International Journal of Advanced Science

https://doi.org/10.70731/pegmyh06

Multi-Feature Power Load Forecasting Model Based on Hyperparameter Optimization and Ensemble Learning

Mingshen Xu [a], **Teng Zhang** [a,*], **Xiaotian Li** [a]

[a] *Department of Mathematics and Physics, North China Electric Power University, Baoding 071003, China*

KEYWORDS

Power Load Forecasting,
Ensemble Learning,
Convolutional Neural Network,
BiLSTM,
Random Forest (RF),
Vector Weighted Average Optimization Algorithm (INFO)

ABSTRACT

Power load forecasting is crucial for the economic dispatch and safe operation of power grids, yet the fluctuating and unstable nature of power load data often limits the accuracy of traditional single-model approaches. This paper presents an integrated learning model designed to improve forecasting accuracy and stability by addressing these challenges. The model incorporates improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN), a vector-weighted average optimization algorithm (INFO), convolutional neural networks (CNN), bidirectional long short-term memory (BiLSTM), and random forest (RF). By thoroughly analyzing and preprocessing the data, the approach effectively handles its nonlinear, non-stationary characteristics. The combination of CNN and BiLSTM enhances the model's ability to capture temporal and spatial features, while RF strengthens generalization. The INFO algorithm dynamically adjusts weights and parameters during training, resulting in significantly improved predictive performance. Experimental results on data from the Australian electricity market confirm that the proposed model outperforms existing approaches in key performance metrics, showcasing its effectiveness and potential for practical applications in power load forecasting.

1. Research Background and Literature Review

Power load forecasting plays an important role in power system. As a key component of power system, a complex multidimensional nonlinear system [1], load-oriented forecasting methods are indispensable. In particular, short-term power load forecasting is very important to maintain the dynamic balance between power generation and power consumption. Accurate short-term load forecasting is of great value and significance for ensuring the economic and safe operation of power grid. With the continuous expansion of the application market, experts and scholars pay more and more attention to short-term power load forecasting [2]. In power system, the accuracy of load forecasting directly affects the stability and economy of power supply. Short-term load forecasting usually refers to the prediction of electricity demand in the next few hours to several days. This forecast is crucial for grid operators as it helps them decide when to

[*] Corresponding author. E-mail address: xianyu041230@outlook.com

Received 23 December 2024; Received in revised from 16 February 2025; Accepted 22 February 2025; Available online 28 February 2025.

turn on or off generating units and how to structure transactions in the electricity market. Accurate forecasting can reduce the reliance on backup power and reduce the cost of generation while ensuring the reliability of the power supply.

To make predictions, experts and engineers have developed a variety of forecasting techniques and methods. In the ultra-short-term power load forecasting, the commonly used and more traditional techniques include time series analysis [3], fuzzy regression [4] and machine learning. Time series analysis includes autoregressive model [5,6], exponential smoothing [7], and gray model analysis [8,9]. However, one limitation of time series analysis is that the prediction technique is not applicable when the research object has abnormal change period or lack of adaptability law in the selected time series. Traditional prediction methods based on machine learning, including support vector machine [10,11], decision tree [12], etc., have also been applied, but the accuracy of relevant methods in processing nonlinear data is relatively limited. In recent years, deep learning methods in the field of machine learning [13], especially recurrent neural networks and convolutional neural networks, have achieved remarkable results in power load prediction due to their powerful network architecture.

Literature [14] and [15] optimized the machine learning prediction method by improving the method, effectively reduced the uncertainty fluctuation of load series, and improved the accuracy of prediction and the reliability of the model. Literature [16] proposes a multi-information fusion deep learning framework based on long short-term memory network (LSTM) for residential power load prediction. Through multi-level information extraction, the framework can more accurately capture the characteristics of residential load in a specific area, better track the changing trend of residential load and have higher prediction accuracy than single-task deep learning and single neural network. In literature [17], based on the LSTM model, cyclic jump components and linear autoregressive components were added, and a LSTNet prediction model capable of capturing the short-term local dependence of the load in the distribution area was constructed. In literature [18], a load prediction model based on random distributed embedding framework and BP neural network was constructed. Kernel density estimation method was used to fit multiple prediction results, and the final load prediction value was obtained by aggregation estimation method, effectively improving the influence of data dimension on the prediction accuracy of BP neural network. Literature [19] proposes a new method to convert load data into RGB images, and then apply LSTM artificial neural network for short-term power load prediction, which is a preliminary exploration of graphical load artificial intelligence prediction. Literature [20] concludes that long short-term memory neural network (LSTM) has better prediction performance by comparing common deep learning models. The research results of literature [21] show that BiLSTM has obvious advantages compared with traditional long short-term memory neural network (LSTM) when processing time series data. By introducing bidirectional structure, BiLSTM can not only capture the forward sequence information in time series, but also effectively use the reverse sequence information, thus improving the accuracy and efficiency of prediction. Yang Long et al. [22] made full use of this feature of BiLSTM, dug deep into the hidden information of historical and future data, and successfully improved the accuracy of prediction.

However, problems such as local minimum and overfitting may occur if the hyperparameters are improperly adjusted during the training of relevant models. To solve this problem, a series of combinatorial prediction models using intelligent optimization algorithms are proposed. In literature [23], Grey Wolf optimization algorithm (GWO) was used to optimize BiLSTM model, and the optimal hyperparameters were obtained. In literature [24], a method combining attention mechanism with whale optimization algorithm (WOA) was proposed to optimize the prediction model of BiLSTM neural network. The core idea of this method is to use the attention mechanism to enhance the model's attention to the key features in the time series data, and at the same time optimize the parameters of the BiLSTM model through the whale optimization algorithm to improve the prediction performance.

Due to the temporal fluctuation of short-term power load data, noise anomaly data may interfere with the load prediction process, making it difficult for the model to achieve the expected accuracy. Therefore, some data processing methods have been applied to load forecasting models: Wei et al. [25] and Lopez C et al. [26] respectively adopted empirical mode decomposition (EMD) and integrated empirical mode decomposition (EEMD) in their work on power load sequence processing. However, in order to overcome their shortcomings, Torres et al. [27] proposed a complete set empirical Mode decomposition (CEEMDAN) method with adaptive white noise. On this basis, Colominas et al. [28] continued to improve the algorithm and proposed the ICEEMDAN method. It further improves the accuracy of the decomposed signal, and facilitates the relevant model to learn the characteristics of the data, thus improving the prediction effect.

In recent years, some integrated load forecasting methods have been gradually proposed: Based on the traditional convolutional neural network (CNN), literature [29]-[31] introduces the concepts of extended convolutional network, causal convolutional network and residual network, and proposes a novel sequential convolutional neural network. This kind of network shows excellent ability to capture time series features in load forecasting tasks. Literature [32] proposed a load forecasting model combining XGBoost and BiLSTM. In this model, the load data and its influencing factors are first processed through an attention-mechanism-based Bi-LSTM model, while the XGBoost limit gradient lifting algorithm is used to generate the prediction results respectively. Then, the inverse error method is used to assign the corresponding weights to the results of these two models. Through this combination, the model not only makes full use of the ability of long short-term memory network to capture the characteristics of load data, but also combines the regularization advantage of the objective function of XGBoost algorithm. Compared with the simple neural network combination model, this model has stronger generalization ability, and after weight distribution, the predicted results have higher precision.

Based on the above analysis, this paper proposes an integrated learning model of fully adaptive noise set empirical Mode decomposition (ICEEMDAN), convolutional neural network (CNN), BiLSTM and Random forest (RF), which is specifically used to improve the accuracy and stability of power load prediction. Through comprehensive analysis and advanced pre-processing of power load data, the model can effectively deal with the nonlinear and non-stationary characteristics of the data. The combination of CNN and BiLSTM is used to optimize the capture of temporal and spatial features, while the random forest enhances the generalization ability of the model. In addition, by introducing a new vector weighted average optimization algorithm (INFO), the weights and parameters of the model are dynamically optimized during the learning process, which significantly improves the prediction performance. Then this paper conducts relevant experiments based on the Australian power data set, and verifies the performance of the proposed model through several sets of experiments, demonstrating its application potential and effectiveness in the field of power load forecasting.

2. Data Analysis and Preprocessing

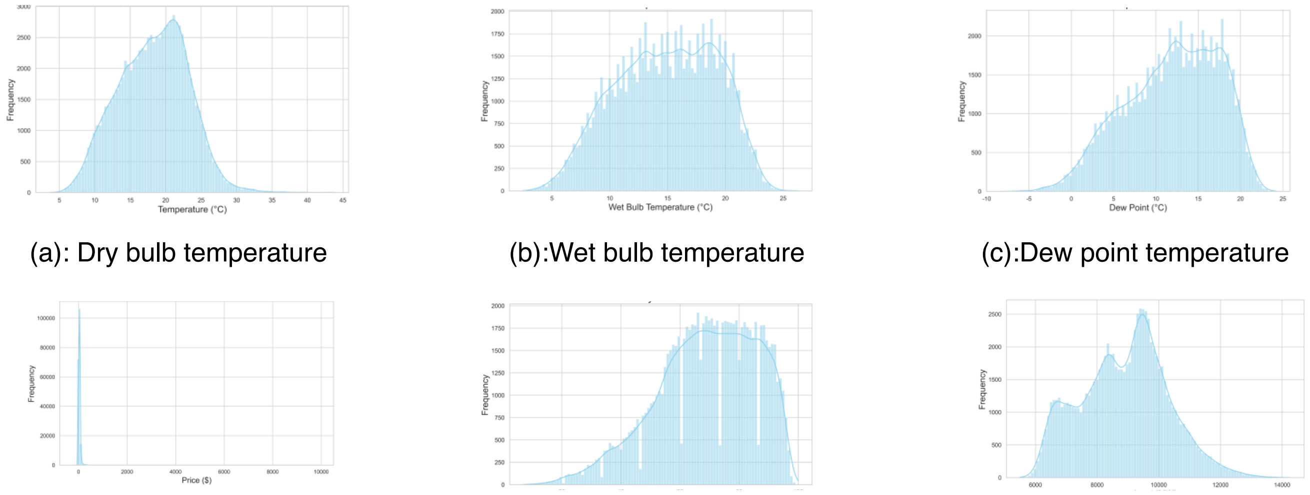
2.1. Descriptive Statistics

In the data preprocessing stage, this paper firstly makes descriptive statistics on the relevant data sets of the Australian electricity market. The dataset covers a wide range of multi-year historical records since 2006, covering key variables such as dry bulb temperature, dew point temperature, wet bulb temperature, humidity, electricity price, and power load. The relevant statistics of specific variables are shown in Table 1.

At the same time, this paper uses Python to check the missing values of relevant data and finds that the

Table 1 I Descriptive statistics of main variables

Variable	Record number	Mean value	Standard deviation	Minimum value	The first quartile	Median	The third quartile	Maximum value
Dry bulb temperature	87648	18.26	4.89	3.70	14.70	18.50	21.80	43.80
Dew point temperature	87648	11.92	5.47	-8.40	8.00	12.45	16.35	24.20
Wet bulb temperature	87648	14.88	4.29	2.50	11.60	15.10	18.40	26.30
humidness	87648	68.90	16.86	7.00	58.00	70.00	82.50	100.00
electrovalence	87648	42.40	215.64	-264.31	21.80	25.81	36.94	10000.00
Electric load	87648	8894.00	1409.05	5498.36	7879.67	8992.59	9832.86	14274.15



(a): Dry bulb temperature (b):Wet bulb temperature (c):Dew point temperature

(d):electrovalence (e):humidness (f):Electric load

Figure 1 I Visualization of variable distribution

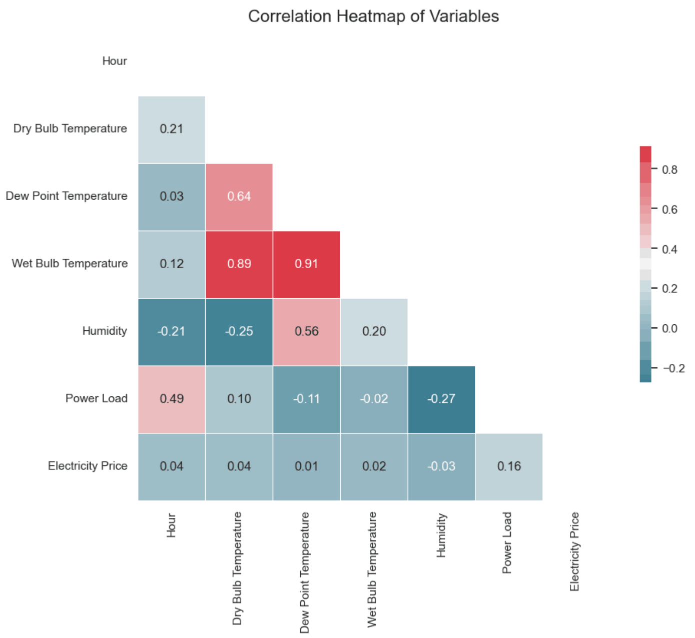


Figure 2 I Pairwise correlation analysis diagram of key power indicators

data is coherent and without missing values, so there is no need to fill in the missing values and other related work.

Below, this paper analyzes and visualizes the distribution of data. Figure 1-2.

To sum up, in view of correlation analysis, this paper selects all the data in the data as characteristics to forecast the power load data.

3. Relevant Methodology

3.1. An Improved Fully Extended Empirical Mode Decomposition

ICEEMDAN decomposition is improved by CEEMDAN algorithm. By adding a set of K-order IMF to a specific white noise, IceEMDAN decomposition adaptively matches the original signal by using the adaptive characteristics of IMF. ICEEDMAN can handle mode aliasing better than CEEMDAN decomposition, and can adapt the background noise to suppress its influence on the decomposition results. The following is a detailed introduction to the ICEEMDAN implementation process.

Firstly, Gaussian white noise is added to the original time series and a new time series is obtained. The initial residual is obtained from this sequence and the average of the results of the first round of empirical mode decomposition (EMD algorithm) is calculated. The details are shown in formula (1):

$$m_1 = \frac{1}{I} \sum_{i=1}^{I} R\left(x + \epsilon_0 \omega^{(i)}\right) \tag{1}$$

In the integrated model, $\omega^{(i)}$ is Gaussian white noise, and ϵ_0 represents the expected noise reduction ratio; The operator $R(\cdot)i$ is used to calculate the mean value using the EMD algorithm.

Secondly, for the initially obtained first modal component IMF ($k = 1$), as shown in equation (2) :

$$I\tilde{M}F_1 = x - m_1 \tag{2}$$

The process is repeated continuously, and specific white noise is further introduced, while local mean decomposition is used to calculate the residual of the second group of IMF, at which time the second modal is generated as shown in equations (3) - (4) :

$$m_2 = \frac{1}{I}\sum_{i=1}^{I} R\left(x + \epsilon_0 E_1\left(\omega^{(i)}\right)\right) \tag{3}$$

$$\tilde{M}F_2 = m_1 - m_2 = m_1 - \frac{1}{I}\sum_{i=1}^{I} M\left(m_1 + \epsilon_1 E_2\left(\omega^{(i)}\right)\right) \tag{4}$$

Here $N_j(\cdot)$ is the j first mode component obtained by EMD decomposition.

Then calculate the sum of k residual components, as shown in equation (5) :

$$m_k = \frac{1}{I}\sum_{i=1}^{I} R\left(m_{k-1} + \epsilon_{k-1} E_k\left(\omega^{(i)}\right)\right) \tag{5}$$

The corresponding k modal components are also given by equation (6) :

$$I\tilde{M}F_k = r_{k-1} - r_k \tag{6}$$

Repeat the above data filtering process until the resulting residuals cannot be decomposed further. The final signal x can be expressed as equation (7):

$$x = o + \sum_{k=1}^{K} IMF_k \tag{7}$$

Where o represents the final residual and K represents the IMF total.

3.2. Vector Weighted Average Optimization Algorithm (INFO)

INFO is a new intelligent optimization algorithm proposed in 2022, which is optimized by using different weighted average rules between vectors [33]-[34]. The INFO algorithm consists of D population of N_p vectors in the A-dimensional search domain. In the initialization stage, the INFO algorithm mainly has two control parameters, weighted average factor δ and scale factor σ, and these two control parameters can be dynamically updated according to the generated results without manual adjustment.

The implementation of INFO mainly includes three steps: update rule, vector merge and local search.

3.2.1. Update the Rules

Update rules are the basis of INFO algorithm. At this stage, new vectors are formed based on convergent acceleration and mean rules. In the INFO algorithm, the update rule phase increases population diversity during the search process. This phase consists of two main parts. The first part of the mean-based approach starts with a random initial solution and updates a set of randomly selected vector weighted average information to the next solution. In the second part, convergence acceleration is added to improve the convergence speed of the algorithm.

The main formula definition of the update rule is as formulas (8) - (9) :

$$\begin{cases} z_p^{1g} = x_1^g + \sigma \times R + \dfrac{\text{randn}(x_{bs} - x_1^{g\alpha 1})}{f(x_{bs}) - f(x_1^{g\alpha 1}) + 1}, \\ z_p^{2g} = x_{bs} + \sigma \times R + \dfrac{\text{randn}(x_1^g - x_1^{g\alpha b})}{f(x_{bs}) - f(x_1^{g\alpha 1}) + 1}, \quad \text{randn} < 0.5 \end{cases} \tag{8}$$

$$\begin{cases} z_p^{1g} = x_\alpha^g + \sigma \times R + \dfrac{\text{randn}(x_{\alpha 2}^g - x_{\alpha 3}^g)}{f(x_{\alpha 2}^g) - f(x_{\alpha 3}^g) + 1}, \\ z_p^{2g} = x_{bs} + \sigma \times R + \dfrac{\text{randn}(x_{\alpha 1}^g - x_{\alpha 2}^g)}{f(x_{\alpha 1}^g) - f(x_{\alpha 2}^g) + 1}, \quad \text{randn} \geq 0.5 \end{cases} \tag{9}$$

Where: z_p^{1g} and z_p^{2g} are the new position vectors of the g iteration, p=1,2... Np; σ is the scaling ratio of the vector; $f(x)$ represents the fitness function of x; x_{bs} is the optimal solution vector in the g generation population α_1, α_2, α_3 is a random different integer distributed in $[1,N_p]$ and $\alpha_1 \neq \alpha_2 \neq \alpha_3 \neq 1$; randn is a random value with a standard normal distribution: the updated formula for α is shown in equation (10):

$$\alpha = 2 \times \exp\left(-\frac{g}{\max g}\right) \tag{10}$$

The definition formula of the mean value rule is as formula (11):

$$M = e_1 \times W_p^{1g} + (1 - e_1) \times W_p^{2g} \tag{11}$$

Where, p represents a random integer from 1 to N_p; g represents the number of iterations; e_1 is a random number whose value ranges from (0,0.5). W_1 represents the weighted average value of the vector, which can be expressed specifically as shown in (12) :

$$W_p^{1g} = \sigma \times \frac{\omega_1(x_a - x_b) + \omega_2(x_a - x_c) + \omega_3(x_b - x_c)}{\omega_1 + \omega_2 + \omega_3 + \varphi} + \varphi + e_1 \tag{12}$$

Where a、b and c are different integers randomly selected between them. σ stands for scaling ratio, and its solution formula is $\alpha = 2 \times \exp\left(-\frac{g}{\max g}\right)$; ω_1, ω_2 and ω_3 are weighting functions, which are used to calculate the weighted average of vectors and improve the global search ability. The specific equation is shown in equation (13):

$$\begin{cases} \omega = \max(f(x_a), f(x_b), f(x_c)) \\ \omega_1 = \cos\left(\pi - f(x_b) - f(x_a)\right) \times \exp\left(\dfrac{f(x_b) - f(x_a)}{\omega}\right) \\ \omega_2 = \cos\left(\pi - f(x_c) - f(x_a)\right) \times \exp\left(\dfrac{f(x_c) - f(x_a)}{\omega}\right) \\ \omega_3 = \cos\left(\pi - f(x_c) - f(x_b)\right) \times \exp\left(\dfrac{f(x_c) - f(x_b)}{\omega}\right) \end{cases} \tag{13}$$

Similarly, W_p^{2g} is defined by formula (14)

$$W_p^{2g} = \delta \times \frac{\omega_1(x_{bs} - x_{bt}) + \omega_2(x_{ws} - x_{bs}) + \omega_3(x_{ws} - x_{bt})}{\omega_1 + \omega_2 + \omega_3 + \varphi} + \varphi + e_1 \tag{14}$$

Among them, the weighting function formula of ω_1, ω_2 and ω_3 is shown in equation (15) :

$$\begin{cases} \xi = f(x_{ws}) \\ \omega_1 = \cos\left(\pi + \left(f(x_{bs}) - f(x_{bt})\right) \times \exp\left(\dfrac{f(x_a) - f(x_b)}{\xi}\right)\right) \\ \omega_2 = \cos\left(\pi + \left(f(x_{bs}) - f(x_{ws})\right) \times \exp\left(\dfrac{f(x_a) - f(x_c)}{\xi}\right)\right) \\ \omega_3 = \cos\left(\pi + \left(f(x_{bt}) - f(x_{ws})\right) \times \exp\left(\dfrac{f(x_b) - f(x_c)}{\xi}\right)\right) \end{cases} \tag{15}$$

Among them, x_{bs}, x_{bt}, and x_{ws} are respectively the optimal solution vector, suboptimal solution vector, and worst solution vector of the g generation in the iteration process.

3.2.2. Vector Merging Phase

Vector merging refers to combining the obtained vector with vector updating rules to form a vector with stronger local search ability. In this phase, the INFO algorithm combines the two vectors obtained during the rule update phase to generate a new vector. The combined formula is shown in equation (16) :

$$u_1^g = \begin{cases} x_1^g, e_1 > 0.5 \\ z_1^g + \lambda \,|\, z_1^g \,| - z_2^g, e_1 < 0.5 \text{ and } e_2 < 0.5 \\ z_1^g + \lambda \,|\, z_1^g \,| - z_2^g, e_1 < 0.5 \text{ and } e_2 \geq 0.5 \end{cases} \tag{16}$$

Where, u_p^g is the new vector obtained from the combination of vectors of generation g; $u = 0.05 \times e_2^p$.

3.2.3. Local Search Stage

In order to improve and prevent falling into the local optimal solution, the INFO algorithm carries out local search after the completion of vector merging to further promote the convergence of the operator to the global optimal solution. If $randn$<0.5, a new vector will be generated, and $randn$ is a random value of [0,1]. Specifically, it can be expressed as formula (17) :

$$\begin{cases} x_{\text{randn}} = \varphi \times x_{\text{average}} + (1 - \varphi) \times \left(\varphi \times x_{bt} + (1 - \varphi) \times x_{bs}\right) \\ x_{\text{average}} = \dfrac{x_a + x_b + x_c}{3} \end{cases} \tag{17}$$

Where φ is a random value of [0,1]; x_{randn} is a new solution consisting of $x_{average}$, x_{bt}, and x_{bs}. The new vector can then be represented as shown in equation (18) :

$$u_1^g = \begin{cases} x_{bs} + \text{randn} \times \left(M - e_2 \times \left(x_a^g - x_{bs}^g\right)\right), & r_1 < 0.5 \text{ and } r_2 < 0.5 \\ x_{\text{randn}} + \text{randn} \times \left(M - e_2 \times \left(h_2 \times x_{\text{randn}} - h_1 \times x_{bs}\right)\right), & r_1 < 0.5 \text{ and } r_2 \geq 0.5 \end{cases} \tag{18}$$

Where, both h_1 and h_2 are random numbers, and the value formula is shown in equation (19) :

$$\begin{cases} h_1 = \begin{cases} 2e, \text{if } k > 0.5 \\ 1, \ \text{ if } k < 0.5 \end{cases} \\ h_2 = \begin{cases} e, \text{if } k > 0.5 \\ 1, \text{if } k < 0.5 \end{cases} \end{cases} \tag{19}$$

The pseudo-code of the algorithm is shown in Table 2.

Table 2 | Pseudo code of INFO algorithm

```
Step 1. initialize
Set parameters $N_p$ and max$_g$
Generate initial population $P^0 = X_i^0, \ldots, X_{Np}^0$
Calculate the objective function values of all vectors $f(X_i^0)$, where $i = 1,...,Np$
Determine the optimal solution vector $X_{bs}$
Step 2.
for $g = 1$ to max$_g$ do
    for $i = 1$ to $Np$ do
        Select the random integer $a, b, c$ in $[1,Np]$, where $\alpha_1 \neq \alpha_2 \neq \alpha_3 \neq i$
        Update rule phase
        Calculate the position vectors $z1_p^g$ and $z2_p^g$
        Vector merging phase
        Calculate the merged new vector$u_p^g$
        Local search phase
        Perform local search calculations
```

```
Calculate the value of the objective function $f(u_i^g, j)$
if $f(u_i^g, j) < f(x_i^g, j) then x_i^{g+1}, j = u_i^g, j$
Otherwise $x_i^{g+1}, j = x_i^g, j$
end for
Update the optimal solution vector $x_{bs}$
end for
Step 3.
Return the final solution vector $x_{best,j}^g$
```

3.3. CNN Convolutional Neural Network

Convolutional neural network (CNN) is an important feedforward neural network. Its core mechanism lies in the convolution layer, which performs convolution operation by convolution checking the input complex multi-feature data, so as to extract the potential features of the data deeply. In addition, the CNN includes a pooling layer whose primary function is downsampling, which is designed to simplify the input data to reduce computational complexity while ensuring that key features are preserved. Finally, the fully connected layer is responsible for integrating information from the convolution layer and the pooling layer to output classification or regression results. In the process of training CNN, we usually use backpropagation algorithm to further optimize its performance.

The convolution layer formula is shown in equation (20)：

$$x_k^l = \sigma\left(\sum_{i\in M_k} x_i^{l-1} * W_{ik}^l + a_j^l\right) \tag{20}$$

Where, x_k^l is the input and output of the k convolution kernel in layer σ; σ is the activation function; M_j represents the set composed of all input mapping layers; W_{ik}^l is the weight matrix of the kth convolution kernel in layer b_k^l; a_k^l is the biased term.

The output results of the convolutional layer will enter the pooling layer, and the dimensionality is reduced by means of maximum pooling and average pooling, so as to facilitate feature extraction.

The fully connected layer joins all the output results of the pooled layer and outputs them to the classifier. The forward propagation output of the fully connected layer can be expressed as equation (21) :

$$x_k^{l+1} = \sum_{k=1}^{n} W_{ik}^l x_k^l + a_k^l \tag{21}$$

Where, the $k + 1$ output of layer $l + 1$ is $x^{l+1(k)}$. CNN network structure diagram as shown in Figure 3.

3.4. Bidirectional Long Short-Term Neural Network (BiLSTM)

Although CNN has the ability to automatically extract multi-dimensional spatial features from data samples, it is poor in processing time series data with strong time dependence. In contrast, LSTM can effectively solve long-term dependency problems due to the introduction of gated units. By combining the two methods, the ability of extracting temporal features and computing performance can be improved. LSTM uses input gate, output gate and forget gate to control the selective flow of information, solves the problem of gradient disappearing and gradient explosion, and is widely used in the prediction of high time correlation. As shown in equation (22), the relation expression between the current state of the LSTM gate control unit and the state of the previous time is presented.

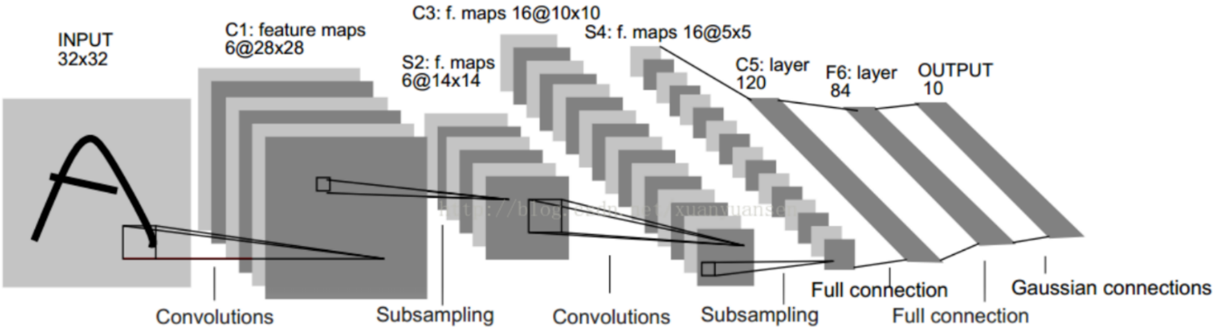


Figure 3 I CNN algorithm structure diagram

$$\begin{cases} f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\ i_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_i\right) \\ \tilde{C}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \\ o_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_o\right) \end{cases} \tag{22}$$

σ is sigmod activation function; tanh is hyperbolic tangent activation function; $h-1$ is the output at time $t-1$; $f, i_t, \tilde{C}_t$ and o_t are respectively the t time forgetting gate, the input gate, the state matrix and the output gate. The weight of W_f, W_i, W_c and W_o corresponding layers; b_f, b_i, b_c and b_o, respectively, are the bias coefficients of each corresponding layer.

BiLSTM, by running two LSTM networks in time dimension at the same time, splicing or merging information in both directions, and sharing weight matrix to connect and influence each other, can better extract bidirectional change features of time series, so as to understand and represent sequence data more comprehensively. For BiLSTM, its final state can be expressed as equation (23) :

$$y_t = \sigma\left(W_y \cdot \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] + b_y\right) \tag{23}$$

Where, $\overrightarrow{h_t}$ is the state of forward hidden layer at time t; $\overleftarrow{h_t}$ backward hidden layer state; W_y and b_y are the weight matrix and the bias term respectively. y_t is the output at time t.

The structure of BiLSTM is shown in Figure 4.

3.5. RandomForest Algorithm

Random forest is an ensemble learning model consisting of multiple decision trees, which is widely used in classification and regression tasks. This model randomly selects samples from the original data through self-sampling, and each tree is built independently without pruning, ensuring that the complex structure of the data is captured. When splitting nodes, the random forest randomly selects some features, effectively reducing the overfitting risk and enhancing the robustness of the model. In time series analysis, random forest predicts future values by analyzing historical data within a time window, and is able to handle nonlinear relationships and seasonal changes in the data. The model structure diagram is shown in Figure 5.

3.6. Construction of Learning Model

Ensemble learning is a method using multiple basic models. By integrating multiple weak classifiers, using sample weighting and learner weighting, a powerful learner with stronger generalization ability than a single model is formed. Its core idea is to create multiple base learners by applying different sampling or preprocessing techniques to the training data, and combine these learners into a comprehensive integrated model to enhance the final prediction effect and generalization ability. This approach requires each individual learner to maintain both appropriate accuracy and differentiation from one another.

Info-cnn-bilstm-rf model is an advanced integrated learning system, which combines the advantages of volume vector weighted average optimization algorithm (INFO), convolutional neural network (CNN),

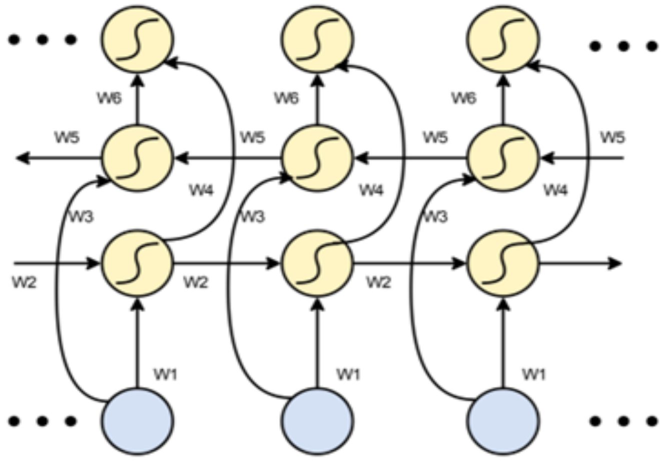


Figure 4 | BiLSTM algorithm structure diagram

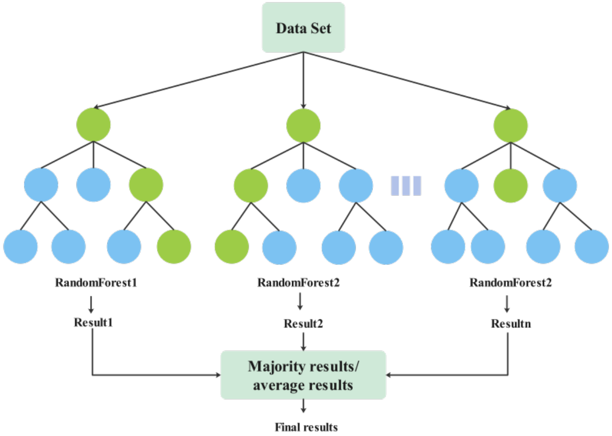


Figure 5 | Structure diagram of random forest algorithm

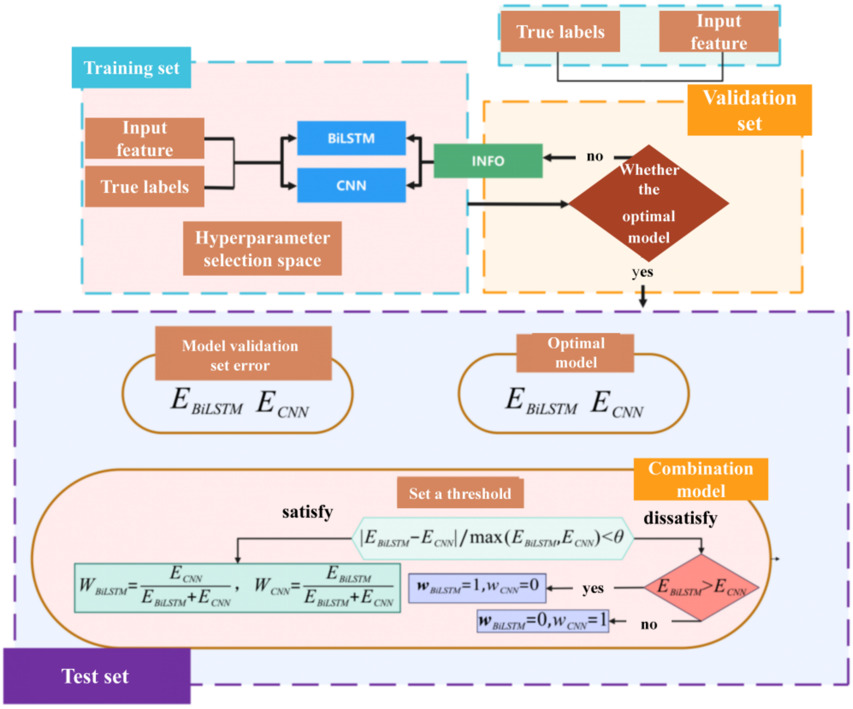

True labels
Input feature
Training set
Validation set
Input feature
True labels
BiLSTM
CNN
INFO
no
Whether the optimal model
yes
Hyperparameter selection space
Model validation set error
E_{BiLSTM} E_{CNN}
Optimal model
E_{BiLSTM} E_{CNN}
Combination model
Set a threshold
satisfy
dissatisfy
$|E_{BiLSTM}-E_{CNN}|/\max(E_{BiLSTM},E_{CNN})<\theta$
$W_{BiLSTM}=\frac{E_{CNN}}{E_{BiLSTM}+E_{CNN}}$, $W_{CNN}=\frac{E_{BiLSTM}}{E_{BiLSTM}+E_{CNN}}$
$w_{BiLSTM}=1, w_{CNN}=0$
yes
$E_{BiLSTM}>E_{CNN}$
$w_{BiLSTM}=0, w_{CNN}=1$
no
Test set

Figure 6 | Flowchart of INFO-CNN-BiLSTM-RF algorithm

BiLSTM and Random forest (RF) to improve the prediction accuracy of time series data. First, the raw data is fed into the CNN layer and BiLSTM layer, and the time dependence is captured synchronously by both models. Then, the INFO optimization algorithm was used to iteratively optimize the relevant hyperparameters of CNN and BiLSTM, and finally the optimal model was obtained. The features from CNN and BiLSTM are then sent to a fusion layer that uses a join strategy to integrate the two types of features. On this basis, the model optimizes the combination of these features through a dynamic weight adjustment mechanism. The weight adjustment is based on the predicted error of each component, and the specific adjustment formula is shown as equation (24) :

$$\Delta W=\frac{E_{\text{BiLSTM}}-E_{\text{CNN}}}{\max\left(E_{\text{BiLSTM}}, E_{\text{CNN}}\right)} \tag{24}$$

Where $E_{\text{BiLSTM}}, E_{\text{BiLSTM}}$ represent the prediction errors of BiLSTM and CNN models on the verification set respectively. The weight update rule is formula (25) :

$$W_{\text{new}}=\alpha \Delta W+W_{\text{old}} \tag{25}$$

The weighted features are fed into the random forest layer for a final prediction. Random forests improve prediction accuracy and stability by integrating multiple decision trees. The specific construction process of the model is shown in Figure 6.

4. Experimental Analysis

4.1. Prediction Performance Evaluation Indicators

Root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) were selected as evaluation indexes to evaluate the performance of the proposed model. The specific calculation formula is as equations (26) - (28):

$$RMSE=\sqrt{\frac{1}{m}\sum_{s=1}^{m}\left(y-y_{\text{real}}\right)^{2}} \tag{26}$$

$$MAE=\frac{1}{m}\sum_{s=1}^{m}\left|y_{\text{pred}}-y_{\text{real}}\right| \tag{27}$$

$$MAPE=\frac{1}{m}\sum_{s=1}^{m}\frac{\left|y_{\text{pred}}-y_{\text{real}}\right|}{y_{\text{real}}} \tag{28}$$

Among them, the smaller the values of MAE, RMSE and MAPE, the higher the prediction accuracy of the evaluated model.

4.2. Model Parameter Setting

The proposed model is programmed based on MATLAB2023a platform. The PC configuration of the training model is as follows: The hardware parameters are NVIDIA GeForce RTX 3060,16 GB, DDR4, 3600 MHz, and Intel Core i9-10900k@3.7 GHz. The

Table 3 | Initial parameter Settings

Parameter name	Parameter value
Initial number of vectors	5
The maximum number of algorithm iterations	10
Optimization dimension	2
Optimization range 1	[50-500]
Optimization range 2	[5-60]
Learning rate	0.005
optimizer	Adam
Lot size	128
Training rounds	1000
Attenuation rate	0.2

Table 4 | Comparison of prediction indicators before and after model data decomposition

Models/related indicators	RMSE	MAE	MAPE
ICEEMDAN-INFO-CNN-BiLSTM-RF	75.012	54.294	0.627
INFO-CNN-BiLSTM-RF	199.22	161.69	1.88

initial parameters of the algorithm are shown in Table 3.

4.3. Initial Signal Decomposition

Due to the randomness and volatility of the original power load data, ICEEMDAN is used to decompose the original power load data in this paper to improve the accuracy of prediction. Figure 8 shows the sequence after ICEEMDAN decomposition for the power load.

After decomposition, this paper selected the undecomposed sequence and the decomposed sequence for experimental comparison. The specific performance is shown in Table 4.

The above experimental results show that compared with INFO-CNN-BiLSTM-RF model, the model integrated with ICEEMDAN technology shows significant improvement in various performance indexes. The RMSE, MAE and MAPE indexes of ICEEMDAN-INFO-CNN-BiLSTM-RF model data are significantly reduced, among which RMSE is reduced by 62%, MAE by 67% and MAPE by 67%. The above results show that ICEEMDAN decomposition can effectively increase the model's ability to recognize hidden patterns in data, thus improving the accuracy and stability of prediction.

To sum up, the subsequent experimental data will be pre-processed using this technology to ensure data quality and improve the overall performance of the model.

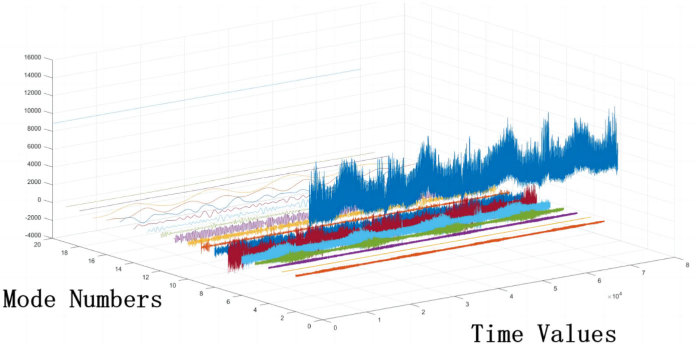


Figure 8 | Sequence after load data decomposition

4.4. Comparative Experimental Analysis of Optimization Algorithms

In order to verify the performance of the selected INFO optimization algorithm, this paper adopts a variety of optimization algorithms based on CNN-BiLSTM-RandomForest integration algorithm, including CGO(Chaos optimization algorithm), WSO(White Shark optimization algorithm), SA(simulated annealing algorithm), PSO(particle swarm optimization algorithm) and so on.

Subsequently, relevant experiments were conducted, and the results were shown in Table 5 and Figure 9.

Table 5 | Performance comparison of INFO optimized prediction models

Models/related indicators	RMSE	MAE	MAPE
INFO-CNN-BiLSTM-RF	75.011	54.294	0.627
CGO-CNN-BiLSTM-RF	124.845	90.701	1.048
PSO-CNN-BiLSTM-RF	119.885	86.920	1.004
WSO-CNN-BiLSTM-RF	135.103	97.993	1.131
SA-CNN-BiLSTM-RF	140.954	102.023	1.179

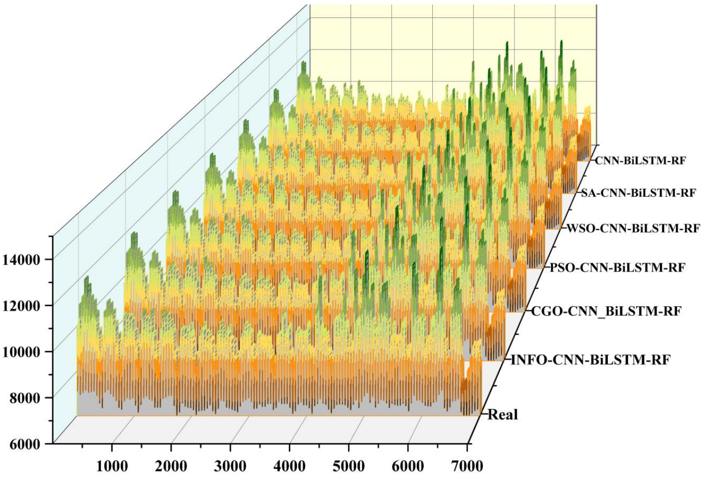


Figure 9 | Comparison of prediction effect of INFO optimization prediction algorithm

Note: Color gradient represents data size

The above results show that INFO-CNN-BiLSTM-RF achieves good results in three key performance indexes of RMSE, MAE and MAPE. Among them, in the comparison experiment with other traditional optimization algorithms, the performance improvement of the selected model compared with CGO-CNN_BiLSTM-RF in RMSE, MAE and MAPE were 66.43%, 67.00% and 67.16% respectively. The increase of PSO-CNN-BiLSTM-RF was 59.82%, 60.06% and 60.07%. The increase of WSO-CNN-BiLSTM-RF was 80.11%, 80.48% and 80.49%. SA-CNN-BiLSTM-RF showed the most significant improvement in various indexes, reaching 88.58%, 87.95% and 87.99%, respectively. See Figure 10 for details. In summary, it can be proved that the INFO optimization algorithm selected in this paper has the most efficient optimization performance under a limited number of iterations. Next, this paper conducts a comparative experiment for traditional optimization algorithms.

4.5. Comparative Experimental Analysis of Prediction Algorithms

In order to verify the predictive performance of INFO-CNN-BiLSTM-RF proposed in this paper, XGBoost(Limit gradient elevator), LightGBM (lightweight gradient elevator), Decision Tree, Kernel Methods, kernel methods were selected. MLP-BP (Multi-layer perceptron-backpropagator) was used for comparative experiments. The relevant comparative prediction models are described as follows:

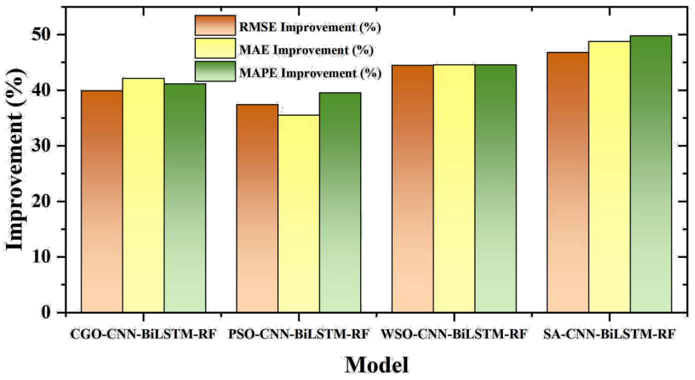


Figure 10 | INFO indicator promotion visualization

Note: The improvement of relevant indicators refers to the percentage improvement of INFO-CNN-BiLSTM-RF model compared with the comparison model

Then, under the condition that the parameters of all models are consistent with the experimental environment, the predicted results are shown in Table 6 and Figure 11.

The above experimental results show that the proposed model is significantly better than other traditional models in three key performance indexes. Compared with XGBoost, INFO-CNN-BiLSTM-RF decreased by 57.0% in RMSE, 57.8% in MAE and 58.1% in MAPE. Compared with LightGBM, RMSE decreased by 57.4%, MAE decreased by 58.7% and MAPE decreased by 59.2%. Compared with decision tree model, RMSE decreased by 59.9%, MAE decreased by 57.3% and MAPE decreased by 57.2%. Compared with the nuclear method, RMSE decreased by 84.5%, MAE decreased by 73.8%, and MAPE decreased by 72.6%, as shown in Figure 12. These results show that the INFO-CNN-BiLSTM-RF

Table 6 | Comparison of prediction performance of prediction models

Models/related indicators	RMSE	MAE	MAPE
INFO-CNN-BiLSTM-RF	75.011	54.294	0.627
XgBoost	173.552	128.889	1.495
LsBoost	175.976	131.377	1.538
DecisionTree	187.190	127.040	1.464
kernel	483.580	207.295	2.291

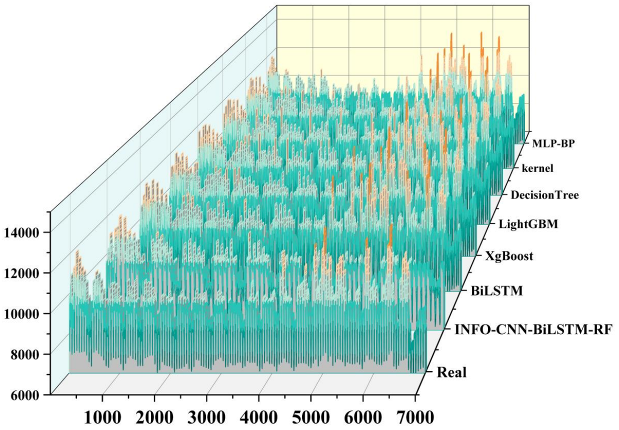


Figure 11 | Comparison of prediction effects of the proposed prediction algorithms

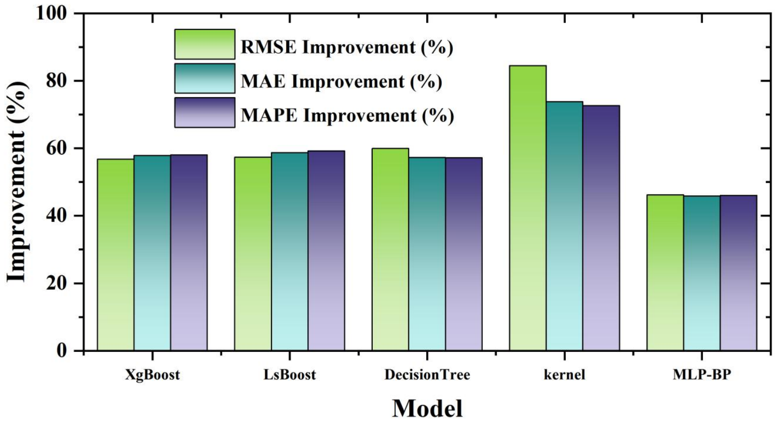


Figure 12 | Visualization of improved prediction effect of the model proposed in this paper

model has higher prediction accuracy and lower error when processing this dataset, demonstrating its obvious advantages over traditional models in complex time series prediction tasks. To sum up, it can be concluded that the model proposed in this paper is not only innovative in structure, but also has obvious advantages in accuracy, which indicates that the model proposed in this paper can be a powerful tool to deal with high-disturbance, high-complexity and high-noise time series prediction problems.

4.6. Optimize the Experimental Analysis of Module Ablation

In order to explore the effect of INFO-CNN-BiLSTM-RF related combined optimization modules on the overall performance of the model, a series of ablation experiments were conducted in this paper. Through these experiments, it is possible to analyze in detail the specific contribution of each component to the prediction accuracy, thus verifying the validity and necessity of the different modules in the INFO-CNN-BiLSTM-RF model. Specifically, we remove CNN, BiLSTM, and RF components from the model and observe how these changes affect the model's performance on the time series prediction task. By comparing the performance of the complete model and its variants on multiple key performance indicators, it is possible to reveal the specific impact of each optimization module on improving prediction accuracy, as well as their synergies within the overall architecture.

The specific ablation experiment results are shown in Table 7 and Figure 13.

The above experimental results show that. The INFO-CNN-BiLSTM-RF model significantly outperformed other models on all key performance indicators, demonstrating the importance of combining CNN, BiLSTM and RF components. Compared to models without any component, INFO-CNN-BiLSTM-RF showed significant performance improvements: RMSE, MAE and MAPE were reduced by 47.5%, 47.5% and 47.4%, respectively, compared to the CNN-BiLSTM-RF model. Compared with the RandomForest model, these indexes are improved by 51.4%, 51.6% and 51.6% respectively. Compared with CNN-BiLSTM model, the improvements are 48.1%, 48.5% and 48.8%, respectively. As shown in FIG. 14, in addition, compared with the CNN model, INFO-CNN-BiLSTM-RF increased by 75.4% on RMSE, 76.5% on MAE and 76.4% on MAPE. Compared with BiLSTM model alone, INFO-CNN-BiLSTM-RF improved 54.3% on RMSE, 54.4% on MAE,

Table 7 I Comparison of experimental predictive performance of ablation models

Models/related indicators	RMSE	MAE	MAPE
INFO-CNN-BiLSTM-RF	75.011	54.294	0.627
CNN-BiLSTM-RF	142.628	103.265	1.193
RandomFrest	154.480	112.169	1.296
CNN-BiLSTM	145.884	105.938	1.224
CNN	305.406	231.134	2.657
BiLSTM	163.98	119.14	1.382

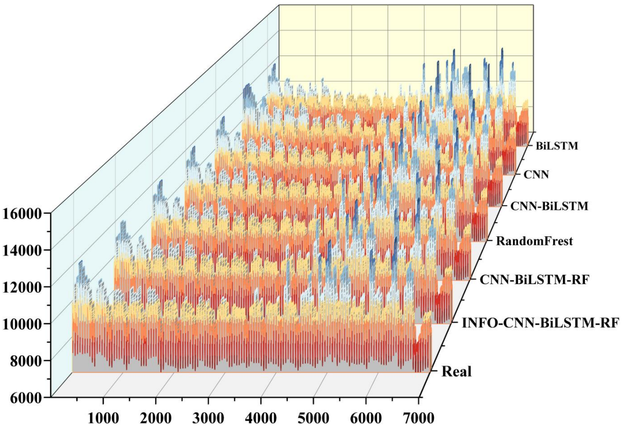


Figure 13 I Comparison of predicted results of ablation experiments

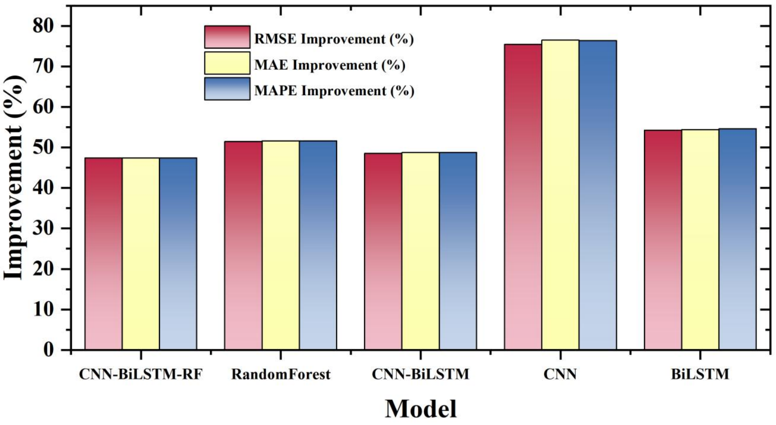


Figure 14 I Visualization of ablation experiment index improvement

and 54.6% on MAPE. In summary, by integrating CNN, BiLSTM and RF, the model can process time series data more comprehensively, and effectively extract and utilize spatial features and time dependence in data. In addition, by integrating the unique advantages of different algorithms, the integrated model significantly improves the accuracy and robustness of the prediction, and ADAPTS to the complex needs of time series prediction. The experimental results also show the effectiveness and necessity of adopting ensemble learning strategy in processing time series data with complex patterns.

5. Conclusion and Prospect

Aiming at the complexity and high dimensional data of power load forecasting, an ICEEMDAN-INFO-CNN-BiLSTM-RF model based on ensemble learning is proposed in this paper. First, ICEEMDAN (Improved Complete Set Empirical Mode Decomposition) is introduced in data processing to deal with nonlinear and non-stationary time series data more efficiently, thus enhancing the ability of the model to capture key features. Subsequently, the model integrated convolutional neural networks (CNN), two-way long short-term memory networks (BiLSTM), and random forests (RF) to improve prediction accuracy and robustness. By processing different data features in parallel, the CNN layer is able to capture spatial dependencies, while the BiLSTM layer focuses on capturing temporal dependencies of time series data. As the final decision level, random forest integrates multiple decision trees to improve the generalization ability and accuracy of the model. At the same time, the model also introduces the INFO (vector weighted average) optimization algorithm to optimize the weight configuration and parameter selection of the model and optimize the network hyperparameters, so as to improve the prediction accuracy and response speed of the whole model. The experimental results show that ICEEMDAN-INFO-CNN-BiLSTM-RF model has significantly improved performance compared with the traditional model in several key performance indicators. Meanwhile, under the same training parameters and conditions, the hyperparameter optimization ability of ICEEMDAN-Info-CNN-BilSTM-RF model is also superior to the traditional optimization algorithm. However, although the prediction accuracy of the proposed model is high, the computational cost in the process of model training and parameter optimization is large, especially in the dynamic adjustment of model weights and the integration of multiple models. Future research will need to explore more efficient algorithms to reduce training time and improve the operational efficiency of models, while maintaining or improving the predictive accuracy of existing models. In addition, the scalability and adaptability of the model in practical applications are also important directions for further research.

Reference

1. Fan, S., Li, L., Wang, S., et al. (2020). Application of artificial intelligence technology in power grid regulation. Power Grid Technology, 44(02), 401-411.
2. Chang, Z., & Xu, Y. (2024). Optimization of BiLSTM short-term power load forecasting based on CEEMDAN and INGO. Control Engineering. Advance online publication.
3. Liu, Q., Hu, Q., Yang, L., et al. (2021). Research on deep learning photovoltaic power generation model based on time series. Power System Protection and Control, 49(19), 87-98.
4. Li, B., Lu, M., Zhang, Y., et al. (2019). A weekend load forecasting model based on semi-parametric regression analysis considering weather and load interaction. Energies, 12(20), 3820.
5. Pappas, S. S., Ekonomou, L., Karamousantas, D. C., et al. (2008). Electricity demand loads modeling using auto regressive moving average (ARMA) models. Energy, 33(9), 1353-1360.
6. Lee, C. M., & Ko, C. N. (2011). Short-term load forecasting using lifting scheme and ARIMA models. Expert Systems with Applications, 38(5), 5902-5911.
7. Wu, D., Ma, W., & Yang, L. (2023). Electrical load forecasting with quadratic exponential smoothing multi-objective combination model. Computer Engineering and Design, 44(8), 2541-2547.
8. Wei, M., Zhou, Q., Cai, S., et al. (2020). Medium and long term load forecasting based on fractional order grey model optimized by BFGS-FA. Journal of Guangxi University (Natural Science Edition), 45(2), 270-276.
9. Bu, F., Bai, H., Wang, Y., et al. (2023). Residential electricity consumption analysis and load forecasting based on human comfort index. Chinese Journal of Test and Analysis, 49(4), 85-91.
10. Liu, Y., Peng, X., & Zheng, S. (2021). Research on short-term power load forecasting method based on improved LS-SVM. Electrical Measurement and Instrumentation, 58(5), 176-181.
11. Gong, Y., & Teng, H. (2019). Short-term load forecasting based on GOA-SVM. Electrical Measurement and Instrumentation, 56(14), 12-16.
12. Quinlan, R. J. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.
13. Yao, H., Li, C., Zheng, X., et al. (2022). Short-term load combination forecasting model with adaptive chirped mode decomposition and BiLSTM. Power System Protection and Control, 50(19), 58-66.
14. Yang, D., Yang, J., Hu, C., et al. (2021). Short-period power load forecasting based on improved LSSVM. Electronic Measurement Technology, 44(18), 47-53.
15. Li, Y., Jia, Y., Li, L., et al. (2020). Short-term power load forecasting based on random forest algorithm. Power System Protection and Control, 48(21), 117-124.
16. Jiang, L., Wang, X., Li, W., et al. (2021). Hybrid multitask multi-information fusion deep learning for household short-term load forecasting. IEEE Transactions on Smart Grid, 12(6), 5362-5372.
17. Gu, J., Shao, L., Lu, C., et al. (2023). Research on short-term power load forecasting of distribution station area based on LSTNet model. Electric Drive, 53(5), 63-70.
18. Li, G., Liu, Z., Jin, G., et al. (2020). Ultra-short term power load forecasting based on random distributed embedded frame and BP neural network. Power Grid Technology, 44(2), 437-445.
19. Yang, S., Wang, T., Tan, X., et al. (2023). Graphical short-period power load forecasting method based on LSTM. Global Energy Internet. Advance online publication.
20. Zhang, J., Ji, Y., & Chen, L. (2019). Application of deep learning in power load forecasting. Automaticity Instrumentation, 40(08), 8-12.
21. Siami-Namini, S., Tavakoli, N., & Namin, A. (2021). A comparative analysis of forecasting financial time series using ARIMA, LSTM, and BiLSTM. arXiv preprint arXiv:2107.10683.
22. Yang, L., Wu, H., Ding, M., et al. (2021). Short-term load forecasting of Bi-LSTM network considering feature selection in new energy grid. Automation of Electric Power Systems, 45(03), 166-173.
23. Liu, Z., & Yang, J. (2022). Research on short-term load forecasting based on GWO-BiLSTM. Journal of Physics: Conference Series, 2290(1).
24. Li, B., & Pan, H. (2023). Global temperature prediction by BiLSTM model based on whale optimization algorithm and attention mechanism. In 2023 3rd International Conference on Neural Networks, Information and Communication Engineering (NNICE) (pp. 651-657). IEEE.
25. Wei, A., Mao, D., Han, W., et al. (2020). Research on short-term power load forecasting based on EMD and long short-term memory network. Thermal Energy and Power Engineering, 35(04), 203-209.
26. López, C., Wei, Z., & Zheng, M. L. (2017). Short-term electric load forecasting based on wavelet neural network, particle swarm optimization and ensemble empirical mode decomposition. Energy Procedia, 105, 3677-3682.
27. Torres, M. E., Colominas, M. A., Schlotthauer, G., et al. (2011). A complete ensemble empirical mode decomposition with adaptive noise. In 2011 IEEE International

Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4144-4147). IEEE.

28. Colominas, M. A., Schlotthauer, G., & Torres, M. E. (2014). Improved complete ensemble EMD: A suitable tool for biomedical signal processing. Biomedical Signal Processing and Control, 14, 19-29.
29. Xu, Y., Wu, Z., Zhu, H., et al. (2020). Short-term power load forecasting based on multi-scale convolutional neural networks. Journal of Shenyang University of Technology, 42(6), 618-623.
30. Tudose, A. M., Sidea, D. O., Picioroaga, I. I., et al. (2020). A CNN based model for short-term load forecasting: A real case study on the Romanian power system. In 2020 55th International Universities Power Engineering Conference (UPEC) (pp. 1-6). IEEE.
31. Li, W., Zhang, P., Shi, Q., et al. (2012). Load correction prediction of integrated energy system based on aggregated hybrid mode decomposition and sequential convolutional neural network. Power Grid Technology, 46(9), 3345-3357.
32. Dai, Y., Zhou, Q., Leng, M., et al. (2022). Improving the Bi-LSTM model with XGBoost and attention mechanism: A combined approach for short-term power load prediction. Applied Soft Computing, 130, 109632.
33. Li, F., Li, P., Gao, L., et al. (2019). Short-term power load forecasting method based on multi-scale model fusion and VMD-TCN-RF hybrid network. Electronic Devices, 46(04), 1035-1042.
34. Li, D., Yang, P., Lian, J., et al. (2023). Based on the improved new extension cloud computer INFO algorithm performance evaluation model. Computer Application Research, 40(12), 3614-362