A combined prediction model with multi-module integration for short-term power load forecasting

Yixiang Ding *

School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, China, 212013

* Corresponding Author Email: 3220613043.ujs@vip.163.com

Abstract. The existing power load forecasting algorithms are constrained by preprocessing limitations and insufficient prediction accuracy. Temporal Convolutional Network (TCN), Bi-directional LSTM (BiLSTM), and Multi-Head Attention (MHA) for high-precision, real-time power load prediction were used in this paper to propose a hybrid model, which combined Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (ICEEMDAN) to address these issues. The ICEEMDAN algorithm is initially employed for multi-layer decomposition to enhance data smoothness, thereby enabling the subsequent model to more effectively capture essential features. To overcome the BiLSTM's inability to capture long-term dependencies in extended sequences, this paper incorporates the TCN module and MH-Attention mechanism. The TCN improves local feature extraction through convolution, while the MH-Attention mechanism enables the model to focus on the most critical features for load prediction, thereby enhancing learning efficiency. The model is validated using an actual power plant in Quanzhou City, China, with a dataset of power load data obtained from real measurements. Experimental results demonstrate an R2 of 0.99802, RMSE of 0.00996, and MAE of 0.00694, showcasing exceptional forecasting accuracy. Compared to alternative models, the R2 improves by 1.5%-3.4%, RMSE decreases by 56.2%-80.6%, and MAE is reduced by 58.3%-84.1%. These results validate the model's superiority. The proposed combinatorial forecasting model framework effectively integrates the advantages of data decomposition, convolution, and attention mechanisms, allowing for an in-depth exploration of temporal features and critical patterns in power load data.

Keywords: Real-time power load forecasting, BiLSTM, Multiple Attention Mechanisms, Temporal Convolutional Network.

1. Introduction

For the purpose of guaranteeing the stability and efficiency of power grids [1], power load forecasting is necessary. With advancements in smart grid technology and the integration of renewable energy, developing effective forecasting algorithms is crucial for better grid management [2]. However, current models face challenges that limit their forecasting performance. Limited datasets and inadequate preprocessing methods hinder these models from fully utilizing the data's potential [3]. Additionally, their prediction accuracy is often low when handling complex power load data, failing to meet expectations [1].

Electric load data is characterized by periodicity, seasonality, and randomness, which complicate prediction due to significant uncertainty [4]. To mitigate the impact of noise and other factors, power load data must be preprocessed or decomposed, providing a robust dataset for feature extraction [5]. Raj et al. [6] applied wavelet denoising to smooth the data, improving prediction accuracy, though it struggled with large fluctuations. Peterson et al. [7] used filtering techniques to reduce outliers, but faced similar challenges with large fluctuations. Wang et al. [8] employed variational modal decomposition (VMD) to decompose complex sequences, improving short-term predictions, but its performance was sensitive to decomposition parameters. Ntunka et al. [9] utilized empirical modal decomposition (EMD) for multi-scale decomposition, capturing features from different frequency bands and enhancing accuracy, though it incurred high computational costs with large datasets. Hu et al. [10] introduced Collective Empirical Mode Decomposition (CEEMDAN), addressing mode mixing and noise through multi-layer decomposition, but it still faced convergence issues and high

complexity. These challenges highlight the need for more efficient data processing techniques to ensure high accuracy and robustness in power load forecasting.

The general application of power load forecasting methods is mainly in the field of deep learning, machine learning and statistical methods. In the statistical domain, Sanjeev et al. [11] used the ARIMA algorithm for preprocessing, enhancing robustness by reducing noise but struggling with non-stationary data. Zhu et al. [12] applied grey correlation analysis for feature extraction and dimensionality reduction, which improved model applicability but was sensitive to parameters. In the field of nonlinear modeling, feature extraction, and scalability, machine learning approaches have the following advantages: Siti et al. [13] employed support vector machines (SVMs) for feature extraction and training, improving short-term accuracy, though computational overhead was high with high-dimensional data. Wang et al. [14] used ensemble learning to enhance accuracy, but hyperparameter tuning challenges limited its potential. Feng et al. [15] combined random forests with principal component analysis to reduce feature redundancy but struggled with outlier detection. Deep learning methods, with multi-layer networks, excel in feature learning and time-series modeling compared to traditional machine learning methods. Zhang et al. [16] used BP neural networks for time-series regression, improving fitting capabilities but facing gradient vanishing issues with long-term dependencies. Yang et al. [17] applied LSTM networks to model seasonality and trends, achieving high accuracy at the cost of significant time overhead. Han et al. [18] introduced BiLSTM networks to reduce information loss via bidirectional feature capture, but the model's complexity and high computational demand remain challenges. These limitations underscore the need for more efficient data processing and modeling techniques for real-time, high-accuracy power load forecasting.

To address the challenges mentioned, this paper proposes a hybrid model combining ICEEMDAN, TCN, BiLSTM, and MH-Attention for accurate, real-time power load prediction. The ICEEMDAN algorithm is used for multi-layer decomposition to improve data smoothness, aiding the model in better capturing key features. To overcome BiLSTM's limitations in capturing long-term dependencies, we incorporate the TCN module and Attention mechanism. The TCN improves local feature extraction via convolution, while Attention helps the model focus on the most relevant features for prediction, enhancing learning efficiency. In order to guarantee great accuracy and real-time forecasting performance, this hybrid model combines information decomposition, convolutional feature extraction, and attention mechanisms in order to make a complete analysis of temporal features and important patterns in power load data.

2. BiLSTM

LSTM, or long-term short-term memory [17], is intended to solve the gradient vanishing problem that is present in the processing of lengthy sequences by conventional recurrent neural networks (RNNs). In order to capture long-term dependencies and enhance sequencing, LSTM increases RNNs by adding a gating mechanism. The structure of LSTM is shown in Figure 1.

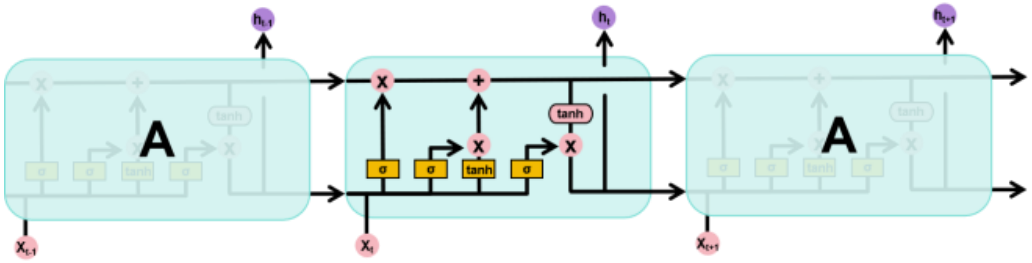

Figure 1. LSTM structure picture

The input vector X_t, which corresponds to the data at the current time step, is depicted in Figure 1. h_{t-1} is the output of the LSTM unit at that point, and h_t is the hidden state at the current time step, which indicates the output of the LSTM unit; and h_{t-1} is the hidden state from the previous time step. Tanh

and the sigmoid are respectively referred to as the symbols of sigmoid; Tanh and tanh are respectively called the sigmoid and hyperbolic tangent activation functions respectively. The sigmoid function maps inputs to the range [0,1], typically used for controlling gate mechanisms, while the tanh function maps inputs to [-1,1], commonly used to update and generate new states. Additionally, C_{t-1} and C_t represent the cellular states at the previous and current time steps.

The LSTM network consists of three main components: the input gate, the forget gate, and the output gate. The input gate controls which parts of the current input information (X_t) should be stored in the cell state, determining how much the current input affects the cell state. Its formula is:

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

$$C_t = \tanh(Wc \cdot [h_{t-1}, x_t] + b_C) \tag{2}$$

Where is the output of the input gate and is the state of the candidate cell at the current moment. i_t C_t

The forget gate determines which information from the preceding cell state (C_{t-1}) should be deleted, influencing the influence of the past state on the current cell state. Its formula is:

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \tag{3}$$

Where f_t is the output of the forgetting gate, σ is the sigmoid activation function, and W_f and b_f are the weight and bias terms.

The output gate, responsible for deciding which elements of the current cell state (C_t) should be transmitted to the hidden state (h_t), essentially determines the output of the current cell. Its formula is

$$o_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \tag{4}$$

$$h_t = o_t \cdot \tanh(c_t) \tag{5}$$

Bi-LSTM (Bidirectional Long Short-Term Memory) [18], introduced by Schuster Tal in 1997, is an extension of LSTM designed to overcome the gradient vanishing problem and handle long-term dependencies in time series data, which traditional RNNs struggle with. BiLSTM is ideal for tasks with complex temporal features, as its bidirectional structure allows the model to learn data in both forward and backward directions. This dual learning approach helps capture both past and future correlations within the sequence. The structure of BiLSTM is shown in Figure 2.

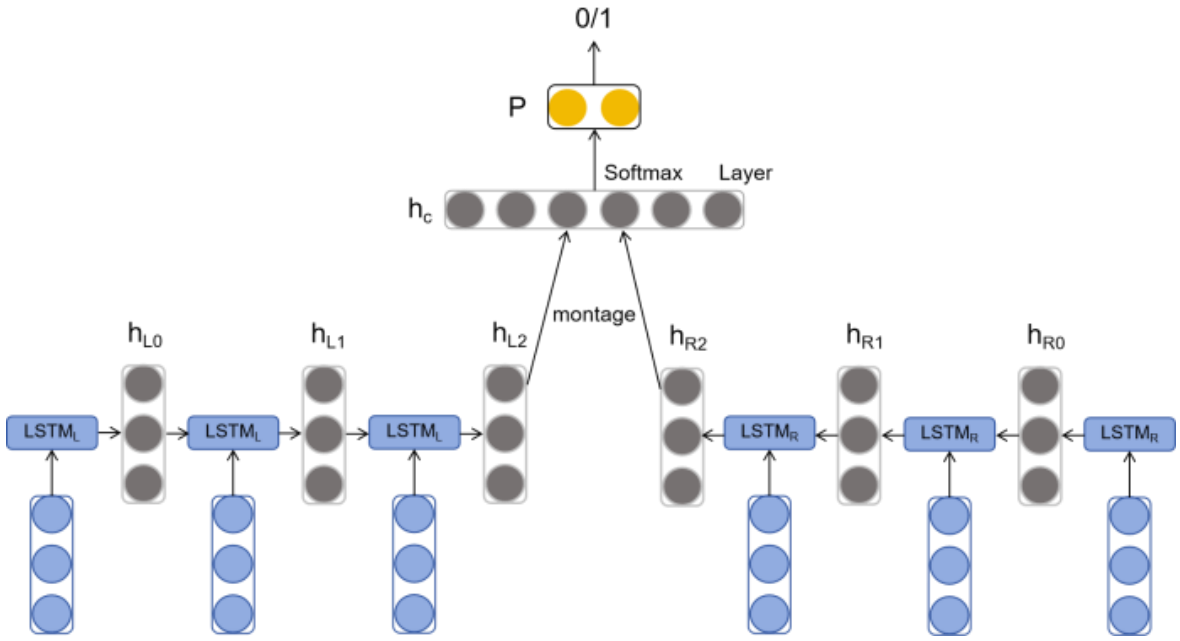

Figure 2. BiLSTM structure picture

Figure 2 illustrates that the BiLSTM network consists of two components the forward LSTM unit and the reverse LSTM unit. From start to finish, the input data from the starting point of the forward LSTM unit process is:

$$h_t^{forward} = LSTM(x_t) \tag{6}$$

Where $h_t^{forward}$, the hidden state of the forward LSTM, can be found at time step t, with x_t representing the input data at the current time step t.

The reverse LSTM unit, which reverses the processing from the end point of the data, is given by

$$h_t^{backward} = LSTM(x_t) \tag{7}$$

At time step t, $h_t^{backward}$, the hidden state of the inverse LSTM, can be found, with x_t representing the input data for the inverse LSTM.

By combining these two components, BiLSTM captures contextual information from both past and future data points, overcoming the issue of limited long-term dependency capture in unidirectional LSTMs. Each LSTM unit includes input, forget, and output gates, which allow it to selectively retain and update information, effectively preventing the gradient vanishing problem common in traditional RNNs.

While BiLSTM performs well in time series prediction, it has certain limitations. First, it struggles with capturing long-term dependencies; despite utilizing both forward and backward information, BiLSTM still experiences information loss in long sequences [18]. Second, it lacks the ability to focus on critical moments, potentially overlooking important details in lengthy data. Third, BiLSTM is less effective with non-smooth data, such as complex and non-stationary electricity load data that exhibits cyclical and seasonal variations. BiLSTM's direct learning from noisy, irregular input data makes it less suitable for such cases [5].

3. BiLSTM-based multi-module combinatorial modelling

In this paper, we present a hybrid model, ICEEMDAN-TCN-BiLSTM-MH-Attention, for high-precision, real-time power load forecasting. To address the complex characteristics of power load data, such as non-linearity and irregularity, we apply the ICEEMDAN algorithm for multi-layer decomposition, improving data smoothness and enabling better feature extraction. Additionally, to overcome the BiLSTM's limitations in capturing long-term dependencies in time series, we incorporate the TCN module and MH-Attention mechanism. The TCN module enhances the model's ability to capture local features via convolution, while MH-Attention focuses on key moments in the sequence, improving the model's ability to learn crucial patterns. Figure 3 displays the suggested model's structure.

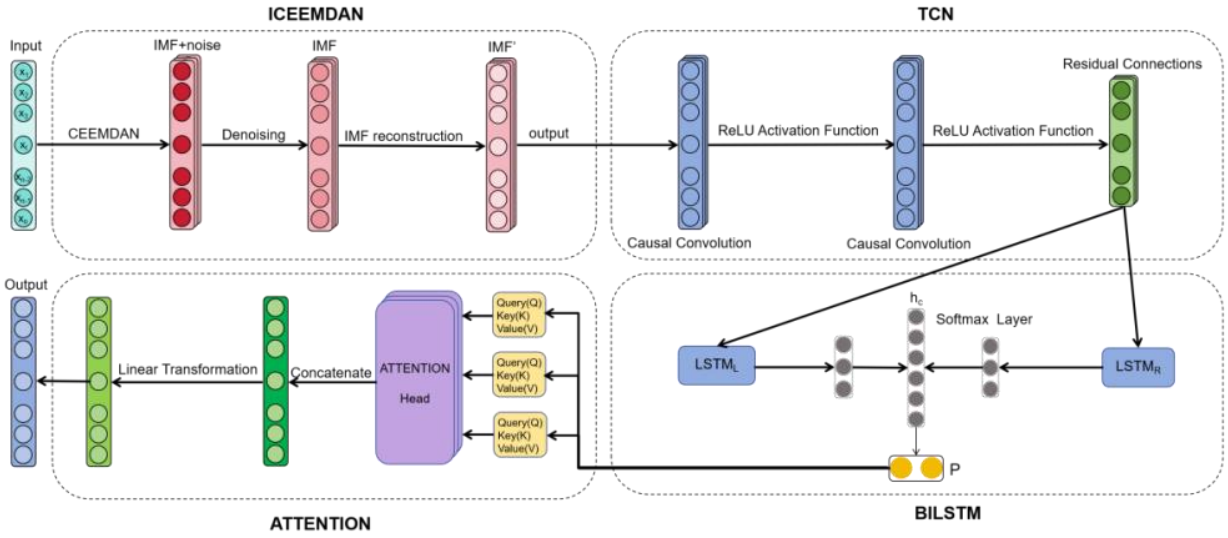

Figure 3. ICEEMDAN-TCN-BiLSTM-MH-Attention structure picture

As illustrated in Figure 3, the model input sequentially traverses several modules. Firstly, the ICEEMDAN module eliminates noise and decomposes the data, enhancing both the data's smoothness and the model's robustness. Following the ICEEMDAN decomposition, we derive a series of intrinsic modal functions (IMFs) that individually capture various frequency components of the original data. In order to obtain the temporal characteristics of these features and to use them to the BiLSTM in order to further capture the forward and backward temporal dependencies, the features represented by each IMF are individually fed into the TCN for processing. Finally, weighted by the multi-attention mechanism, the model is able to automatically focus on the most critical moment information in the sequence, thus generating more accurate power load forecasts.

3.1. Iceemdan

Ensemble Empirical Mode Decomposition with Noise Removal (ICEEMDAN) [19] is a signal processing technique aimed at eliminating noise from time-series data and extracting meaningful features. By breaking the input signal into several Intrinsic Mode Functions (IMFs), each of which represents a different frequency band, it fosters the combination of EMD and ensemble noise removal approaches. This method is very good for the time-series data, and is also very suitable for noise removal and feature extraction because it can deal with the non-stationary and non-linear signals. The structure of ICEEMDAN is shown in Figure 4.

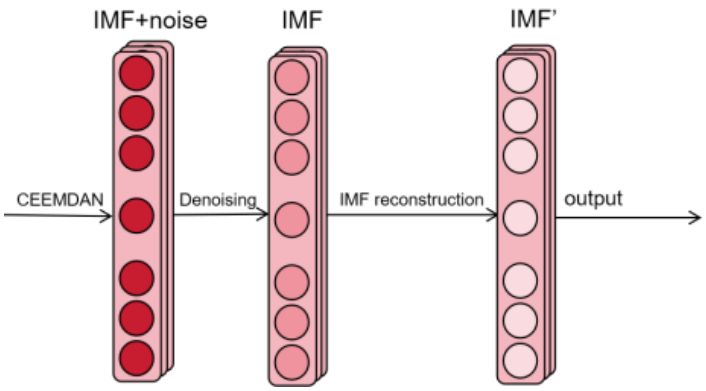

Figure 4. ICEEMDAN structure picture

The ICEEMDAN decomposition steps are as follows: Add white noise *n(t)* to the initial signal *x(t)*

$$x_{noisy}(t) = x(t) + n(t) \tag{8}$$

Where white noise $n(t)$ has an expected value of 0 and constant variance, the decomposition $x_{noisy}(t)$ is performed using EMD to extract multiple intrinsic mode functions and residual components.

$$x_{noisy}(t) = \sum_{i=1}^{m} IMF_i(t) + r(t) \tag{9}$$

Where $IMF_i(t)$ is the ith intrinsic modal function and $r(t)$ is the residual term. EEMD was used in multiple decompositions, each of which included a different noise sequence

$$x_{noisy}^{k}(t) = x(t) + n_k(t) \tag{10}$$

Where $n_k(t)$ is the k-th added white noise. The modes obtained from multiple decompositions are averaged and the noise is removed to obtain the final modal function

$$IMF_i(t) = \frac{1}{K} \sum_{k=1}^{K} IMF_i^{\,k}(t) \tag{11}$$

Where $IMF_i^{\,k}(t)$ is the i-th IMF derived from the k-th decomposition; K is the number of decompositions.

3.2. TCN

Temporal Convolutional Network (TCN) [19] addresses efficiency issues and long-term dependency modeling challenges in traditional RNNs for long sequence data. TCN is a CNN-based framework that uses Causal and Dilated Convolutions to capture long-distance dependencies without compromising efficiency. Compared to LSTM, TCN is more parallelizable, computationally efficient, and processes temporal data globally. The structure of TCN is shown in Figure 5.

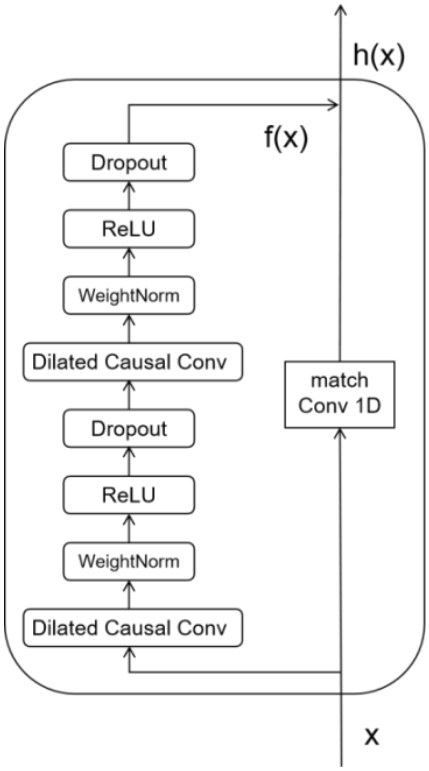

Figure 5. TCN structure picture

In Figure 5, Causal Convolution ensures that the output of the network depends only on current and past inputs, not future inputs. For the input sequence xt and the convolution kernel w, the output of Causal Convolution y_t:

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-i} \tag{12}$$

where k is the size of the convolution kernel and x_{t-i} is the input at moment t and the past k-1 values.

Dilated Convolution increases the receptive field by inserting gaps (i.e., dilation) between convolutional kernels, thus allowing each convolutional layer to capture longer time scale dependencies simultaneously. It is defined as:

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-r \cdot i} \tag{13}$$

Where r is the dilation factor indicating the spacing between elements in the convolutional kernel.

TCNs usually consist of multiple convolutional layers stacked on top of each other, and the output of each layer will depend on the result of the previous layer. for the input data x_t, the output from the first convolutional layer is h_1.

$$h_1 = Conv1D(x_t) \tag{14}$$

Then h_1 is passed as input to the next layer of convolution to get the output h_2:

$$h_2 = Conv1D(h_1) \tag{15}$$

The data are processed by each of the following layers using the methods of dilation and causal convolution: causal convolution. Finally, the output y_t of the TCN is a combination of the outputs of all the convolutional layers, which can be nonlinearly transformed by an activation function.

3.3. MH-Attention

The Multi-Head Attention (MHA) mechanism [20] addresses computational inefficiencies and limited long-term dependency capture in traditional sequence models (e.g., RNN, LSTM). It allows the model to focus on relevant information at different positions by assigning dynamic weights to all positions in the input sequence. HÀ uses the same number of attention heads in parallel to calculate the number of each, which can improve the learning ability of the model for complicated pattern recognition.

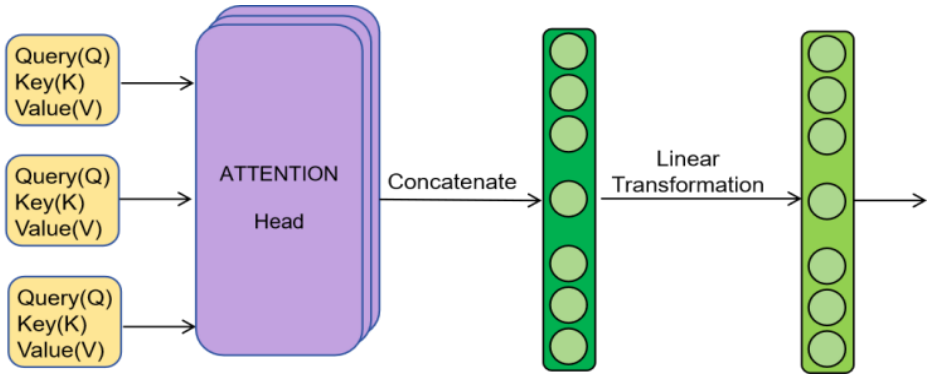

Figure 6. MH-Attention structure picture

In Figure 6, the attention mechanism functions in the following way the input sequence is transformed into query, key, and value vectors. Each query vector performs a dot product with all key vectors to generate a correlation score, which is then processed by Softmax to determine attention weights. These weights are used to combine the value vectors, and the results from multiple heads are concatenated and linearly transformed to produce the final output. Given an input sequence $X=[x_1, x_2, ..., x_n]$, where x_i is the i-th element, the Query, Key, and Value vectors of each element are first

computed, followed by attention weight calculation, and finally, the data is concatenated to give the final multi-head attention output. The specific processes are as follows:

Calculate the Key and Value vectors of the Query: The input data is converted linearly with the following formula and various weight matrices.

$$Q = XW_Q \tag{16}$$

$$K = XW_K \tag{17}$$

$$V = XW_V \tag{18}$$

Where W_Q, W_K and W_V are the learned weight matrices.

Calculation of Attention Weights: for each query, the relevance is measured by calculating the dot product of the query and the keys, scaled, and then the normalized weights are obtained by SoftMax. The formula is:

$$Attention(Q,K,V) = soft\max(\frac{QK^T}{\sqrt{d_k}})V \tag{19}$$

Where $\frac{QK^T}{\sqrt{d_k}}$ is the dot product of the query and the key, d_k is the dimension of the key, and the SoftMax operation ensures that all weights sum to 1.

Multi-head attention: by computing multiple attention heads in parallel, the model learns different subspace features. Each head has its own query, key, and value, with outputs calculated independently. To get the finished product, these outputs are linearly converted and concatenated. The following is the formula:

$$\text{MultiHead}(Q,K,V) = \text{Con}cat(head_1, head_2, ..., head_h)Wo \tag{20}$$

The output weight matrix Wo is derived from the learning process, with h representing the number of attention heads.

4. Experiments and analyses

4.1. Simulation environment

1) Dataset

In this paper, the power load dataset is sourced from a power plant in Quanzhou, Southern China, covering the period from 2016-1-1 0:00:00 to 2016-1-31 23:45:00. The data, collected on-site, consists of a total of 2976 samples with a sampling frequency of 15 minutes per interval. Comprise the dataset, which consists of a test set (20%) and a validation set (10%) as well as a training set. The input variables are related to the electricity consumption of the power load.

2) Experimental environment

Table 1 displays the computer environment that was used in this paper.

Table 1. Experimental environment

Categories	Parametric
CPU	Intel (R) Core(TM) i9-14900HX 2.20 GHz
Graphics Board	NVIDIA GeForce RTX 4070 Laptop GPU 8GB
RAM	32GB
Code Language	Python 3.8
Software	Pycharm 2024.2.1

3) Parameterisation

Table 2 displays the model parameters that were used in this paper.

Table 2. Model-related parameters

Model parameter name	Value
Num_imfs	9
Time_step	30
Filters	64
Kernel_size	3
Batch_size	64
Epochs	500
Learning_rate	0.001

4) Assessment of indicators

Root Mean Square Error (RMSE) and the coefficient of determination (R) are two popular short-term power load forecasting tools for assessing the accuracy of the model: the Root Mean Square Error (RMSE), the mean absolute error (MAE), and the Coefficient of Determination (R). RMSE and MAE are key metrics for assessing forecasting accuracy; the closer their values are to 0, the better the model's accuracy and the smaller the error indicates how well the model fits the data; the closer its value is to 1, the better the fit. These parameters were calculated using the following formulae:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{21}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|(y_i - \hat{y}_i)\right| \tag{22}$$

$$R^2 = 1 - \frac{\sum_{i}^{n}(\hat{y}_i - y_i)}{\sum_{i}^{n}(\overline{y_i} - y_i)} \tag{23}$$

Where n is the total number of test samples; y_i, the value of the ith sample point, is indicated by the true value; $\hat{y}_i$, the average value of ith sample point, is shown by the value; and $\overline{y_i}$, the average of ith sampling point, is indicated by the symbol.

4.2. Model validation

In order to verify the validity and applicability of the proposed model, we collated the trend of the Loss function during the model training process and recorded it in Figure 7.

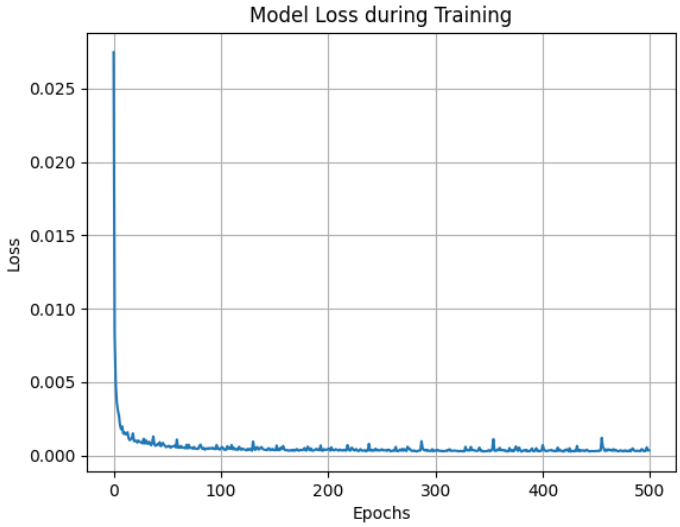

Figure 7. Loss fold plot for combinatorial model training

Figure 7 illustrates that the loss curve consistently declines with each increasing epoch. There is a significant decline in the early stages, followed by a slower decrease around epoch 50, and the curve stabilizes near epoch 100, approaching 0. It is shown that there is good model convergence, and there are no obvious indications of overfitting or underfitting. As the model determines the best parameters to achieve, it can be used to decrease errors and achieve better results, which makes the training process work well.

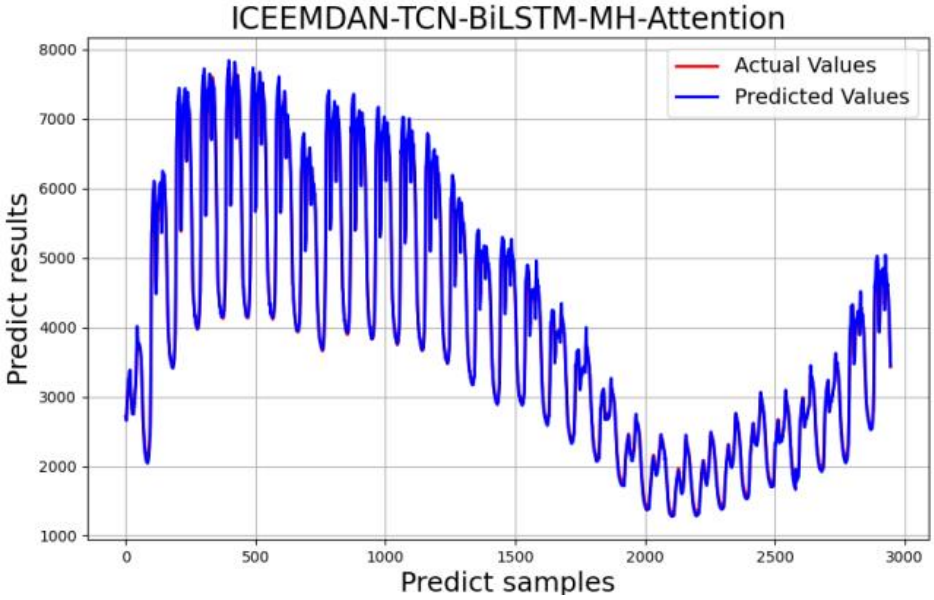

Figure 8. Comparison of combined model predictions and true values

The red curve in Figure 8 represents the actual values, whereas the blue curve shows the ICEEMDAN-TCN-BiLSTM-MH-Attention model forecasts. The predicted values closely match the true values, indicating a high degree of model fit. This suggests that the model accurately captures the data's characteristics, demonstrating strong accuracy, generalization performance, and effective load prediction capability.

Table 3. Indicators for model evaluation

Model	RMSE	MAE	R^2
Train	0.00996	0.00694	0.99802
Test	0.00992	0.00689	0.99768

The ICEEMDAN-TCN-BiLSTM-MH-Attention combined model evaluation metrics are shown in Table III. In terms of training, the RMSE metric of 0.00996 in the training set and 0.00999 in the test set, the MAE metric of 0.00689 in the training set and 0.99802 in the test set; and the R^2 metric of 0.99768 in the training set and 0.00992 in the test set. This can reflect that the model has a good metrics evaluation result and the model performance is good.

In summary, the Loss curve of the model converges and tends to be close to 0, and the true value and the predicted value are very close to each other, and at the same time, it has good evaluation indexes on both the training set and the test set, thus verifying the validity and applicability of the model.

4.3. Ablation experiment

In order to evaluate the contribution effect of individual modules in the ICEEMDAN-TCN-BiLSTM-MH-Attention combination model, this paper will compare the performance of different combination modules (BILSTM, ICEEMDAN-BiLSTM, ICEEMDAN-BiLSTM-MH-Attention, ICEEMDAN-TCN- BiLSTM-MH-Attention) on the same dataset with RMSE, MAE and R^2 as evaluation metrics. The experimental results are shown in Table 4.

Table 4. Comparison of different model indicators

Model	RMSE	MAE	R^2
BiLSTM	0.05183	0.04398	0.96435
ICEEMDAN-BiLSTM	0.00595	0.00386	0.99449
ICEEMDAN-BiLSTM-MH-Attention	0.01619	0.01335	0.99652
ICEEMDAN-TCN-BiLSTM-MH-Attention	0.00996	0.00694	0.99802

As shown in Table 4, the test set prediction results of the BILSTM model are 0.05183 for RMSE, 0.04398 for MAE, as 0.96435 for R^2, which shows that the model fit is high, but the error is relatively large. The combined ICEEMDAN-BiLSTM model has a substantial improvement in model performance by introducing ICEEMDAN and modal decomposition of the original data, the RMSE decreased by 88.5%, the MAE decreased by 91.2%, the R^2 improved by 3.1%, and the value of R^2 increased to 0.99652 by combining with the multi-attention mechanism. By adding the convolutional neural network, the ICEEMDAN-TCN-BiLSTM-MH-Attention combination model compared to the ICEEMDAN-BiLSTM-MH-Attention, the RMSE decreased by 38.5% to 0.00996, the MAE dropped by 48.1% to 0.00694, while the R^2 increased by 0.15% to 0.99802. The experimental results show that the prediction error can be effectively reduced and the goodness of fit can be improved by gradually optimizing the model structure. This study confirms the efficacy of the combined model in load forecasting activities by demonstrating that the addition of ICEEMDAN, MH-Attention, and TCN may successfully increase the model performance.

In addition, in order to reflect the deviation between the predicted and true values of different combination forecasting models, we collated the deviation between their predicted trends and true values.

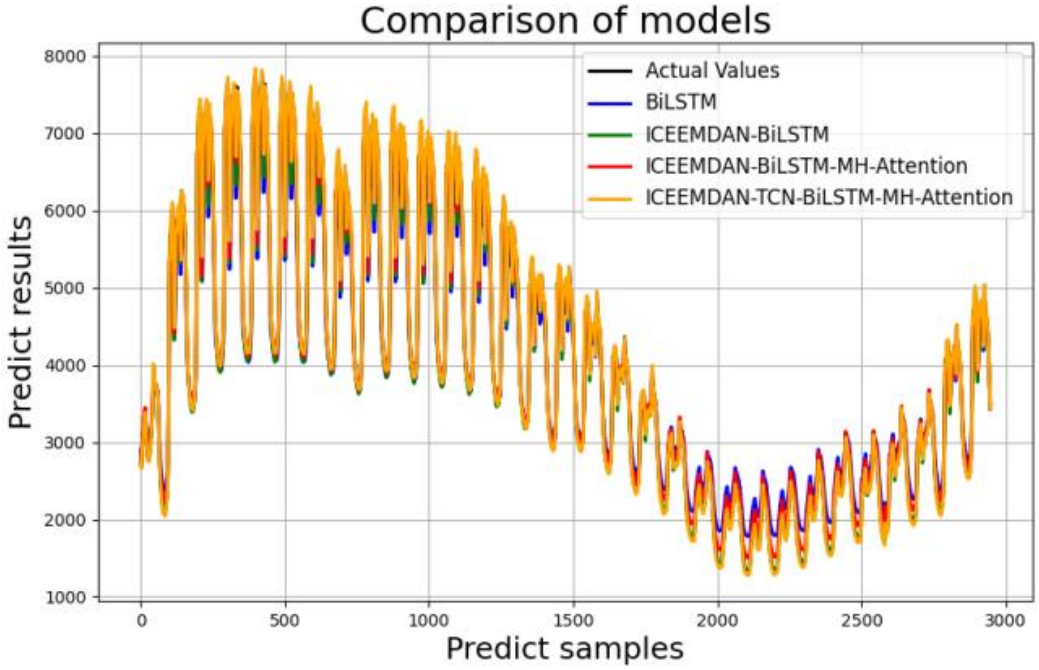

Figure 9. Comparison chart of predictions from different models

Figure 9 illustrates that the black curve denotes the true values, the blue curve indicates the BiLSTM prediction, the green curve represents the ICEEMDAN-BiLSTM prediction, the red curve signifies the ICEEMDAN-BiLSTM-MH-Attention prediction, and the yellow curve corresponds to the ICEEMDAN-TCN-BiLSTM-MH-Attention prediction. As more strategies are incorporated, the prediction curves increasingly align with the true values, with the final prediction almost perfectly matching the true trend, demonstrating the effectiveness of each strategy introduced.

4.4. Comparison experiment

This paper analyzes the performance of the ICEEMDAN-TCN-BiLSTM-MH-Attention model with the same dataset, and compares the performances of the two models in the short-term electricity load forecasting to the different models of forecasting in the same dataset to assess the efficiency of the ICEEMDAN-TCN-BiLSTM-MH-Attention model.The following were the comparison of different models: CEMDAN-BiGRU-SH-Attention; EWT-BiLSTM-Self-Attention; EMD-BiLSTM-MH-Attention; Conv1D-BiLSTM-SANET; and ICEEMDAN-TCN-BiLSTM-MH-Attention. Performance measures are RMSE, MAE, and R^2. The experimental results are presented in Table 5.

Table 5. Evaluation metrics for different strategy models

Models	RMSE	MAE	R^2
CEMDAN-BiGRU-SH-Attention	0.04184	0.03249	0.97677
EWT-BiLSTM-Self-Attention	0.02274	0.01666	0.97062
EMD-BiLSTM-MH-Atteention	0.05148	0.04354	0.96482
Conv1D-BiLSTM-SANET	0.02387	0.01777	0.98243
ICEEMDAN-TCN-BiLSTM-MH-Attention	0.00996	0.00694	0.99802

As shown in Table 5, compared with other models, the ICEEMDAN-TCN-BiLSTM-MH-Attention combination model proposed in this paper improves the RMSE by 58%-84% to 0.00996, reduces the MAE by 56%-80% to 0.00694, and reduces the R^2 by 1%-3% to 0.99802. The superiority of the models was verified.

Furthermore, to capture the discrepancy between the predicted and actual values of various models, we documented the trends of both the predicted and true values across these models.

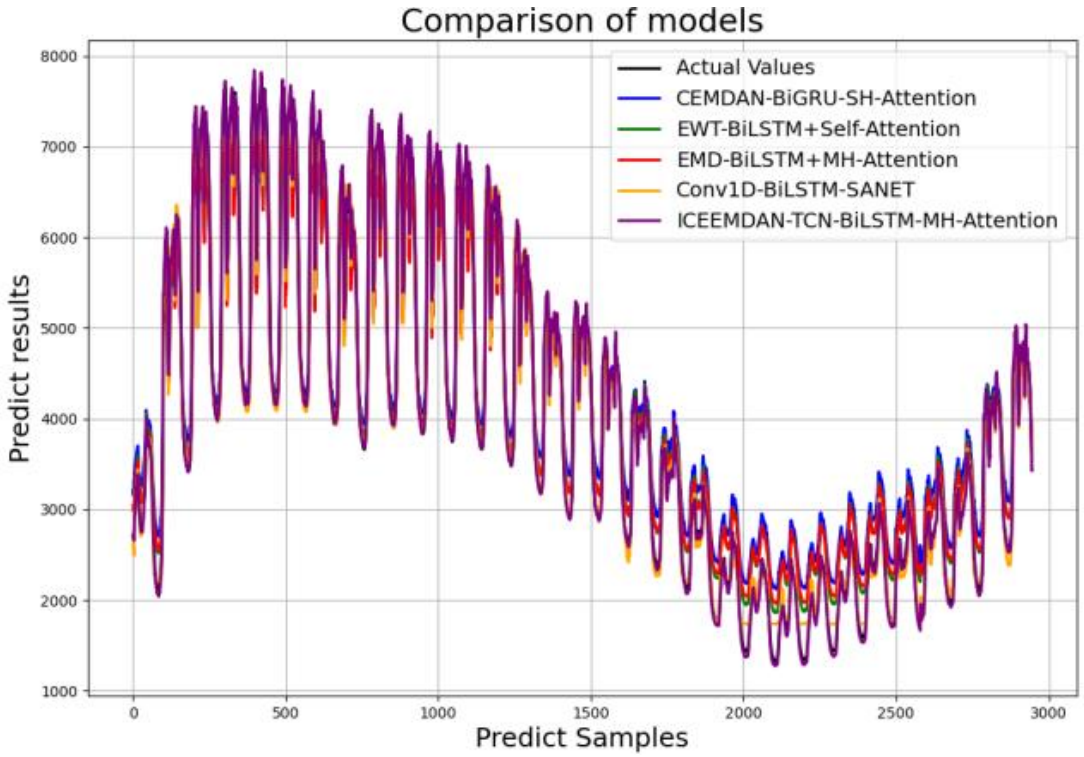

Figure 10. Shows the various model prediction strategies compared.

As shown in Figure 10, the black curve shows the true value, the blue curve shows the CEMDAN-BiGRU-SH-Attention model predicted trend, the green curve shows the EWT-BiLSTM-Self-Attention model predicted trend, the red curve shows the EMD-BiLSTM-MH-Attention model predicted trend, the yellow curve shows the Conv1D-BiLSTM-ANET model prediction trend, and the purple curve shows the ICEEMDAN-TCN-BiLSTM-MH-Attention model prediction trend. The green curve has the largest deviation, the rest of the models behave similarly to the true values, and the purple curve almost completely overlaps with the true values, thus proving the superiority of the models.

5. Conclusion

In this paper, we propose a hybrid model based on the ICEEMDAN-TCN-BiLSTM-MH-Attention mechanism for high-precision, real-time power load forecasting. Specifically, due to the complex nature of power load data, such as non-linearity and non-smoothness, the ICEEMDAN algorithm is applied for multi-layer decomposition. This process effectively enhances the smoothness of the data, allowing the subsequent prediction model to capture key features more effectively. Furthermore, considering that the BiLSTM algorithm tends to overlook long-term dependencies when dealing with long time series, we introduce the TCN module and the MH-Attention mechanism. Through the activation operation of the MH-Attention mechanism, the TCN module improves the ability of the model to capture local features through the convolution operation, and the MH-Attention mechanism can realize the most important time of load prediction, so as to realize the learning of the important information.

The proposed model is validated using an electricity load dataset from a power plant in the southern region. Experimental results demonstrate the model's strong forecasting capability, with R^2 reaching 0.99802, RMSE reaching 0.00996, and MAE reaching 0.00694, confirming the model's effectiveness. When compared to other combined prediction models, R^2 increases by 1.5%-3.4%, RMSE decreases by 56.2%-80.6%, and MAE drops by 58.3%-84.1%, further validating the model's superiority. The ICEEMDAN-TCN-BiLSTM-MH-Attention mechanism effectively integrates the advantages of data decomposition, convolutional feature extraction, and attention mechanisms, enabling it to fully explore temporal features and key patterns in power load data. The purpose of this system is to meet the requirements for real-time performance in power load forecasting and high accuracy. Looking forward, the model could be further enhanced by incorporating other ensemble learning methods (e.g., XGBoost, Random Forest) to further improve prediction accuracy and robustness, enabling its application to diverse, multi-region power load data forecasting.

References

[1] Ma K, Nie X, Yang J, et al. A power load forecasting method in port based on VMD-ICSS-hybrid neural network [J]. Applied Energy, 2025, 377 (PB): 124246.

[2] Hua Q, Fan Z, Mu W, et al. A short-term power load forecasting method using CNN-GRU with an attention mechanism [J]. Energies, 2024, 18 (1): 106 - 122.

[3] Lin Y, Luo H, Wang D, et al. An ensemble model based on machine learning methods and data preprocessing for Short-Term electric load forecasting [J]. Energies, 2017, 10 (8): 1186.

[4] Yang R, Zhang C, Li F, et al. A method for short-term power load forecasting [J]. Software, 2017, 38 (03): 6 - 11.

[5] Li X, Zhu Z, Zhang C, et al. Power data analysis and mining technology in smart grid [J]. Energy Informatics, 2024, 7 (1): 93 - 112.

[6] Raj S A, Oliver H D, Srinivas Y, et al. Wavelet denoising algorithm to refine noisy geoelectrical data for versatile inversion [J]. Modeling Earth Systems and Environment, 2016, 2 (1): 1 - 11.

[7] Peterson V, Vissani M, Luo S Y, et al. A supervised data-driven spatial filter denoising method for speech artifacts in intracranial electrophysiological recordings[J]. Imaging Neuroscience, 2024, 1 - 22.

[8] Wang Y C, Jiang B J, Han D, et al. multi-channel short-term power load prediction model based on VMD data decomposition [J]. Fluid Measurement & Control, 2024, 5 (03): 25 - 31.

[9] Ntunka M, Loveday B. Sustainable production of electrolytic manganese dioxide (EMD): A conceptual flowsheet [J]. Cleaner Chemical Engineering, 2025, 11: 100145.

[10] Hu C J, Zhao Y, Jiang H, et al. Prediction of ultra-short-term wind power based on CEEMDAN-LSTM-TCN [J]. Energy Reports, 2022, 8 (S8): 483 - 492.

[11] Sanjeev, Rohit K, Ajay S, et al. Development of seasonal ARIMA model to predict wholesale price of rice in Delhi market [J]. Current Journal of Applied Science and Technology, 2022, 155 - 161.

[12] Zhu X, Cui J, Li Y. Energy savings bottleneck diagnosis of cooling system based on integrated correlation analysis[J]. The Canadian Journal of Chemical Engineering, 2022, 101 (8): 4762 - 4770.

[13] Siti A, Asi A S, Didit A, et al. Exploratory weather data analysis for electricity load forecasting using SVM and GRNN, case study in Bali, Indonesia [J]. Energies, 2022, 15 (10): 3566 - 3582.

[14] Wang A, Yu Q, Wang J, et al. Electric load forecasting based on deep ensemble learning [J]. Applied Sciences, 2023, 13 (17): 9706 - 9725.

[15] Feng G F, Ling L P, Chiang W H. Short-term load forecasting based on empirical wavelet transform and random forest [J]. Electrical Engineering, 2022, 104 (6): 4433 - 4449.

[16] Zhang Y. Research of short-term power load forecasting based on improved GA-BP neural network model [J]. Computer Engineering and Applications, 2009, 45 (13): 223 - 226.

[17] Yang G, Du S, Duan Q, et al. Short-term price forecasting method in electricity spot markets based on Attention-LSTM-mTCN [J]. Journal of Electrical Engineering & Technology, 2022, 17(2): 1009 - 1018.

[18] Han J C, Zeng P. Residual BiLSTM based hybrid model for short-term load forecasting in buildings [J]. Journal of Building Engineering, 2025, 99: 111593 - 111609.

[19] Zheng Q G, Kong R L, Su E Z, et al. Approach for short-term power load prediction utilizing the ICEEMDAN-LSTM-TCN-Bagging Model [J]. Journal of Electrical Engineering & Technology, 2024, 20 (1): 1 - 13.

[20] Gou Y F, Guo C, Qin R S. Ultra short-term power load forecasting based on the fusion of Seq2Seq BiLSTM and multi head attention mechanism [J]. PloS one, 2024, 19 (3): e0299632.