm}(Z_d + MSA^{time}(Z_d, Z_d, Z_d)) \\ Z^{time} = \text{LayerNorm}(Z^{time} + MLP(Z^{time})) \end{cases} \quad (7)$$

where 1 ⩽ d ⩽ D, where MLP denotes a multilayer feed-forward network and MSA(Q,K,V) denotes a multi-headed self-attention layer, with all dimensions sharing an MSA layer. After this stage, the dependencies between time periods of one dimension are found in Z time and become inputs to the cross-dimensional stage of attention. Z_d^{time}, Z^{time} denotes the output of the MSA and MLP.

For the feature dimension attention layer, Spatial dimension attention refers to aligning the different variables at each time step. As shown in the figure below, the purpose of spatial dimension alignment is to find the pairwise relationships between variables at different time steps, in order to more deeply capture the influence of one variable on another. As shown in figure b above, this is achieved by performing self-attention on the variable dimension.

However, attention on the feature dimension will significantly increase computational complexity. Therefore, the model uses a routing approach, which introduces a certain number of intermediate variables. By first aggregating the information of the variable's different time steps into intermediate variables using one layer of attention, and then using self-attention between the intermediate variables and the original time series, the intermediate variables can be seen as a pathway or a convenient "intermediary" for simplified computations. By first passing the information to the intermediary and then allowing the intermediary to interact with the original sequence, the efficiency is increased, achieving the goal of simplification. The diagram c in the figure above is an

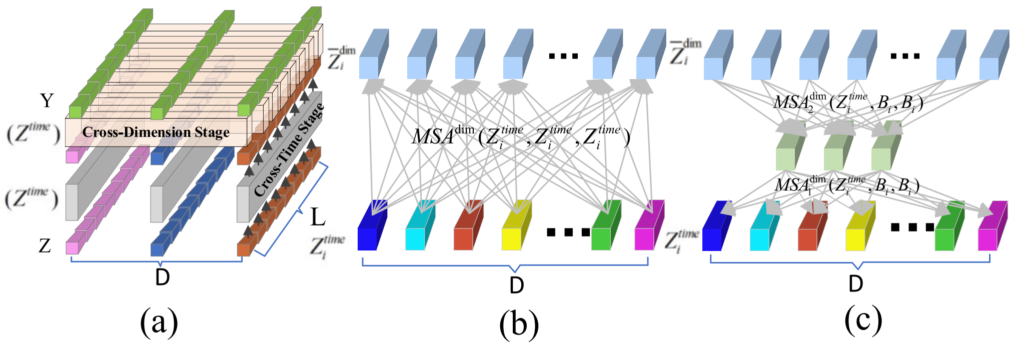


FIGURE 3. **Two-stage attention layer and cross-stage routing mechanism diagram.**

illustrative diagram, and the specific computational formula is as follows:

$$\begin{cases} B_I = MSA_1^{\dim}(R_i, Z_i^{time}, Z_i^{time}), \ 1 \le i \le L \\ \overline{Z}_i^{\dim} = MSA_2^{\dim}(Z_i^{time}, B_i, B_i), \ 1 \le i \le L \\ Z^{\dim} = LayerNorm(Z^{time} + \overline{Z}^{\dim}) \\ Z^{\dim} = LayerNorm(Z^{\dim} + MLP(Z^{\dim})) \end{cases} \tag{8}$$

where $R \in \mathbb{R}^{L \times c \times d \text{ mod } el}$ (c is a constant) is the learnable vectors used as routers. $B \in \mathbb{R}^{L \times C \times d_{\text{model}}}$ is the aggregated messages from all dimensions. $\overline{Z}^{\dim}$ represents the output of the routing mechanism. All time steps share the same $MSA_1^{\dim}$, $MSA_2^{\dim}$。

C. LSTM

LSTM has three gates, input gate, forget gate and output gate. The ability to pass on information is removed or increased by carefully setting up the gates. In other words, for unimportant information, LSTM forgets it through computation so that too much information is not passed on to increase computation. For important information, the transfer capability is increased so that it can be remembered in long sequence tasks. The structure of which LSTM is shown in Fig. 4:

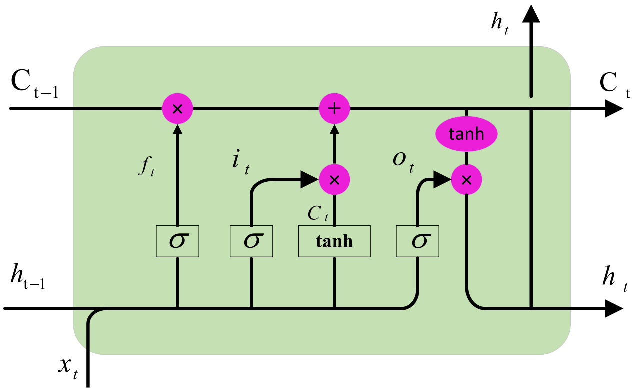


FIGURE 4. **Structure of LSTM.**

Where C_{t-1} represents the memory cell of the previous moment, x_t represents the input information of the current moment, h_{t-1} represents the hidden state of the previous moment, h_t represents the hidden state to be passed to the next moment; σ is the sigmoid function, which can normalize the input value to between 0 and 1. tanh function is to normalize the input data to between -1 and 1.

The first part of the LSTM module is the forgetting gate, which determines what information should be retained. The weight matrix W_f is multiplied by the splicing matrix of the output h_{t-1} from the previous step and the current input x_t, plus the bias value b_f, and then a sigmoid nonlinear mapping is performed, followed by the output of a vector f_t with values ranging from 0 to 1, in which the closer the value is to 1 means that it should be retained, and the closer the value is to 0 means that it should be discarded, and then finally it is multiplied by C_{t-1}, and the formula for the forgetting gate is:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{9}$$

The second part is the input gate, which is used to add memories to the memory cells and decide what new information to store in the memory cells. The input gate is obtained by a nonlinear mapping to a matrix i_t, and the values in the matrix i_t determine whether the corresponding information in C_t is retained or discarded. The formula for the input gate is:

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{10}$$

The third part is the output gate, which determines what the final output of the model will be, and the process is carried out in two parts: the first half uses a sigmoid function to obtain o_t,whose matrix value determines which portion of the memory cells needs to be output, and the formula for o_t:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_0) \tag{11}$$

The second half takes the new memory cells and processes them with the tanh function so that the output value is normalized to [−1,1] and multiplied by o_t, to control the output part. where h_t is given by the following formula:

$$h_t = o_t * \tanh(C_t) \tag{12}$$

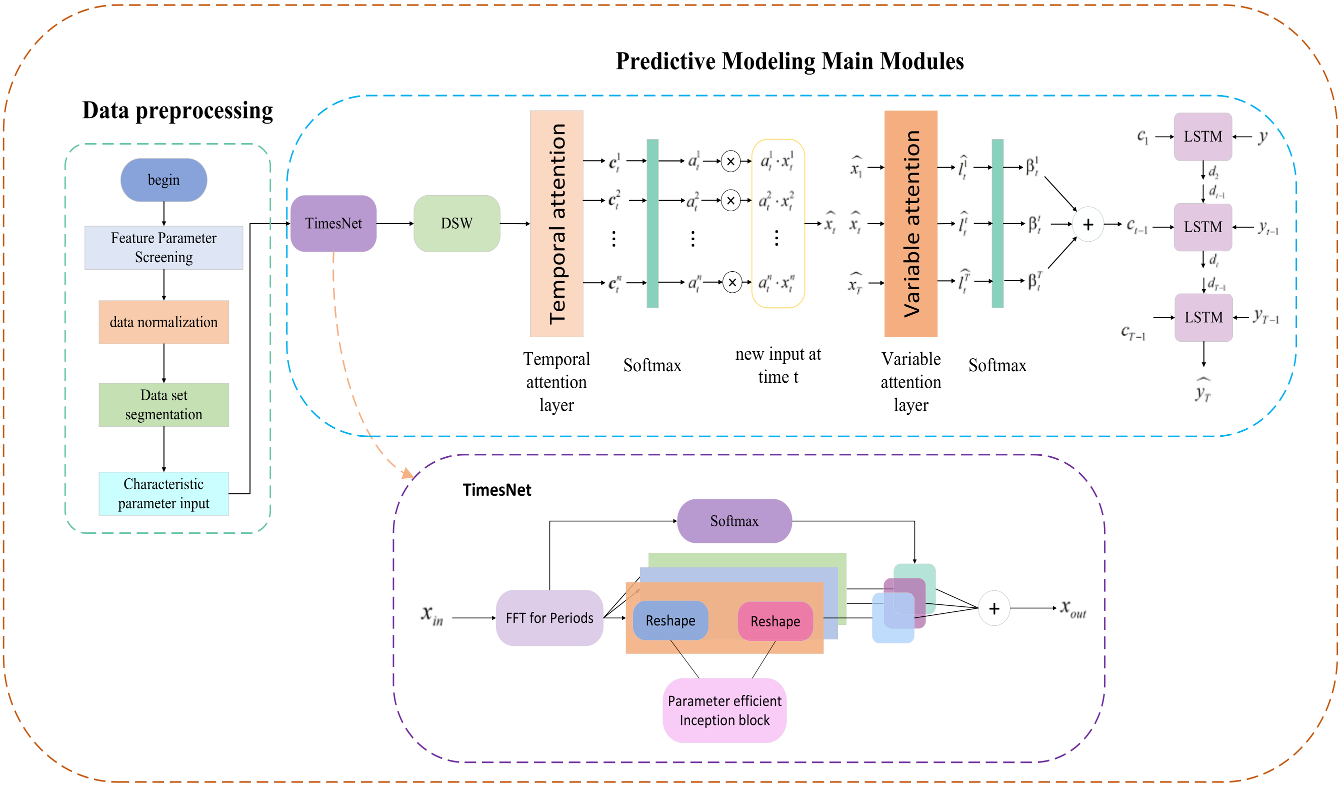


FIGURE 5. Structure of TimesNet-Crossformer-LSTM model.

III. SHORT-TERM ELECTRICITY LOAD FORECASTING MODEL BASED ON TIMESNET-CROSSFORMER-LSTM

Short-term electricity load forecasting involves predicting unknown loads by extracting information from historical load sequences. The conventional approach in machine learning often only captures one-dimensional features from time series data, disregarding the intricate relationships between each time point and its neighboring points as well as those in different periods but with the same phase. Consequently, this limitation frequently leads to unsatisfactory prediction accuracy.The unique structure of TimesNet enables the transformation of one-dimensional time series into two-dimensional space for information extraction, effectively capturing both intra-period and inter-period variations, forming the basis for improved prediction accuracy. Unlike traditional attention mechanisms, the Crossformer network employs a two-stage attention mechanism to provide weights both in the temporal dimension and to explore relationships between feature dimensions. This enables the network to capture deeper information, enhance the transmission and memory of important information, and better capture the interdependencies between long-distance time series. Additionally, the introduction of LSTM networks addresses the issues of gradient vanishing and exploding in RNNs. The inclusion of the feature information extracted by Crossformer into the LSTM network enables improved extraction of the periodic and varying characteristics of load demands in historical load data. This amalgamation facilitates the LSTM network in capturing the temporal dependencies and patterns inherent in the data, thereby enhancing its capacity to model the cyclical and fluctuating nature of load requirements. The proposed TimesNet-Crossformer-LSTM model combines the strengths of all three models and learns from multidimensional feature time series. This allows the model to uncover the rich underlying periodicity and feature patterns, resulting in improved prediction accuracy for short-term power load forecasting.

A. MODEL STRUCTURE

The Timesnet-Crossformer-LSTM structure used in this paper is shown in Figure 5. Each layer in the model is described as follows:

1) Data preprocessing layer: Firstly, feature selection is performed on the imported dataset using the Pearson correlation coefficient analysis method. The selected features are then normalized, and the dataset is subsequently divided into training, validation, and testing sets. The training and validation sets are inputted into the model for training. Its input can be represented by $X = [x_1 \cdots x_{i-1}, x_i \cdots x_z]^T$.
2) TimesNet layer: TimesNet focuses on transforming the input one-dimensional time series into a two-dimensional space for information extraction, thus enabling the extraction of intra-periodic and inter-periodic relationships in the time series. TimesNet is comprised of stacked TimesBlocks. As shown in the figure, the input vector first undergoes the

FFT layer, which uses the fast Fourier transform along the time dimension to calculate the periodicity of the input sequence and thereby transform the input data into the frequency domain for time series analysis. This is followed by reshaping the data into a two-dimensional tensor to represent the two-dimensional temporal changes. To extract the representation of two-dimensional temporal changes, we utilize 2D convolution for information extraction, employing the Inception model due to the two-dimensional tensor's local spatial properties. In order to extract temporal features, the two-dimensional tensor outputted by Inception is then transformed back into one-dimensional space for information aggregation. Subsequently, the obtained one-dimensional array and its corresponding frequency strengths are weighted and summed, resulting in the output of TimesNet.

3) Crcossformer layer: As shown in the figure, the output of TimesNet is used as the input to the Crossformer, which first goes through the DWG to segment the time series into patches, and the resulting patch embedding is used as the input to the subsequent steps. After that, the TSW performs the attention on time dimension and the attention between feature dimensions to find the relationship between sequence and time, feature and feature, and then output the results.

4) LSTM layer: The LSTM layer learns the feature vectors extracted by the Crossformer layer. A single-layer LSTM structure is constructed to comprehensively learn the characteristics of the input time load sequence and uncover the inherent patterns of variation within the sequence. This model adopts a single-layer LSTM.

The output layer activates, linearizes, and inverse normalizes the information tensor computed by LSTM to obtain the final predictive results. Through experimental comparison in this study, the activation function most suitable for this model was determined to be the hyperbolic tangent (tanh) function. In summary, we ingeniously constructed the structure of the model to leverage the advantages of each module. We first applied dimensionality elevation-convolution-dimensionality reduction on the original time series. Then, through two rounds of attention mechanisms, we scored each point in the sequence and each dimension within the sequence, enabling the exploration of relationships between two points with significant temporal gaps. This approach provides a more feasible and accurate method for sequence prediction.

B. FEATURE FILTERING

The Australian public data used in this study contains six dimensions of features: time, dew point temperature, dry bulb temperature, wet bulb temperature, humidity, electricity price and load. The feature information is sampled every 30 minutes, resulting in 48 data points collected per day. In the process of short-term power load forecasting, numerous factors can affect load variations, such as temperature, weather, and season. However, it is not always beneficial to input as many features as possible into the deep learning model. Excessive dimensions can lead to the curse of dimensionality during the model training process, significantly reducing prediction accuracy and diminishing model interpretability. Therefore, the first step in data preprocessing is to select the most favorable feature variables for prediction. This approach not only reduces the computational load of the model but also enhances the model's ability to mine information from the feature parameters, reduces the occurrence of overfitting, and improves the predictive efficiency and generalization ability of the model, ultimately enhancing prediction accuracy

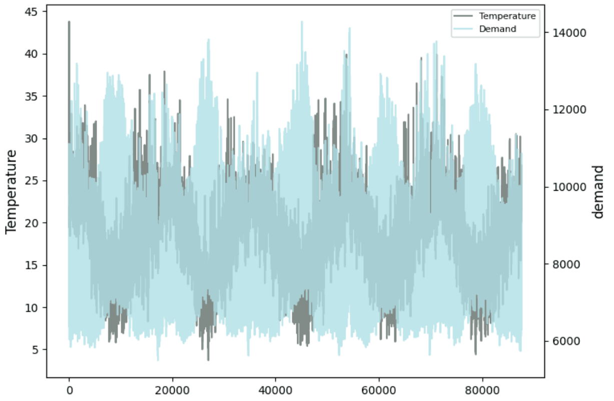


FIGURE 6. Electricity load and temperature profiles for Australian datasets.

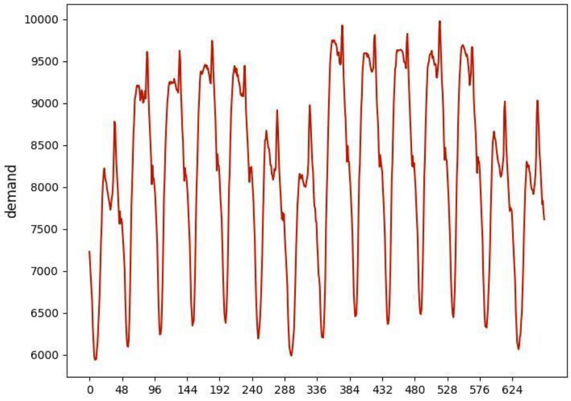


FIGURE 7. Australian two-week daily load profile.

To identify the feature variables highly correlated with the load, we visualized the temperature and load data from the Australian dataset for more than 80,000 records from January 1, 2006, to January 1, 2011, as shown in Figure 6. It is evident that the load reaches its peaks when the temperature is at its high and low points, indicating that both excessively high and low temperatures lead to increased load, So the season has an effect on the load. Additionally, we randomly visualized the load data for consecutive two weeks from, clearly showing that the load on weekends is consistently lower than that on weekdays. For the given feature parameters in the dataset, we used the Pearson correlation coefficient to examine the impact of various factors on the load, calculated according to formula 3.

$$r_{xy} = \frac{\sum_{i=1}^{n} (x_{\mathrm{i}} - \overline{x})}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2 \sum_{i=1}^{n} (y_i - \overline{y})^2}} \tag{13}$$

where $\{x_i, i = 1, 2, \cdot, n\}$ and $\{y_i, i = 1, 2, \cdot, n\}$ represent two sets of sequences of length n. By calculating the sequences of each feature variable, we obtained the heat map as shown in Figure 8.

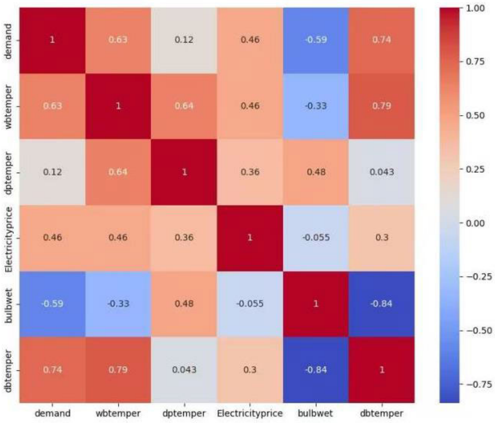


FIGURE 8. Pearson correlation heatmaps for the Australian dataset.

We selected features with absolute correlation coefficients greater than 0.5 for prediction, namely dry bulb temperature, wet bulb temperature, demand, and humidity. Additionally, we included season and weekend indicators, resulting in a ten-dimensional feature input. Season comprises four dimensions, weekend indicators are represented in two dimensions, and the rest are represented in one dimension.

C. DATA PREPROCESSING

To facilitate the training of the model network and eliminate inconsistent dimension scales, we normalized the dimension values using the following operation:

$$z' = \frac{z - z_{\min}}{z_{\max} - z_{\min}} \tag{14}$$

where z is the original data in a dimension; $z_{\min}$ is the minimum value of the dimension in the dataset; $z_{\max}$ is the maximum value of the dimension in the dataset; z' is the data after z normalization.

D. EVALUATION INDICATORS

In this paper, four evaluation metrics are used: Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) coefficient of determination (R^2). Its value can be used to determine the accuracy of model prediction [23], [24]. The expressions are given below:

$$MAPE = \frac{1}{N} \sum_{s=1}^{N} \frac{|y_s - y_{rs}|}{y_s} \times 100\% \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{s=1}^{n} (y_s - y_{rs})^2} \tag{16}$$

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=1}}^{\mathrm{N}} |\mathrm{y_{s-}y_{rs}}| \tag{17}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{\mathrm{i=1}}^{\mathrm{n}} (\mathrm{y_s} - \mathrm{y_{rs}})^2}{\sum_{\mathrm{i=1}}^{\mathrm{n}} (\mathrm{y_{rs}} - \mathrm{y_{average}})^2} \tag{18}$$

where: $\mathrm{y_s}$ is the predicted value at time s; $\mathrm{y_{rs}}$ is the actual value at time s, $\mathrm{y_{average}}$ is the average of the real values, N is the number of samples.

E. SLIDING WINDOW OPTIONS

Since the model used in this paper belongs to the sliding window iterative load forecasting model, and the size of the sliding window will have a greater impact on the model performance, so this paper for different sliding window parameters into the prediction experiments, and in the comprehensive analysis of the sliding window size optimisation and selection. Table 1 presents the error comparison of different sliding window widths for different models and datasets. The experimental results clearly indicate that the Australian dataset performs better with a sliding window width of 48, while the U.S. dataset performs better with a sliding window width of 24. The characteristics of the datasets can also provide a reasonable explanation for the optimal window width for these two datasets. The Australian dataset is sampled every 30 minutes, resulting in 48 data points per day, while the U.S. dataset is sampled every hour, resulting in 24 data points per day. Load forecasting based on daily cycles can better discover potential patterns in the sequences, and the experimental results precisely support our viewpoint.

The comparison of the data reveals that choosing the appropriate sliding window width reduces each evaluation index compared to other sliding window widths. In order to further demonstrate the difference between the load forecasting results under different sliding window widths, Fig. 9 and Fig. 10 show the visual comparison between the forecasting results and the actual values of 48 points in the test set under different sliding window widths, and it can be found that choosing the appropriate sliding window forecasting curves can better fit the real load curves.

IV. EXAMPLE ANALYSIS

In this paper, two real load datasets are used to experiment the model, which are the public dataset provided by a public utility in the United States and the public multidimensional load data of an area in Australia. The U.S. public dataset

TABLE 1. Experimental results for two datasets with different sliding windows.

Sliding window width		Australian dataset			US dataset		
Methods		12	24	48	12	24	48
LSTM	MAPE(%)	1.10	0.92	0.83	1.03	0.95	0.96
	RMSE	130.92	110.13	101.26	294.29	302.07	313.33
	MAE	95.55	82.54	75.31	153.16	142.35	139.61
GRU	MAPE(%)	1.20	1.02	0.89	1.06	0.82	0.88
	RMSE	138.14	124.31	109.05	289.90	221.73	242.22
	MAE	100.83	93.15	80.94	159.55	120.11	130.85
TimesNet	MAPE(%)	0.85	0.83	0.82	1.7	0.74	0.91
	RMSE	101.81	99.82	98.30	397.07	219.49	286.63
	MAE	75.87	73.65	72.20	259.96	111.15	131.92
TimesNet+lstm	MAPE(%)	0.80	0.78	0.67	0.71	0.65	0.62
	RMSE	95.07	97.53	79.66	219.07	209.45	198.39
	MAE	71.44	70.54	60.07	107.43	98.89	91.36
ours	MAPE(%)	0.65	0.59	0.52	0.67	0.53	0.65
	RMSE	80.04	72.78	63.40	227.65	195.13	205.34
	MAE	57.86	53.74	46.85	101.36	79.97	96.27

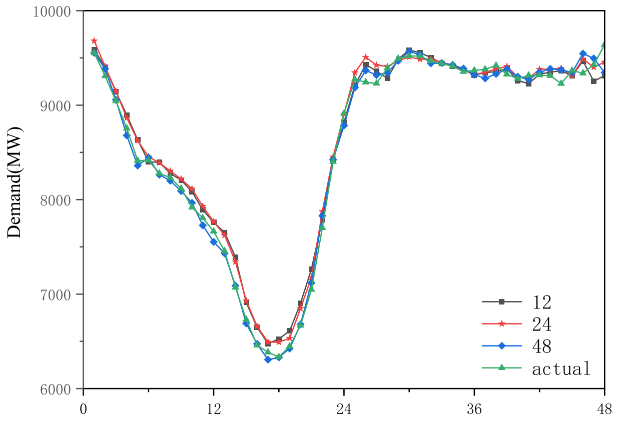


FIGURE 9. Comparison of the predictions of the proposed model across windows in the Australian dataset.

was selected from 1 June 2003 to 1 June 2008, a total of 60 months of electric load data, 24 points were collected in a day, and the time interval was 1 h. The Australian region dataset was selected from 1 January 2006 to 31 June 2008, a total of 30 months of electric load data, 48 points were collected in a day, and the time interval was 30 min. 48 points with a time interval of 30 min. In this paper, the models combining Timesnet with Crossformer and LSTM (denoted as TimesNet-Crossformer-LSTM), TimesNet, LSTM, GRU, TCN, CNN, CNN-LSTM, BiLSTM, TimesNet- LSTM and Attention mechanism of TimesNet- LSTM model are compared. To ensure the scientific validity and rigor of the proposed methods and to make the experimental results more reliable, all models were trained, validated, and tested using the same datasets. Additionally, the historical load sequence data input for each model was consistent.

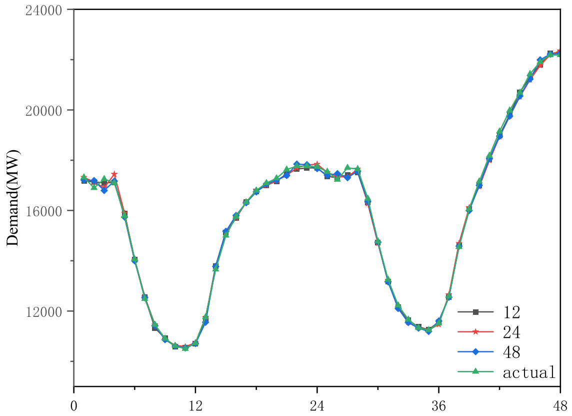


FIGURE 10. Comparison of the predictions of the proposed model across windows in the Australian dataset.

A. COMPARISON OF PREDICTED RESULTS FOR AUSTRALIAN DATASETS

This paper conducted training, validation, and prediction on the electricity load data from a specific region in Australia, utilizing the preceding 24 months as the training set, the subsequent 3 months for validation to adjust coefficients, and the final 3 months for testing to assess the accuracy of the model. The results of the quantitative evaluation of each model the final 3 months for testing to assess the accuracy

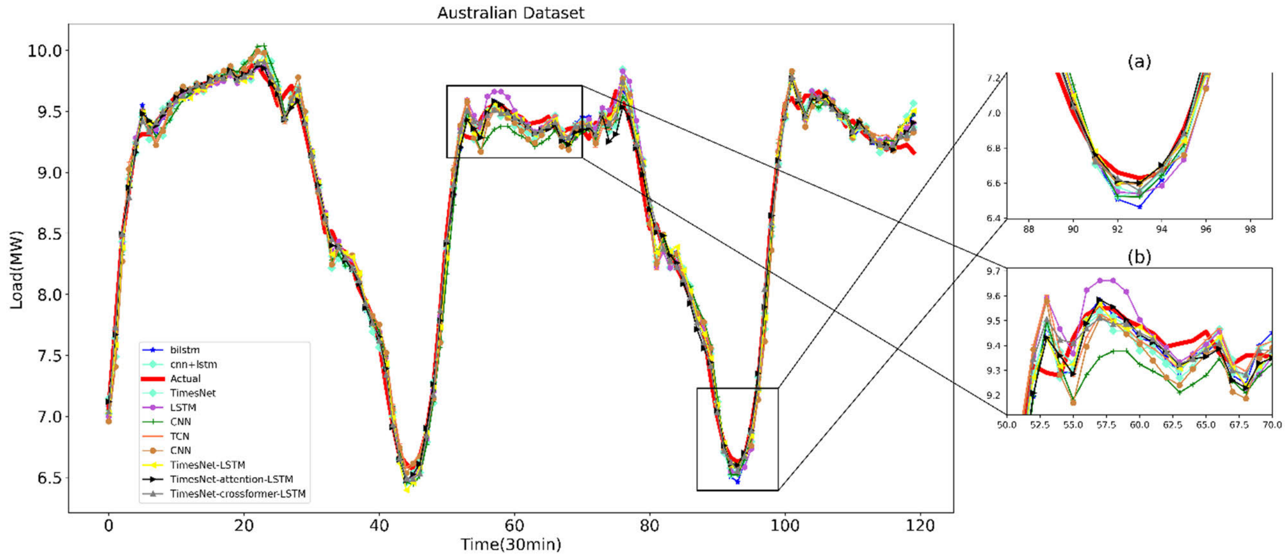


FIGURE 11. **Prediction curves for each model for the Australian dataset.**

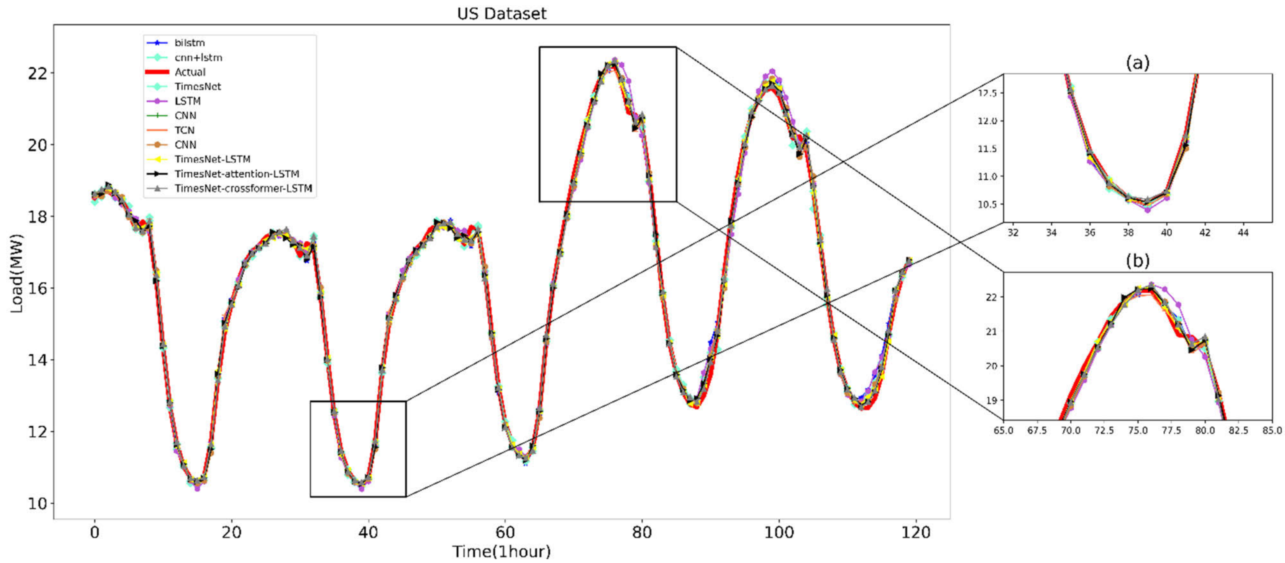


FIGURE 12. **Prediction curves for each model for the US dataset.**

of the model. The results of the quantitative evaluation of each model are shown in Table 1. From the perspective of predictive results, the model proposed in this paper exhibited higher accuracy compared to other models. Specifically, its predictive accuracy reached 99.48%. In comparison to nine other methods, the MAPE was reduced by 0.62%, 0.31%, 0.37%, 0.44%, 0.27%, 0.30%, 0.20%, 0.15%, and 0.08% respectively. Additionally, the RMSE decreased by 55.08% from the highest error, and the MAE decreased by 54.81%. The coefficient of determination also increased by 0.01067. The information conveyed by the four evaluation indicators confirmed that the proposed model in this paper exhibited improved predictive performance in both predictive capability and accuracy.

To emphasize the model's advantages, this paper visualized the real values and predicted values of the 120 points in the test set for each model. As shown in Figure 11, it is evident that the proposed model in this paper better fits the original load curve, demonstrating a significant advantage in predicting peak and valley load patterns compared to other models. To further highlight the advantages of the model, this paper visualized the real values and predicted values of the 120 points in the test set for each model, as shown in Figure 11. It is evident that the proposed model in this paper aligns more closely with the original load curve. Compared to other models, the model in this paper exhibits a significant advantage in predicting peak and valley load patterns.

TABLE 2. Load prediction evaluation on test sets.

Datasets	Australian dataset				US dataset			
Methods	MAPE(%)	RMSE	MAE	R^2	MAPE(%)	RMSE	MAE	R^2
CNN	1.14	141.14	103.66	0.98664	1.21	328.38	177.30	0.98294
LSTM	0.83	101.26	75.31	0.99301	0.95	302.07	142.35	0.98557
GRU	0.89	109.05	80.94	0.99204	0.82	221.73	120.11	0.99098
TCN	0.96	120.59	87.55	0.99025	0.77	229.99	117.29	0.99163
BiLSTM	0.79	96.02	71.09	0.99382	0.95	278.97	142.01	0.98769
TimesNet	0.82	99.34	74.20	0.99341	0.74	219.49	111.15	0.99238
CNN-LSTM	0.72	111.87	83.71	0.99161	0.97	274.32	140.85	0.98810
TimsNet-LSTM	0.67	79.66	60.07	0.99576	0.65	209.45	98.89	0.99306
TimesNet-attention-lstm	0.60	72.78	53.74	0.99644	0.58	195.76	85.74	0.99393
ours	0.52	63.40	46.85	0.99731	0.53	195.13	79.97	0.99397

TABLE 3. Multi-step load forecasting evaluation on a test set.

Methods		Australian dataset(30 minutes per point)					US dataset(60 minutes per point)				
Predicted step size		TimesNet	LSTM	GRU	TimesNet+ LSTM	Ours	TimesNet	LSTM	GRU	TimesNet+ LSTM	Ours
1point	MAPE(%)	0.82	0.83	0.89	0.67	0.52	0.74	0.95	0.88	0.65	0.53
	RMSE	98.30	101.26	109.05	79.66	63.40	219.49	302.07	242.22	209.45	195.13
	MAE	72.20	75.31	80.94	60.07	46.85	111.15	142.35	130.85	98.89	79.97
2points	MAPE(%)	1.10	0.93	1.25	0.94	0.70	0.91	1.26	1.30	0.78	0.71
	RMSE	137.05	115.49	153.35	117.71	87.87	247.99	364.29	339.18	227.71	216.38
	MAE	99.75	83.15	112.10	84.47	63.08	136.74	190.46	191.11	117.70	106.25
4points	MAPE(%)	1.33	1.16	1.70	1.25	0.91	1.37	1.95	1.67	1.29	1.23
	RMSE	168.64	149.93	220.65	159.58	114.56	326.85	478.70	381.94	314.66	304.67
	MAE	119.84	103.32	152.99	111.88	82.34	204.09	248.89	247.11	193.89	185.31
6points	MAPE(%)	1.69	1.48	1.79	1.38	1.01	1.65	2.11	1.91	1.42	1.53
	RMSE	210.42	194.42	225.93	184.12	127.65	387.56	508.71	426.09	343.53	360.27
	MAE	150.21	131.73	159.15	126.06	91.18	247.36	311.53	283.37	213.53	228.59

B. COMPARISON OF FORECAST RESULTS FROM PUBLICLY AVAILABLE US DATASETS

The paper divided the electricity load data from a specific region in the United States, published by the public sector, into three datasets according to a predetermined ratio: 16 months for the training set, 4 months for the validation set, and 4 months for the test set. Various models were then used to validate the partitioned dataset, and the quantitative evaluation results of each model are presented in Table 2. From the perspective of predictive results, the model proposed in this paper exhibited higher accuracy compared to other models, with a predictive accuracy of 99.47%. In comparison to nine other methods, the MAPE was reduced by 0.68%, 0.42%, 0.29%, 0.24%, 0.42%, 0.21%, 0.44%, 0.12%, and 0.05% respectively. Additionally, the RMSE decreased by 40.58% from the highest error, and the MAE decreased by 54.89%. The coefficient of determination also increased by 0.01103. Therefore, the proposed model in this paper demonstrated a significant improvement in accuracy across all four evaluation indicators compared to other models, showing greater advantage in short-term electricity load forecasting.

To provide a clearer demonstration of the model's advantages, this paper visualized the real values and predicted values for the 120 points in the test set for each model, as shown in Figure 12. It is evident that the proposed model in this paper exhibits higher fitting between the predicted values and real values, resulting in higher predictive accuracy and better forecasting effectiveness compared to other models.

C. COMPARISON OF PREDICTION RESULTS FOR MULTI-STEP PREDICTION

In load forecasting, multi-step prediction refers to predicting the load values for multiple future time steps. Typically, load forecasting entails predicting the load for a future time period, such as load values for the next day, week, or month. On the other hand, multi-step prediction involves simultaneously predicting multiple consecutive time steps in a single forecast, for example, predicting the load value for a future time point and multiple subsequent points in one go. Multi-step prediction contributes to a more comprehensive understanding of the load variation trends and evolving patterns over a future time horizon, and is of significant importance for power system scheduling and resource planning. To demonstrate that the proposed model in this paper exhibits good performance in multi-step prediction compared to other models, experiments were conducted to compare the prediction results for 1, 2, 4, and 6 time steps. The performance of each model in predicting at various time steps is presented in Table three.

The experimental results comparison in Table 3 reveals that the proposed model in this paper demonstrates superior performance in both single-step and multi-step prediction on two datasets, exhibiting high accuracy and strong robustness. These findings provide valuable reference for future research endeavors.

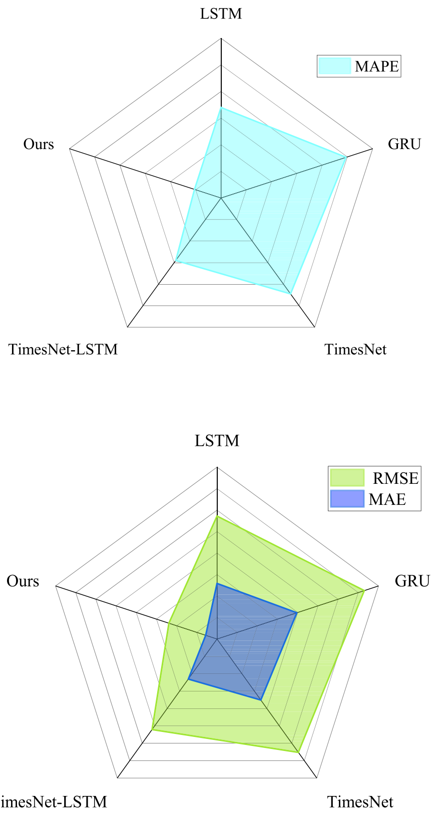


FIGURE 13. Radar chart of three indicators for multi-step forecasting (6 points).

D. ACTIVATION FUNCTION TESTING

LSTM models face a significant challenge known as the vanishing or exploding gradient problem. Generally, LSTM tend to choose the tanh activation function over the ReLU. The tanh activation function has a range of [−1, 1], which effectively normalizes input and output data, thus improving convergence to some extent. This is highly beneficial for the inputs and outputs of the LSTM, ultimately enhancing model stability and accuracy. However, whether other activation functions are more suitable for the proposed model in this paper remains unknown. Therefore, we conducted experiments to test the accuracy of each activation function. The formula for the activation functions we tested is as follows:

tanhshrink:

$$\text{tanhshrink}(x) = x - \frac{e^x - e^{-x}}{e^x - e^{-x}} \tag{19}$$

handtanh:

$$\text{handtanh}(x) = \begin{cases} 1 & \textit{while} > 1 \\ -1 & \textit{while} < -1 \\ x & \textit{otherwise} \end{cases} \tag{20}$$

silu:

$$\text{silu} = x * \frac{1}{1 + e^{-x}} \tag{21}$$

tanh:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x - e^{-x}} \tag{22}$$

gelu:

$$gelu(x) = x * P(X \leq x) = x * \Phi(x) \tag{23}$$

relu:

$$relu(x) = \begin{cases} \text{x}, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{24}$$

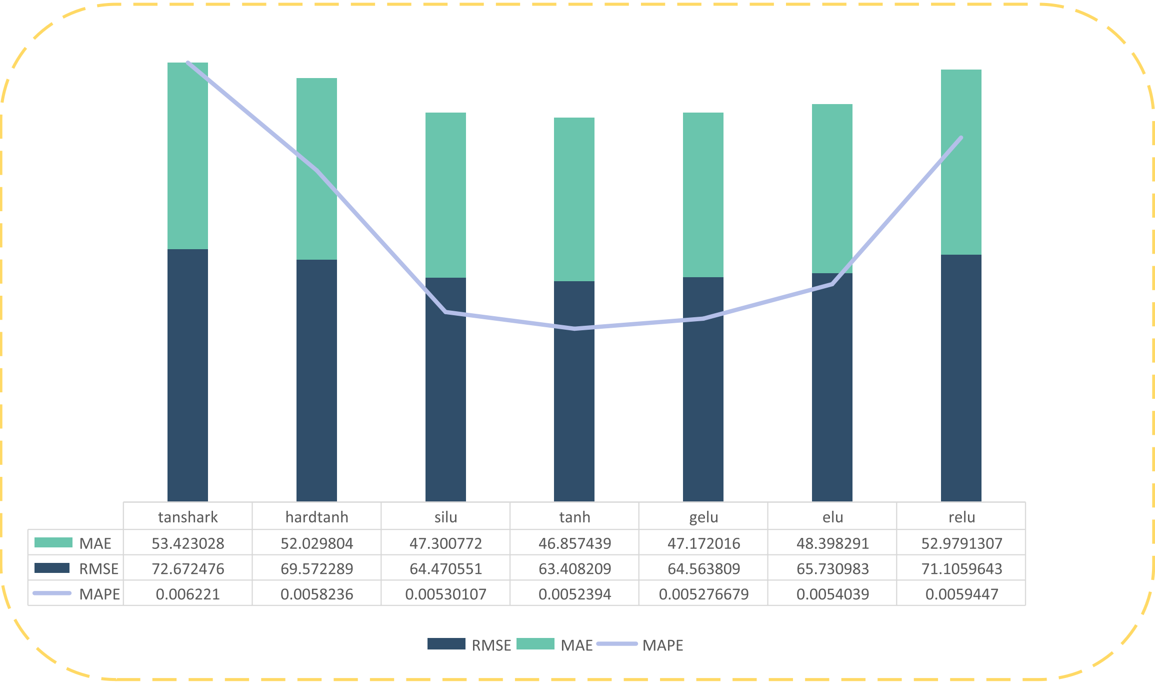
The experimental results of the above activation functions are shown in Figure 14, where it is evident that when using the tanh activation function, all evaluation metrics are lower than with other activation functions. Thus, it is proven that tanh is more suitable for the proposed model in this paper compared to other commonly used activation functions.

E. EXPERIMENTS RESULTS AND ANALYSIS

To ensure the reproducibility of the results, the sizes of each module of the proposed model and other crucial parameters are listed in Table 4. All model experiments described in this paper were carried out on a computer system equipped with an Intel Core i5 10400 CPU and an NVIDIA GeForce RTX 2060 6GB GPU. The programming environments employed in this study encompassed PyTorch 2.0.0 and CUDA 11.7. PyTorch, a widely adopted deep learning framework, offers a range of pretrained models, tools, and extensions that facilitate the development and training of deep neural network models. Furthermore, PyTorch enables efficient computation on the GPU, thereby accelerating the training process.

F. THE INFLUENCE OF MODULE PARAMETER SIZE ON EXPERIMENTAL RESULTS

To substantiate the rationale behind the chosen parameters in this study, we examined the influence of different parameters on the experimental outcomes. The impact of parameter size on iteration speed and prediction accuracy is demonstrated in Table 5. It is evident that parameter size does have a



	tanshark	hardtanh	silu	tanh	gelu	elu	relu
MAE	53.423028	52.029804	47.300772	46.857439	47.172016	48.398291	52.9791307
RMSE	72.672476	69.572289	64.470551	63.408209	64.563809	65.730983	71.1059643
MAPE	0.006221	0.0058236	0.00530107	0.0052394	0.005276679	0.0054039	0.0059447

FIGURE 14. Comparison of activation function.

TABLE 4. Description of model parameters.

Models	Parameters	Description	Value
TimesNet	T_K	Select the k elements with the largest amplitudes	2
	N_C	Number of convolutional layers	6
	D_f	dimension of function	40
Crossformer	S_l	segment length	6
	N_h	num of heads	4
LSTM	N_e	Number of hidden-state of the network.	16
	N_l	Number of layer	1
	D_p	Dropout	0.1
Proposed Model	L_r	The learning rate on training.	0.005
	I_s	Input sequence length	24,48
	B_s	Batch size	256
	A_f	Activation function of the network.	tanh
	O_{pt}	The optimizer of the network.	Adam

certain effect on these aspects. Hence, the parameter selection for each module in this study was based on a comparative analysis of ablation experiments. As depicted in Table 1 earlier, we conducted ablation experiments on various datasets

TABLE 5. Effects of different parameters on experimental results (Australian dataset).

Parameters	value	Mape(%)	one epoch need time(s)
N_l	1	0.52	5.23
	2	0.54	5.6
L_r	0.005	0.56	5.23
	0.01	0.52	5.21
S_l	4	0.60	5.09
	6	0.52	5.23
	8	0.55	5.42

and determined the sliding windows that were most suitable for each dataset.

G. THE APPLICATION AND DEFICIENCY OF THE MODEL

This paper presents a new model which can be used for short-term load forecasting, and provides a high precision short-term load forecasting method for reference. The method is applied to collect the historical data of the prediction quantity and the historical data of the features related to the prediction quantity, and then through deep learning, the hidden information between the features is mined, so as to predict the future results through the historical data. However, with the addition of model stacking, activation layer, linear layer, etc., the parameters increase, and more computing power and time are needed to obtain the predicted results. Therefore, although the proposed model has higher accuracy than the traditional model and other hybrid models, it is also time-consuming and labor-intensive. Therefore, in the future, we need to reduce the complexity of the model while ensuring the accuracy of the model, and try to achieve high precision and low cost.

V. CONCLUSION

In this study, a novel ensemble model based on TimesNet-Crossformer-LSTM is proposed for short-term electricity load forecasting. The method employs TimesNet to transform the one-dimensional time series into two-dimensional space for information extraction before converting it back to one-dimensional form. The Crossformer module is then used to capture relationships between the time and variable dimensions of the time series through two-stage attention mechanisms. Finally, the extracted features are dynamically modeled using LSTM to obtain load prediction results and accomplish the forecasting task. Experimental comparisons are conducted with TimesNet, LSTM, GRU, CNN, CNN-LSTM, TCN, TimesNet-LSTM, BiLSTM, and TimesNet-attention-LSTM.The results demonstrate that the proposed model performs better and achieves higher accuracy on two datasets, showcasing its significant potential. Sensitivity analysis is also conducted to validate the appropriateness of the selected model parameters. Overall, this study provides a new and viable model for load forecasting in future research. The transformation of one-dimensional time series into two-dimensional space enables better extraction of deeper hidden information. The two-stage attention mechanism considers the relationships among different feature dimensions, offering further insights for future time series prediction.

REFERENCES

[1] S. Qiu, T. Lei, J. Wu, and S. Bi, "Energy demand and supply planning of China through 2060," *Energy*, vol. 234, Nov. 2021, Art. no. 121193.

[2] A. S. Khwaja, X. Zhang, A. Anpalagan, and B. Venkatesh, "Boosted neural networks for improved short-term electric load forecasting," *Electr. Power Syst. Res.*, vol. 143, pp. 431–437, Feb. 2017.

[3] D. Deng, J. Li, and Z. Zhang, "Short-term electric load forecasting based on EEMD-GRU-MLR," *Power Syst. Technol.*, vol. 44, no. 2, pp. 593–602, 2020.

[4] Z. Hu, J. Hu, and H. Peng, "Short-term customer load forecasting based on deep random forest algorithm—An example from Jinhua area," *J. Univ. Electron. Sci. Technol.*, vol. 52, no. 3, pp. 430–437, 2023.

[5] Y. Chang, H. Sun, and T. Gu, "Monthly forecast of wind power generation using historical data expansion method," *Power Syst. Technol.*, vol. 45, no. 3, pp. 1059–1068, 2021.

[6] X. Bai, X. Zhao, and Z. Jiang, "Spatial load forecasting method using fuzzy information granulation support vector machine," *Power Syst. Technol.*, vol. 45, no. 1, pp. 251–260, 2021.

[7] B. Deng, N. Zhang, and J. Wang, "Medium and long-term powerload forecasting method based on LTC-RNN model," *J. Tianjin Univ. (Natural Sci. Eng. Technol. Ed.)*, vol. 55, no. 10, pp. 1026–1033, 2022.

[8] X. Li, L. Ma, and X. Zhao, ""Multi-time scale electric heating load forecasting based on long short-term memory network," *Proc. CSU-EPSA*, vol. 33, no. 4, pp. 71–75, 2021.

[9] K. Li, W. Huang, G. Hu, and J. Li, "Ultra-short term power load forecasting based on CEEMDAN-SE and LSTM neural network," *Energy Buildings*, vol. 279, Jan. 2023, Art. no. 112666.

[10] Y. Cui, H. Zhu, Y. Wang, L. Zhang, and Y. Li, "A short-term power load forecasting method based on CNN-SAEDN-Res," *Power Automat. Equip.*, vol. 44, no. 4, pp. 164–170, 2024, doi: 10.16081/j.epae.202308018.

[11] Y. Li, X. Liu, and F. Xing, "Daily peak load forecasting based on Bi-LSTM and feature correlation analysis," *Grid Technol.*, vol. 45, no. 7, pp. 2719–2730, 2021.

[12] Y. Wang, Y. Shi, and Z. Xu, "Ultra-short-term power prediction of multi-wind turbines by BiLSTM based on temporal pattern attention mechanism," *High Voltage Technol.*, vol. 48, no. 5, pp. 1884–1892, 2022.

[13] J. Ren, H. Bit, and Z. Zou, "Ultra-short-term power load forecasting based on CNN-BiLSTM-Attention," *Power Syst. Protection Control*, vol. 50, no. 8, pp. 108–116, 2022.

[14] Z. Zou, T. Wu, and X. Zhang, "Short-term load forecasting based on Bayesian optimization CNN-BiGRU hybrid neural network," *High Voltage Technol.*, vol. 48, no. 10, pp. 3935–3945, 2022.

[15] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "TimesNet: Temporal 2D-variation modeling for general time series analysis," in *Proc. ICLR*, 2023, pp. 1–23.

[16] S. Zhang, W. Liu, and H. Tang, "A short-term load forecasting method based on transformer multi-feature fusion," *J. North China Electr. Power Univ. (Natural Sci. Ed.)*, pp. 1–9, 2023. Accessed: Apr. 22, 2024. [Online]. Available: http://kns.cnki.net/kcms/detail13.1212.TM.20230607.0918.002.html

[17] P. Ran, K. Dong, X. Liu, and J. Wang, "Short-term load forecasting based on CEEMDAN and transformer," *Electr. Power Syst. Res.*, vol. 214, Jan. 2023, Art. no. 108885.

[18] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *Proc. ICLR*, 2023, pp. 1–21.

[19] C. Teng, Y. Ding, and Y. Zhang, "Ultra-s-hort-term photovoltaic power prediction based on VMD-informer-BiLSTM model," *High Voltage Technol.*, vol. 49, no. 7, pp. 2961–2971, 2023.

[20] L. Zhu, Z. Xun, and W. Yuxin, "Short-term power load forecasting based on CNN-Bi LSTM," *Grid Technol.*, vol. 45, no. 11, pp. 4532–4539, 2021.

[21] Z. Bing, Z. Wang, and J. Weijia, "CNN-GRU short-term power load forecasting method based on attention mechanism," *Grid Technol.*, vol. 43, no. 12, pp. 4370–4376, 2019.

[22] S. Zhang, L. Du, and Z. Wang, "A wind power prediction method based on Gram angle field and improved CNN-ResNet," *Grid Technol.*, vol. 47, no. 4, pp. 1540–1548, 2023.

[23] S. Chao, L. Qi, and S. Zhu, "Ultra-short-term power load forecasting based on two-layer XGBoost algorithm considering the influence of multiple features," *High Voltage Technol.*, vol. 47, no. 8, pp. 2885–2898, 2021.

[24] J. Zhang, W. Siya, T. Zhongfu, and S. Anli, "An improved hybrid model for short term power load prediction," *Energy*, vol. 268, Apr. 2023, Art. no. 126561.

[25] N. Mounir, H. Ouadi, and I. Jrhilifa, "Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system," *Energy Buildings*, vol. 288, Jun. 2023, Art. no. 113022.

[26] L. Xiao, W. Shao, M. Yu, J. Ma, and C. Jin, "Research and application of a combined model based on multi-objective optimization for electrical load forecasting," *Energy*, vol. 119, pp. 1057–1074, Jan. 2017.

[27] Y. Zhang and J. Zhang, "Volatility forecasting of crude oil market: A new hybrid method," *J. Forecasting*, vol. 37, no. 8, pp. 781–789, Dec. 2018.

[28] J. Zhang, D. Li, Y. Hao, and Z. Tan, "A hybrid model using signal processing technology, econometric models and neural network for carbon spot price forecasting," *J. Cleaner Prod.*, vol. 204, pp. 958–964, Dec. 2018.

JUN HE received the B.Sc., M.Sc., and Ph.D. degrees from Nanchang University, Nanchang, China. He is currently a Professor with the School of Information Engineering, Nanchang University. His main research interests include power system analysis, wind power forecasting, and deep learning.

KUIDONG YUAN received the B.Sc. degree from Xinyu University, Xinyu, China, in 2022. He is currently pursuing the M.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China. His research interests include power system analysis, power load forecasting, and wind power forecasting.

ZIJIE ZHONG received the B.Sc. degree from Xinyu University, Xinyu, China, in 2021. He is currently pursuing the M.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China. He focuses on power system analysis, wind power forecasting, and deep learning.

YIFAN SUN received the B.Sc. degree from Shanxi University, Shanxi, China, in 2022. He is currently pursuing the M.Sc. degree with the School of Information Engineering, Nanchang University, Nanchang, China. He focuses on power system analysis, photovoltaic forecasting, and deep learning.