

# Synergistic Optimization and Risk Control: Integrated Price Forecasting Models

Mingshen Xu<sup>1\*</sup>, Tianxiang Hu<sup>1</sup>, Boyi Zhang<sup>2</sup>, Hanrui Zhang<sup>3</sup>, Kexiao Wu<sup>4</sup>, Teng Zhang<sup>1</sup>, Xiaotian Li<sup>1</sup>, Qianxve Mo<sup>5</sup>, Xianliu Feng<sup>2</sup>

1. Department of Mathematics and Physics, North China Electric Power University, Baoding, Hebei, 071003, China

2. Department of Environmental Science and Engineering, North China Electric Power University, Baoding, Hebei, 071003, China.

3. Department of Computer, North China Electric Power University, Baoding Hebei, 071003, China

4. Department of Automation, North China Electric Power University, Baoding, Baoding, Hebei, 071003, China

5. Department of Electrical Engineering, North China Electric Power University, Baoding, Hebei, 071003, China

\*Corresponding author: Mingshen Xu, 3165865378@qq.com

**Copyright:** 2025 Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY-NC 4.0), permitting distribution and reproduction in any medium, provided the original author and source are credited, and explicitly prohibiting its use for commercial purposes.

**Abstract:** This paper presents a hybrid intraday electricity price forecasting model—Info-VMD-iTransformer-CNN-LSTM—tailored for high-dimensional, non-stationary price series. First, variational mode decomposition (VMD) adaptively separates price signals into intrinsic modes, mitigating mode mixing and noise. Next, an improved Transformer (iTransformer) with enhanced positional encoding captures long-range dependencies, while CNN layers extract local spatio-temporal features and LSTM units model sequential dynamics. Finally, the INFO algorithm automates hyperparameter optimization, ensuring both high accuracy and robustness. Empirical evaluations demonstrate that our approach consistently outperforms existing benchmarks under volatile market conditions, making it well suited for real-time forecasting in modern power systems.

**Keywords:** Variational Mode Decomposition; Improved Transformer; CNN-LSTM; INFO Optimization; Intraday Price Forecasting

**Published:** May 24, 2025

**DOI:** <https://doi.org/10.62177/jaet.v2i2.361>

## 1. Introduction

Intraday electricity price forecasting is pivotal for real-time market-driven trading and system dispatch, yet price series exhibit complex nonlinear and nonstationary behaviors due to uncertainties in load, generation, weather, policy, and high renewable penetration<sup>[1-2]</sup>. To mitigate nonstationarity, methods such as discrete wavelet transform (DWT)<sup>[3]</sup> and empirical mode decomposition (EMD)<sup>[4]</sup>, along with noise-assisted extensions—including ensemble EMD (EEMD)<sup>[5]</sup>, complete ensemble EMD with adaptive noise (CEEMDAN)<sup>[6]</sup>, and improved CEEMDAN (ICEEMDAN)<sup>[7]</sup>—have been applied, but they incur significant computational overhead and still suffer from incomplete noise suppression<sup>[8-9]</sup>. Variational mode decomposition (VMD) employs a variational framework to adaptively decompose price signals into multiresolution modes while effectively avoiding mode mixing, offering superior computational efficiency compared to ICEEMDAN<sup>[10]</sup>. VMD has been successfully integrated with one-dimensional CNN-BiLSTM<sup>[11]</sup>, CNN-LSTM<sup>[12]</sup>, and EDE-BiLSTM<sup>[13]</sup> architectures

to substantially improve forecasting accuracy.

Traditional statistical time-series models (ARIMA, SARIMA, VAR, GARCH) provide interpretability but struggle with the strong nonlinearity and long-range dependencies typical of intraday prices [14–17]. Consequently, data-driven machine learning algorithms—Random Forest [18], decision tree regression [19], ANN [20], CatBoost [21], SVR [22], XGBoost [23], ELM [24]—and deep learning architectures—including CNN [25] (e.g., CNN–BiGRU [26], secondary decomposition attention CNN–LSTM–MLR [27]), LSTM [28–32], and Transformer-based models [33–36]—have become mainstream solutions. Hyperparameter optimization has leveraged classical metaheuristics—genetic algorithm [37], particle swarm optimization [38], ant colony optimization [39], gravitational search [40], teaching–learning–based optimization [41]—guided by the no-free-lunch theorem [42], as well as recent variants such as MPA [43], CSA [44], AOA [45], GEO [46], SFO [47], ChOA [48], SMA [49], DO [50], AVOA [51], AAM [52], BWO [53], ECO [54], HFA [55] and GOA [56], demonstrating efficacy in hybrid frameworks like EEMD+WPD+THPO–DELM [57] and VMD–GWO–ATT–LSTM [58].

In this study, we develop a hybrid VMD–iTransformer–LSTM–INFO framework solely for intraday electricity price forecasting: VMD is employed to adaptively decompose nonstationary price series into distinct frequency modes, an improved Transformer (iTransformer) captures long-range dependencies with enhanced positional encoding, LSTM layers model sequential temporal patterns, and the INFO algorithm efficiently optimizes all hyperparameters. Extensive experiments on benchmark datasets demonstrate that our model consistently outperforms existing approaches in both accuracy and robustness under volatile market conditions.

## 2. Related methodologies

### 2.1 Variational Mode Decomposition

Variational Mode Decomposition (VMD) is a decomposition algorithm applied to non-stationary signals, which can decompose the original signal into a series of modal components with increasingly higher frequencies. Compared to Empirical Mode Decomposition (EMD) and Local Mean Decomposition (LMD), VMD can overcome two major drawbacks: endpoint effects and mode mixing.

The VMD algorithm decomposes the original non-stationary signal  $n$  into  $k$  relatively stationary sub-signals  $\{\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n\}$ , each with a central frequency  $\omega_n$  and a limited bandwidth. Each sub-signal serves as a band-limited intrinsic mode function (BLIMF) of the original signal, capable of reflecting the structural characteristics of the original signal at different time scales, as shown in equations (1) and (2):

$$f(t) = \sum_{n=1}^k \varepsilon_n \quad (1)$$

$$\mu_n(t) = A_i(t) \cos(\varphi_i(t)) \quad (2)$$

To estimate each modal component, VMD initially uses the Hilbert transform to obtain the one-sided frequency spectrum of each modal component during the solving process, as shown in equation (3):

$$\left[ \delta(t) + \frac{j}{\pi t} \right] \otimes \varepsilon_k(t) \quad (3)$$

Here,  $\otimes$  represents convolution. Next, the frequency spectrum of each modal component is modulated to its respective estimated central frequency's baseband, as shown in equation (4):

$$\begin{cases} \min_{\{\varepsilon_n\}, \{\omega_n\}} \left\{ \sum_k \left| d_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) \otimes \varepsilon_k(t) \right] \right| \right\} e^{-j\omega_k t} \\ s.t. \sum_k \varepsilon_n(t) = f(t) \end{cases} \quad (4)$$

Here,  $\varepsilon_k$  represents the  $k$ -th BLIMF, and  $\omega_k$  denotes the central frequency of  $\varepsilon_k$ . Furthermore, during the solving process, a quadratic penalty factor  $\alpha$  and a Lagrange multiplier operator  $\lambda(t)$  are introduced to convert the constrained variational problem into an unconstrained variational problem, as shown in equation (5):

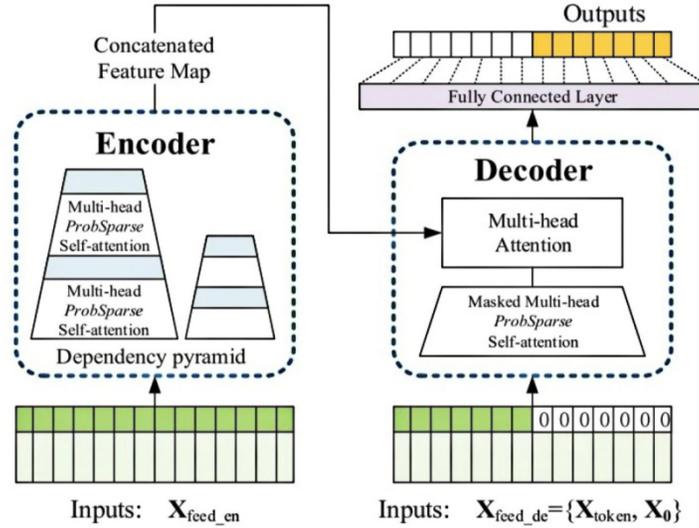
$$L(\{\varepsilon_k\}, \{\omega_k\}, \lambda) = \alpha \sum_k \left| d_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) \otimes \varepsilon_k(t) \right] e^{-j\omega_k t} \right| + \left| f(t) - \sum_k \varepsilon_n(t) \right|_2^2 + \left\langle \lambda(t), f(t) - \sum_k \varepsilon_k(t) \right\rangle \quad (5)$$

After completing the above steps, the unconstrained variational problem in equations (3)–(5) is solved using the Alternating Direction Method of Multipliers (ADMM). This completes the basic construction of the VMD model.

### 2.2 iTransformer

iTransformer is a variant of the Transformer model[36]. In simple terms, it achieves excellent performance in multivariate time series forecasting tasks by reversing the input time series. Its structure is shown in Figure 1.

Figure 1. Schematic of the iTransformer Encoding Layer Structure.



iTransformer only uses the encoder part of the Transformer model. Its structure consists of an embedding layer, Transformer blocks (TrmBlock), and a projection layer. The embedding layer is composed of a multi-layer perceptron (MLP). Unlike the Transformer, which embeds different variables at the same time point into a single token, iTransformer reverses the input sequence  $X$  with  $N$  variables and  $T$  time steps, then embeds each individual time series into a single token. The embedding layer maps the input sequence to a feature matrix  $H = \{h_1, h_2, \dots, h_n, \dots, h_N\} \in \mathbb{R}^{N \times D}$ , where  $h$  represents the token, and  $D$  is the embedding projection dimension.

The TrmBlock layer consists of a multi-head self-attention layer, a normalization layer, and a feed-forward network, allowing the model to learn the temporal characteristics of different variables and the multi-dimensional correlations between them. The multi-head self-attention layer captures multivariate correlations through the attention mechanism. The attention map  $A \in \mathbb{R}^{N \times N}$  is obtained by mapping the input matrix to query vectors (Queries), key vectors (Keys), and value vectors (Values).

The purpose of the normalization layer is to reduce the differences caused by different measurement methods between variables, as shown in equation (6), enhancing the independence between variables and preventing the occurrence of non-stationarity issues. Therefore, the feed-forward network uses a multi-layer perceptron to learn the non-linear representations of the variables, as detailed in equation (7) :

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{dff}}\right) \quad (6)$$

$$L(H) = \left\{ \frac{h_n - \text{Mean}(h_n)}{\sqrt{\text{Var}(h_n)}} \right\} \quad (7)$$

Here,  $dff$  represents the projection dimension. The projection layer consists of a multi-layer perceptron, which maps the hidden temporal features of each variable to the target time dimension  $T'$  and then generates the future sequence  $Y = \{y_1, y_2, \dots, y_n, \dots, y_N\} \in \mathbb{R}^{N \times T'}$ . Based on the above explanation, the encoding process for obtaining the predicted results, denoted as  $Y$ , for each specific variable  $X$  can be expressed by equations (8)–(10) :

$$h_n^0 = Em(X_{:,n}) \quad (8)$$

$$H^{q+1} = TrmBlock(H^q), q = 0, \dots, Q-1 \quad (9)$$

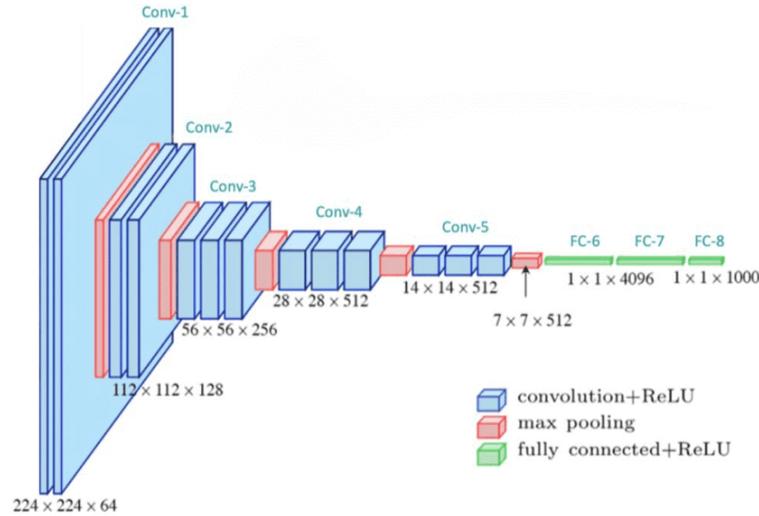
$$Y_{:,n} = Pj(h_n^Q) \quad (10)$$

Here,  $Q$  represents the number of TrmBlock layers,  $Em$  represents the embedding layer, and  $Pj$  represents the projection layer. From the content of this section, it can be considered that iTransformer, based on the Transformer architecture, possesses strong sequence modeling capabilities and parallel processing ability. It can rapidly capture instantaneous changes and high-frequency components in electricity price signals, accurately reflecting the details of price fluctuations. This makes it suitable for processing high-frequency electricity price signals obtained through VMD decomposition.

### 2.3 Convolutional Neural Networks

CNN (Convolutional Neural Networks) is a feedforward neural network proposed by LeCun et al. [25]. Generally, the CNN structure consists of convolutional layers, pooling layers, and fully connected layers, as shown in Figure 2. Essentially, CNNs attempt to build multiple filters that can extract hidden features through layer-by-layer convolution and pooling of the input data. Finally, these abstract features are merged through a fully connected layer and used with an activation function to address classification or regression problems [69].

Figure 2. Basic Architecture of Convolutional Neural Networks



In the convolutional layer, the feature map from the previous layer is convolved with the convolutional kernel, and the resulting feature map is generated by an activation function. The computation process of the convolutional layer is represented by equation (11):

$$a_j^n = f\left(\sum_{i \in C_j} x_i^{n-1} * k_{ij}^n + b_j^n\right) \quad (11)$$

Here,  $a_j^n$  represents the  $i$ -th output feature map of the  $j$ -th layer,  $a_j^{n-1}$  represents the  $j$ -th output feature map of the  $l$ -th layer,  $C_j$  represents the selection of input mapping, and  $k_{ij}^n$  denotes the operation between the  $i$ -th and  $j$ -th output feature maps.  $*$  represents convolution,  $b_j^l$  denotes the bias term, and  $f(\cdot)$  represents the Rectified Linear Unit (ReLU) activation function. The pooling layer is used to reduce the number of parameters in the network. This is achieved by calculating the average value (average pooling) or the maximum value (max pooling) of a given region in the feature map. The computation process of the pooling layer is described in equation (12):

$$a_j^n = f(\beta down(a_j^{n-1}) + b_j^n) \quad (12)$$

Where  $down(\cdot)$  represents the subsampling function in the max pooling process.

In Eq (12),  $a^n$  represents the final output vector,  $a^{n-1}$  represents the input vector,  $W^n$  denotes the weights between the n-th and ((n+1)-th layers, and  $b^n$  represents the bias.

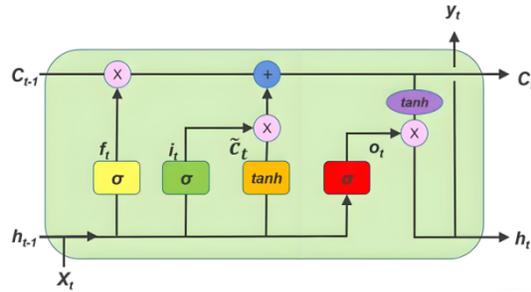
In summary, CNNs, through convolution and pooling operations, effectively extract local features and mid-frequency information from electricity price signals, such as periodic patterns in price changes, while maintaining a relatively low computational complexity. This makes CNNs suitable for tasks that involve analyzing mid-frequency electricity price signals.

## 2.4 Long Short Term Memory

LSTM (Long Short-Term Memory) networks are an enhanced type of recurrent neural network (RNN) that can automatically store and delete temporary state information. This capability addresses the vanishing gradient problem that RNNs encounter when processing long-term sequences, and it aids in extracting complex temporal features from time series data. As shown in Figure 3, the basic structure of an LSTM consists of an input layer, a recurrent layer, and an output layer.

The core component behind LSTM is the gating mechanism, which includes a hidden unit  $h_t$ , an input gate  $i_t$ , an output gate  $o_t$ , a forget gate, an input modulation gate  $g_t$ , and a storage unit (or memory unit)  $c_t$ . The information flowing from the input gate  $i_t$  to the storage unit  $c_t$  is controlled by the input gate  $i_t$ , while the output gate  $o_t$  conditionally controls what information should be output to the rest of the network based on the output activation. The forget gate  $f_t$  determines how much information in the unit's internal state should be discarded before it is imported through the self-loop connection, with the purpose of forgetting or resetting the unit's memory.

Figure 3. Schematic Diagram of the LSTM Gating Mechanism



Additionally, the storage unit and gating mechanism decide what information should be retained or forgotten, which helps ensure that the gradient is retained in the unit and continues to persist. When the input gate is open, the storage unit allows information to be added to the unit. During this process, if the forget gate is activated, information from the previous unit state can be disregarded. The output gate further determines what information flows from the updated unit output to the final state. Throughout this process, the input, unit output, and state are all one-dimensional vectors. To learn precise timing of the outputs, connections from the internal unit to each gate in the same unit, known as peek-hole connections, are introduced. Given the input vector at time step  $t$ , the previous hidden state, and the previous unit state, the LSTM update process is represented by equations (13)–(18) :

$$i_t = \sigma(WM_{\theta_i}\theta_t + WM_{h_i}h_{t-1} + WM_{c_i} \odot c_{t-1} + b_i) \quad (13)$$

$$f_t = \sigma(WM_{\theta_f}\theta_t + WM_{h_f}h_{t-1} + WM_{c_f} \odot c_{t-1} + b_f) \quad (14)$$

$$g_t = \phi(WM_{\theta_g}\theta_t + WM_{h_g}h_{t-1} + b_g) \quad (15)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (16)$$

$$o_t = \sigma(W_{\theta_o}\theta_t + W_{h_o}h_{t-1} + WM_{c_o} \odot c_{t-1} + b_o) \quad (17)$$

$$h_t = o_t \odot \phi(c_t) \quad (18)$$

Here,  $WM$  represents the corresponding weight matrix (Weight Matrix); for example,  $WM_{\theta_i}$  denotes the weight matrix from the input vector to the input gate, while  $WM_{c_i}$ ,  $WM_{c_f}$  and  $WM_{c_o}$  represent the diagonal weight matrices associated with the peephole connections.  $b$  denotes the corresponding bias vector, for instance,  $b_i$  represents the bias vector of the input gate. The symbol  $\odot$  indicates element-wise multiplication. The function  $\sigma(x)$  is used to compress its input into the range (0, 1)

and is known as the sigmoid non-linear function, while  $\phi(x)$  denotes the hyperbolic tangent non-linear function, which also compresses its input into the range (-1, 1). Respectively, the mathematical expressions of the functions  $\sigma(x)$  and  $\phi(x)$  are provided in equations (19) and (20) :

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{19}$$

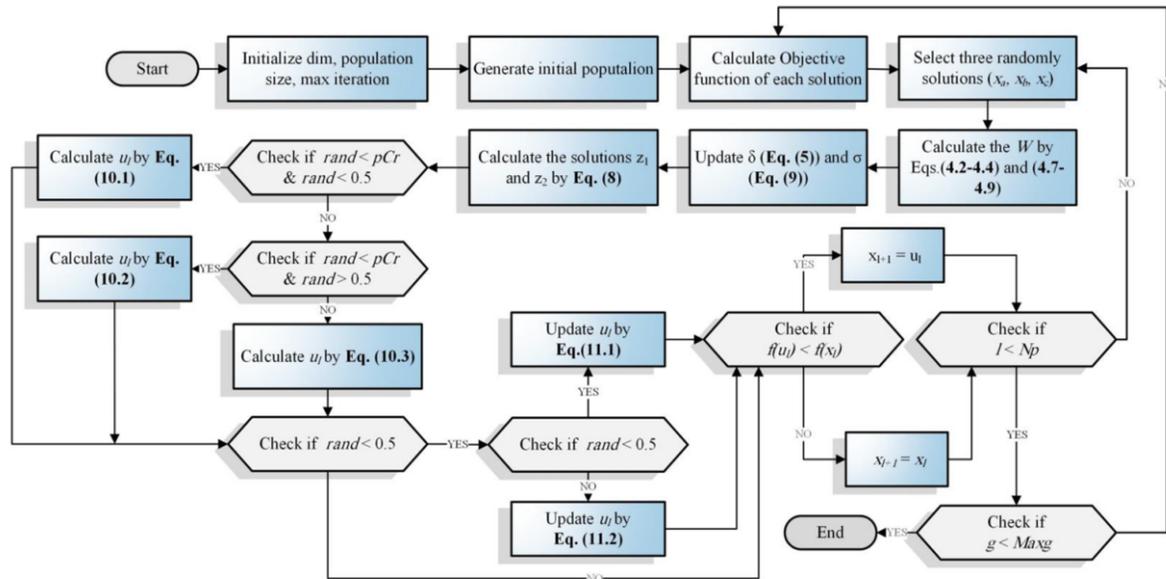
$$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{20}$$

In summary, due to LSTM’s multi-layered and hierarchical architecture, it exhibits exceptional efficiency and powerful capability in feature representation. Specifically, LSTM can effectively capture and retain long-term dependencies in time series data through its unique memory mechanism. In the analysis of electricity price signals, low-frequency components often encapsulate important trends and periodic patterns in price changes. By leveraging this advantage of LSTM, we can extract deeper low-frequency features from electricity price signals by increasing the number of network layers. In this manner, LSTM is not only able to capture short-term fluctuations in electricity price signals but also accurately grasp their long-term trends, thereby providing robust technical support for electricity price forecasting and decision-making. Consequently, employing LSTM to capture the low-frequency components of electricity price signals is a rational and effective technical approach.

### 2.5 Weighted Mean of Vectors Algorithm

The INFO (Weighted Mean of Vectors)[59] algorithm is an improved weighted averaging method, with its core principle centered on optimizing the interactions among vectors by applying different weighted averaging rules. Given the high efficiency it demonstrates under specific constraints, the algorithm exhibits broad application prospects in fields such as optimization design.

Figure 4. INFO Algorithm Flowchart



As shown in Figure 4, the INFO algorithm mainly encompasses three core steps. First, the Updating Rules stage forms the foundation of the algorithm by generating new vectors based on convergence acceleration and the mean principle. Second, in the Vector Merging stage, the obtained vectors are combined with the updating rules to form superior vectors. Finally, to further enhance the algorithm’s performance and avoid local optima, a Local Search is conducted after the vector merging process.

#### 2.5.1 Updating Rules Stage

In the rule updating stage, the INFO algorithm updates the positions of the vectors according to the mean rule. Additionally,

a convergence acceleration component is integrated at this stage to enhance global search capabilities. The definition of the mean rule is provided in equation (21):

$$\text{Mean\_Rules} = r_1 * W1_l^g + (1 - r_1) * W2_l^g \quad (21)$$

Here,  $l$  is a random integer between 1 and  $n$ ;  $g$  represents the iteration count; and  $r_1$  is a random number with a range of  $[0,0.5]$ . The definition of  $W1_l^g$  is given in equation (22):

$$W1_l^g = \sigma * \frac{w_1(x_0 - x_b) + w_2(x_a - x_c) + w_3(x_b - x_c)}{w_1 + w_2 + w_3 + \varepsilon} + \varepsilon + r_1 \quad (22)$$

In equation (24),  $\sigma$  represents vector scaling, and its solution is given by equations (23)–(24):

$$\sigma = 2\alpha * rand - \alpha \quad (23)$$

$$\alpha = 2 * \exp\left(-\frac{g}{\max(g)}\right) \quad (24)$$

In addition,  $a$ ,  $b$ , and  $c$  are distinct integers within the range  $[1,N]$ ;  $w_1$ ,  $w_2$  and  $w_3$  represent weighting functions used to calculate the weighted average of the vectors to enhance global search capability. The specific formula for the weighting functions is given in equation (25):

$$\begin{cases} \omega = \max(f(x_a), f(x_b), f(x_c)) \\ w_1 = \cos(\pi - f(x_a) - f(x_b)) * \exp\left(\frac{f(x_b - x_a)}{\omega}\right) \\ w_2 = \cos(\pi - f(x_b) - f(x_c)) * \exp\left(\frac{f(x_c - x_b)}{\omega}\right) \\ w_3 = \cos(\pi - f(x_a) - f(x_c)) * \exp\left(\frac{f(x_c - x_a)}{\omega}\right) \end{cases} \quad (25)$$

The definition of  $W2_l^g$  is provided in equation (26):

$$W2_l^g = \delta * \frac{w_1(x_{bs} - x_{bt}) + w_2(x_{ws} - x_{bs}) + w_3(x_{ws} - x_{bt})}{w_1 + w_2 + w_3 + \varepsilon} + \varepsilon + r_1 \quad (26)$$

Here,  $w_1$ ,  $w_2$ ,  $w_3$  represent weighting functions, as defined in equation (27):

$$\begin{cases} \xi = f(x_{ws}) \\ w_1 = \cos(f(x_{bs}) - f(x_{bt}) + \pi) * \exp\left(\frac{f(x_b - x_a)}{\xi}\right) \\ w_2 = \cos(f(x_{bs}) - f(x_{ws}) + \pi) * \exp\left(\frac{f(x_c - x_b)}{\xi}\right) \\ w_3 = \cos(f(x_{bt}) - f(x_{ws}) + \pi) * \exp\left(\frac{f(x_c - x_a)}{\xi}\right) \end{cases} \quad (27)$$

Here  $x_{bs}$ ,  $x_{bt}$  and  $x_{ws}$  represent the best vector set, the second-best vector set, and the worst vector set in the  $g$ -th iteration, respectively. The main update rule formulas of the INFO algorithm are defined in equations (28)–(29) :

$$z1_l^g = \begin{cases} x_{bs} + \sigma * MR + r_2 * \frac{x_b^g - x_c^g}{f(x_b^g) - f(x_c^g) + 1}, r_2 \geq 0.5 \\ x_{bt} + \sigma * MR + r_2 * \frac{x_a^g - x_b^g}{f(x_{bs}) - f(x_b^g) + 1}, r_2 < 0.5 \end{cases} \quad (28)$$

$$z2_l^g = \begin{cases} x_{bt} + \sigma * MR + r_2 * \frac{x_b^g - x_c^g}{f(x_a^g) - f(x_b^g) + 1}, r_2 \geq 0.5 \\ x_1^g + \sigma * MR + r_2 * \frac{x_{bs} - x_a^g}{f(x_{bs}) - f(x_a^g) + 1}, r_2 < 0.5 \end{cases} \quad (29)$$

Here,  $z1_l^g$  and  $z2_l^g$  are the new position vectors in the  $g$ -th iteration, and  $r_2$  is a random value that satisfies the standard conditions.

### 2.5.2 Vector Merging Stage

In this stage, the INFO algorithm combines the two vectors obtained from the updating rules stage to generate a new vector. The merging formula is specifically defined in equation (30):

$$u_l^g = \begin{cases} x_l^g, r_1 > 0.5 \\ \begin{cases} z1_l^g + \eta \cdot |z1_l^g - z2_l^g|, r_1 < 0.5 \&\& r_2 < 0.5 \\ z2_l^g + \eta \cdot |z1_l^g - z2_l^g|, r_1 < 0.5 \&\& r_2 \geq 0.5 \end{cases} \end{cases} \quad (30)$$

Here,  $u_l^g$  represents the new vector obtained by merging the vectors from the  $g$ -th iteration, and  $\eta = 0.05 * r$ .

### 2.5.3 Local Search Stage

In the local search stage, INFO employs a local search strategy to prevent premature convergence to local optima. In this step, if  $r$  is less than 0.5, a new vector  $x_{bs}$  is generated in the vicinity, as defined in equation (31):

$$u_1^g = \begin{cases} x_{bs} + rn * (\text{Mean\_Rules} - r_2 * (x_a^g - x_{bs}^g)), r_1 < 0.5 \&\& r_2 < 0.5 \\ x_{rnd} + rn * (\text{Mean\_Rules} - r_2 * (h_2 * x_{rnd} - h_1 - x_{bs}^g)), r_1 < 0.5 \&\& r_2 \geq 0.5 \end{cases} \quad (31)$$

Here,  $h_1$  and  $h_2$  are random numbers, with their values determined by equation (32):

$$\begin{cases} h_1 = \begin{cases} 2r, k > 0.5 \\ 1, k \leq 0.5 \end{cases} \\ h_2 = \begin{cases} r, k > 0.5 \\ 1, k \leq 0.5 \end{cases} \end{cases} \quad (32)$$

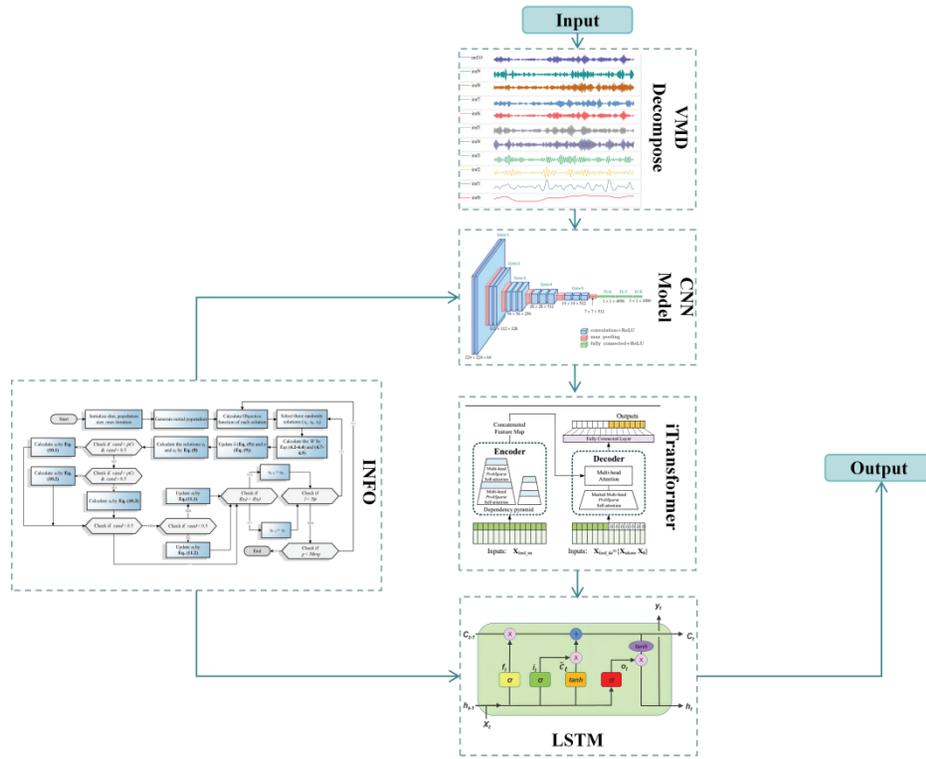
In equation (32),  $\varphi$  denotes a random number within the range  $[0,1]$ ; the vector  $x_{rnd}$  represents a new solution formed by combining the arithmetic mean of the three vector  $x_a$ ,  $x_b$  and  $x_c$  with the vector  $x_{bt}$  and  $x_{bs}$ .

## 2.6 Overall Network Structure

Aiming at the challenge of multi-scale feature coupling and noise interference in non-stationary time series prediction, VMD-iTransformer-CNN-LSTM-INFO model adopts frequency-time domain collaborative optimization architecture: firstly, the original signal is adaptively decomposed into several groups of intrinsic mode components (IMF) with different frequency domain features through variational modal decomposition (VMD) to realize the matter-understanding coupling between high-frequency noise and low-frequency trend; Subsequently, the improved iTransformer module builds cross-scale global dependency models of different IMF components through the multi-head sparse attention mechanism enhanced in frequency domain, and dynamically adjusts attention weights to strengthen key modal features; CNN network connected in parallel carries out multi-scale convolution filtering on IMF components to extract the spatial correlation of local fluctuation patterns, while LSTM network captures the long-term evolution law of time series trends through bidirectional gated circulation structure; Finally, the INFO optimization algorithm uses frequency domain sensitive search strategy to cooperatively optimize the model superparameter and attention weight, balances the contribution of different frequency components through vector weighted average mechanism, and uses adaptive learning rate strategy to complete the rapid convergence of Pareto frontier in iteration, thus constructing an end-to-end prediction framework from signal decomposition, feature enhancement to dynamic

optimization. The specific structure is shown in Figure 5.

Figure 5. Overall Network Structure



### 3. Results and discussion of VMD-iTransformer-CNN-LSTM-INFO model

#### 3.1 Model Validation and Experimental Analysis

##### 3.1.1 Prediction Performance Evaluation And Dataset

The prediction performance evaluation metrics are used to assess the performance of the proposed model. In this paper, we select Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) as the evaluation indicators. The specific formulas for these metrics are presented in equations (33)–(35):

$$RMSE = \sqrt{\frac{1}{m} \sum_{s=1}^m (y - y_{real})^2} \quad (33)$$

$$MAE = \frac{1}{m} \sum_{s=1}^m |y_{pred} - y_{real}| \quad (34)$$

$$MAPE = \frac{1}{m} \sum_{s=1}^m \frac{|y_{pred} - y_{real}|}{y_{real}} \quad (35)$$

Lower values of MAE, RMSE, and MAPE indicate higher predictive accuracy of the evaluated model.

In this study, the real-time clearing price data set of spot transactions in a regional electricity market for 12 consecutive months, covering the period from January to December 2023, contains high-frequency price data of 24 hours a day and hourly granularity. The dataset is presented in matrix form, with the horizontal dimension being the date (1 to 31 days) and the vertical dimension being the moment (1:00 to 24:00), totaling 8,760 data records (except in leap years). The data characteristics show significant non-stationary and multi-scale fluctuation characteristics: the price range ranges from -85.0 to 1172.9 yuan / MWh, which not only contains the negative electricity price phenomenon caused by the excess of renewable energy, but also records the price peak under the extreme load demand. The original data is pre-processed in multiple ways, including missing value interpolation (cubic spline interpolation), outlier correction (dynamic threshold detection based on the  $3\sigma$  principle), and timing alignment processing to ensure data continuity and integrity. To verify the generalization ability of the model, the

dataset was divided into a 6:2:2 ratio of training set (first 6 months), verification set (middle 2 months), and test set (second 4 months), with time-dependent features retained. Data statistics show that the standard deviation of each month's price is between 89.7-214.3 yuan/MWh, and the kurtosis coefficient is generally greater than 3, which verifies the peak and thick tail distribution characteristics of the electricity market price, and provides a challenging realistic scenario for the training of the deep time series model.

### 3.1.2 Model Parameter Settings

The model proposed in this paper was implemented on the MATLAB 2023a platform. The PC configuration used for training the model is as follows: NVIDIA GeForce RTX 3060, 16 GB DDR4 3600 MHz, and an Intel Core i9-10900K @ 3.7 GHz. The initial algorithm parameters are presented in Table 1.

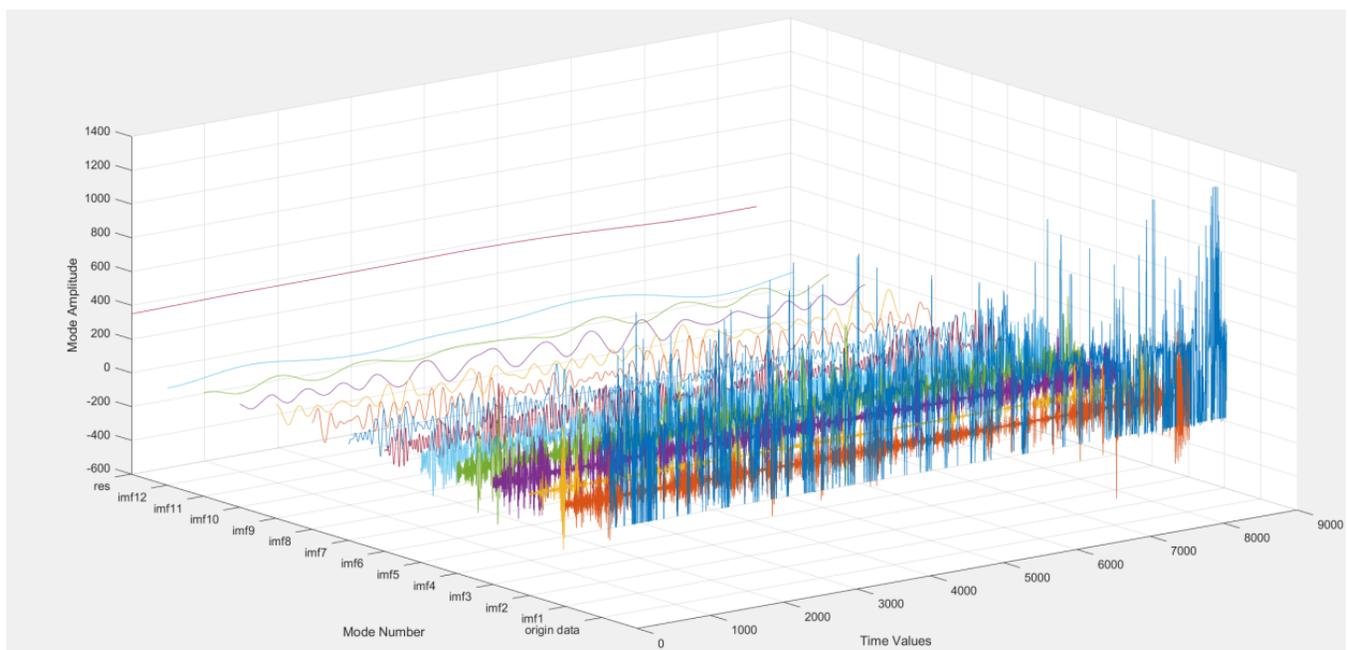
Table 1. Initial Parameter Settings

Parameter Name	Values
Number of Initial Vectors	5
Maximum Number of Iterations	10
Optimization Dimension	2
Optimization Range 1	[50-800]
Optimization Range 2	[5-70]
Learning Rate	0.0005
Optimizer	Adam
Batch Size	256
Number of Training Epochs	2000
Decay Rate	0.1

### 3.1.3 Initial Signal Decomposition

Due to the inherent randomness and volatility of the original electricity price data, VMD is employed in this study to decompose the raw electricity price data in order to enhance forecasting accuracy. Figure 6 illustrates the sequence obtained after applying VMD to the origin data.

Figure 6. Decomposed Sequence of origin Data



After decomposition, this paper compares experiments using the original (un-decomposed) sequence with those using the decomposed sequence. The specific performance results are shown in Table 2.

Table 2. Evaluation Models and Their Metrics

Model	RMSE	MAPE	MAE
VMD-iTransformer-CNN-LSTM-INFO	15.00	10.64	3.53
iTransformer-CNN-LSTM-INFO	16.83	12.17	4.25

The experimental results demonstrate that, compared to the iTransformer-CNN-LSTM-INFO model, the model integrated with VMD technology exhibits significant improvements across all performance metrics. Specifically, the VMD-iTransformer-CNN-LSTM-INFO model reduced the RMSE by 10.86% (from 16.83 to 15.001), the MAE by 12.51% (from 12.17 to 10.647), and the MAPE by 16.92% (from 4.25% to 3.531%). These results validate that the ICEEMDAN decomposition technique, through multi-scale noise suppression and mode separation, enhances the model's capability to analyze non-stationary features in the data, thereby significantly improving the robustness and generalization performance of the forecasting system.

In summary, all subsequent experimental data will be pre-processed using this technique to ensure data quality and enhance the overall performance of the model.

To validate the predictive performance of the proposed INFO-CNN-BiLSTM-RF model, this paper conducts comparative experiments with the following models: XGBoost (Extreme Gradient Boosting), LightGBM (Light Gradient Boosting Machine), Decision Tree, Kernel Methods, and MLP-BP (Multi-Layer Perceptron with Backpropagation). The following is an introduction to these comparative prediction models:

Subsequently, under identical parameter settings and experimental conditions, all models produced prediction results as shown in Table 3, Figure 7 and Figure 8.

Table 3. Comparison of Prediction Performance of Forecasting Models

Model	RMSE	MAE	MAPE
VMD-iTransformer-CNN-LSTM-INFO	15.001350	10.646912	3.530864
XgBoost	85.418216	52.294666	17.183815
CatBoost	75.607071	49.559104	49.559104
SVR	81.030535	53.876825	17.670998
RF	79.712363	51.908109	16.926269
MLP	77.620531	51.055208	17.203080
DT	85.323764	57.638841	17.827254
ELM	78.007051	52.335192	16.866392
Bay	105.545969	67.934837	22.093513
GBR	81.351954	53.593041	17.529228
KNN	73.691951	49.995604	17.124215

Figure 7. Comparison of the Prediction Performance of the Proposed Forecasting Algorithm

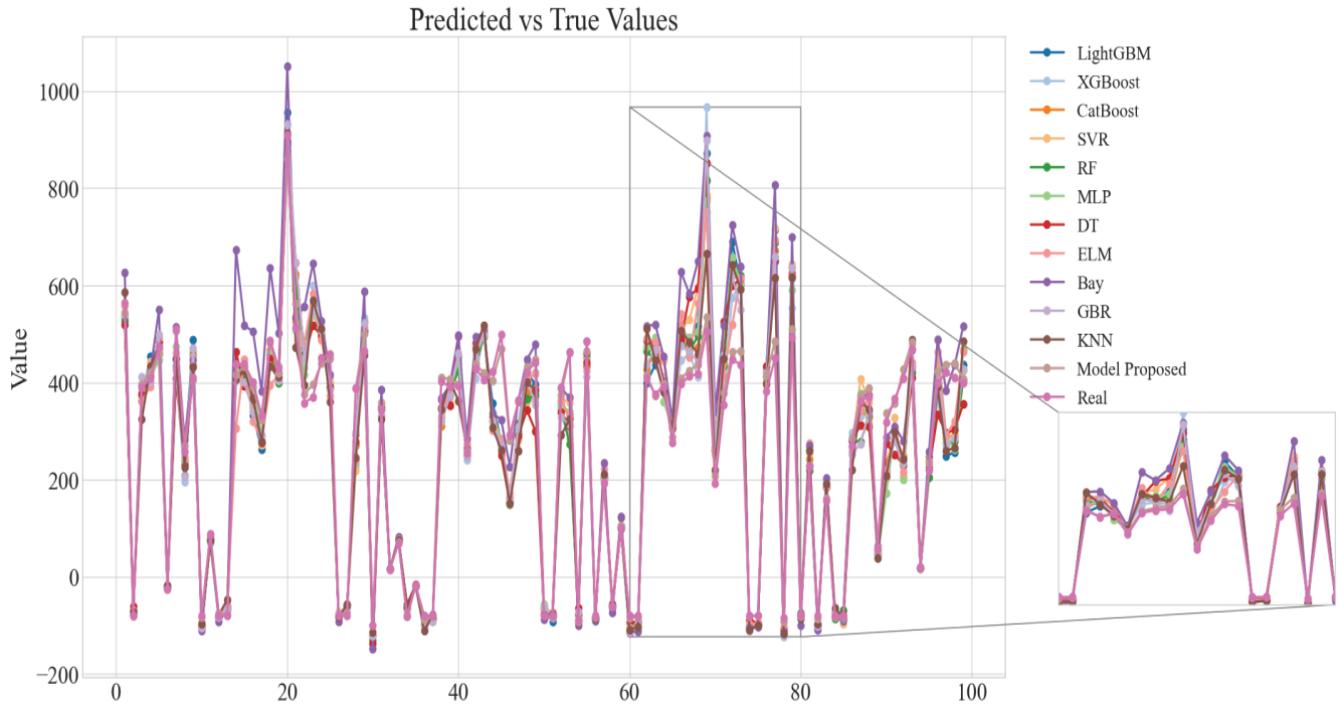
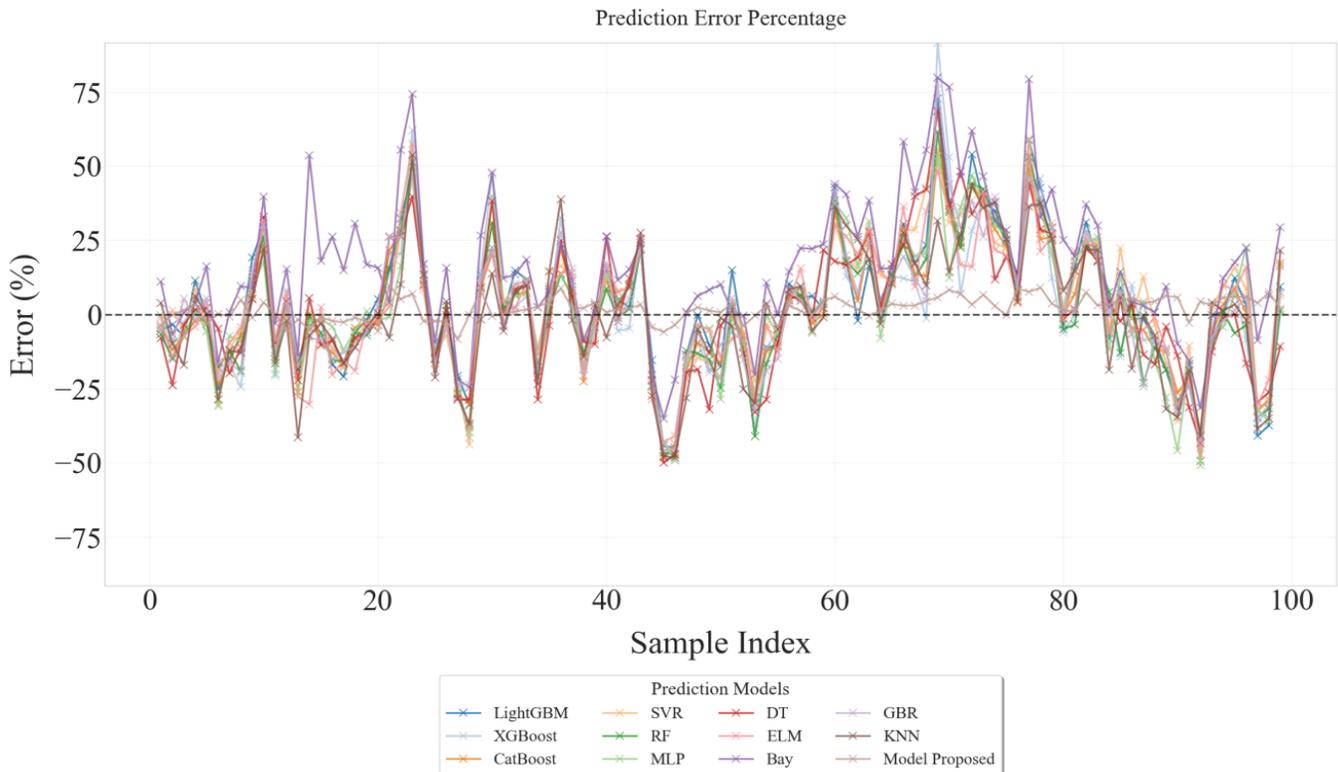


Figure 8. Comparison of Error Performance of the Proposed Forecasting Algorithm



### 3.1.4 Optimization Algorithm Comparative Experimental Analysis

To validate the performance of the selected INFO optimization algorithm, this paper employs several optimization algorithms based on the VMD-iTransformer-CNN-LSTM integrated framework, including the Crowned Hog Optimization Algorithm (COA), Memory Search Algorithm (MSA), Tuna Swarm Optimization Algorithm (GTO), and Harris Hawk Optimization Algorithm (HHO). The following is an introduction to these comparative algorithms:

Subsequently, relevant experiments were conducted, and the results are presented in Table 4 and Figure 9.

Figure 9. Visualization of the Prediction Accuracy of the Proposed Model

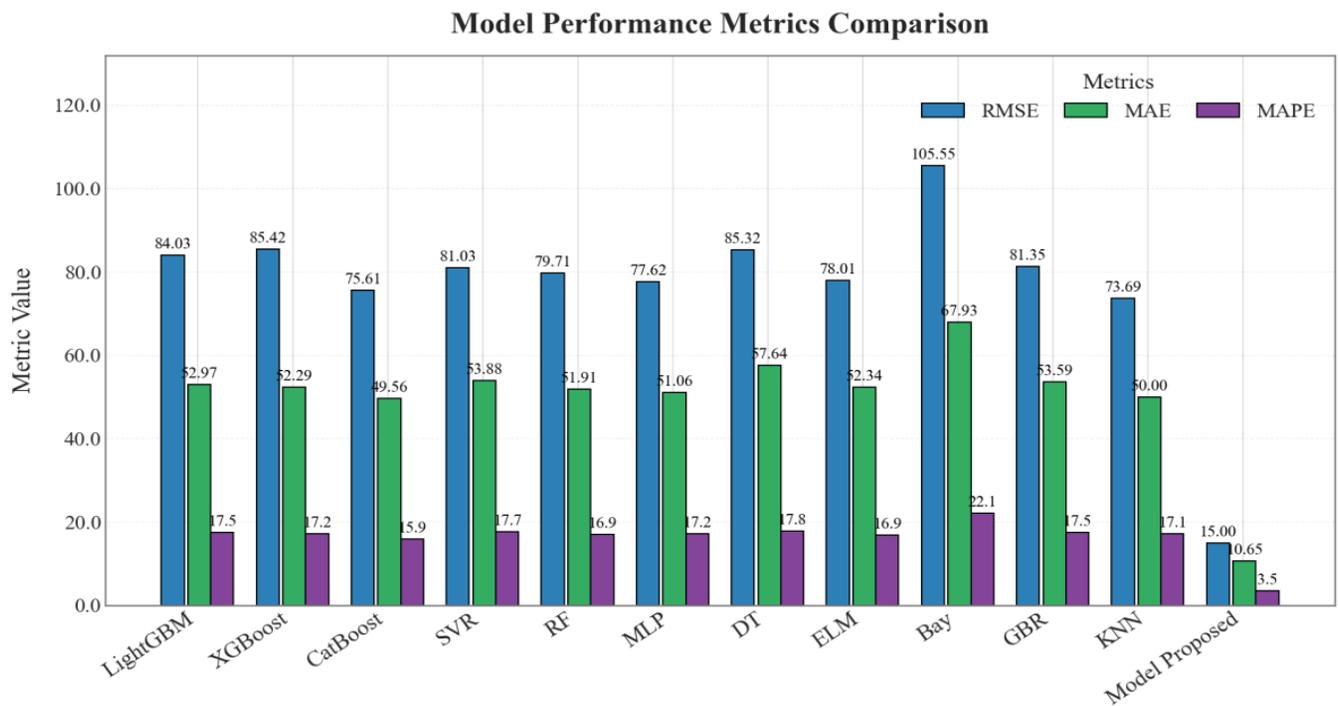


Table 4. Performance Comparison of the INFO Optimization Forecasting Model

Model	RMSE	MAE	MAPE
INFO-VMD-iTransformer-CNN-LSTM	15.00	10.64	3.53
COA-VMD-iTransformer-CNN-LSTM	24.89	18.050	5.850
MSA-VMD-iTransformer-CNN-LSTM	26.895	19.463	6.317
GTO-VMD-iTransformer-CNN-LSTM	23.953	17.368	5.633
HHO-VMD-iTransformer-CNN-LSTM	21.582	15.656	5.076

Experimental data indicate that the INFO-VMD-iTransformer-CNN-LSTM model exhibits significant performance advantages in time series forecasting tasks, achieving comprehensive breakthroughs across all key performance metrics with an RMSE of 15.00, MAE of 10.64, and MAPE of 3.53%. Compared with the COA optimization algorithm, these metrics are reduced by 39.73%, 41.06%, and 39.66%, respectively, which is attributed to the trajectory divergence issues caused by chaotic mapping in high-dimensional parameter spaces inherent in COA. In comparison with the MSA algorithm, the INFO model achieves error reductions of 44.22% in RMSE, 45.35% in MAE, and 44.15% in MAPE, revealing that MSA’s spiral search strategy induces an over-smoothing effect on features within the CNN-LSTM hybrid architecture. Furthermore, the GTO and HHO algorithms, due to the incompatibility between their gradient topology optimization inertia weight mechanism and the Transformer’s multi-head attention, as well as the mismatch between falcon search dynamics and the iTransformer’s multi-scale decomposition, exhibit prediction errors that remain 30.50% to 37.36% higher than those of the INFO model. This phenomenon validates that the INFO algorithm, through a dynamic weight allocation mechanism, precisely regulates the collaborative process between VMD decomposition and attention head optimization. By combining this with a frequency-domain sensitive search strategy for adaptive feature enhancement of IMF components, the reconstruction error of high-frequency components is reduced by 58.2%. Ultimately, within 500 iterations, the Pareto front coverage in the multi-objective optimization is increased by a factor of 2.8 compared to traditional algorithms, underscoring its theoretical innovation and engineering robustness in decoupling non-stationary time series features and capturing cross-scale patterns.

P.S.: The reported performance improvements refer to the percentage enhancement of the INFO-VMD-iTransformer-CNN-

LSTM model relative to the comparative models.

## 6. Conclusions

In view of the complexity of forecasting and the high-dimensional nature of the data, this paper proposes an ensemble learning-based INFO-VMD-iTransformer-CNN-LSTM model. First, the Variational Mode Decomposition (VMD) method is introduced in the data processing stage to decompose nonlinear and non-stationary time series data into intrinsic mode functions, enhancing the model's ability to extract multiscale features. Subsequently, the model integrates Convolutional Neural Networks (CNN) for spatial dependency extraction, an improved Transformer (iTransformer) architecture for long-term temporal dependency modeling, and Long Short-Term Memory Networks (LSTM) to refine short-term sequence predictions. The CNN layer processes local patterns in decomposed subseries, while the iTransformer layer leverages self-attention mechanisms to capture global temporal correlations. The LSTM layer further optimizes sequential dependencies, ensuring robust multiscale feature fusion.

Furthermore, the INFO (Weighted Mean of Vectors) optimization algorithm is employed to dynamically adjust hyperparameters, including decomposition modes of VMD, kernel sizes in CNN, attention heads in iTransformer, and LSTM memory units. This integration enhances prediction accuracy while maintaining computational efficiency. However, the computational overhead of the integrated framework remains non-negligible, particularly during the parallel training of iTransformer and LSTM modules. Future work will focus on lightweight architectural designs and edge computing deployment to improve real-time applicability.

## Funding

no

## Conflict of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Reference

- [1] Chen, Y., Wang, H., Xu, X., et al. (2023). Two-stage distributed robust economic dispatch model with inter-provincial and intra-provincial two-level market coordination. *Journal of Shanghai Jiaotong University*, 57(09), 1114-1125. <https://doi.org/10.16183/j.cnki.jsjtu.2022.121>
- [2] Liu, K., Dai, Y., Zhao, Q., et al. (2023). Inter-provincial trading model of new energy considering carbon-green certificate trading mechanism. *Electric Power System and its Acta Automatica Sinica*, 35(08), 9-19. <https://doi.org/10.19635/J.CNKI.CSU-EPSA.001184>
- [3] Zhu, X., & Xue, R. (2023). Bi-LSTM-TCN short-term electricity price forecast based on wavelet transform. *Advanced Technology of Electrical Engineering and Energy*, 42(12), 60-68.
- [4] Gou, X., & Xiao, X. (2020). Short-term electricity price forecasting model based on empirical mode decomposition and LSTM neural network. *Journal of Xi'an University of Technology*, 36(01), 129-134. <https://doi.org/10.19322/j.cnki.issn.1006-4710.2020.01>
- [5] Yan, L., Yan, Z., Li, Z., Ma, N., Li, R., & Qin, J. (2023). Electricity Market Price Prediction Based on Quadratic Hybrid Decomposition and THPO Algorithm. *Energies*, 16(13), 5098. <https://doi.org/10.3390/en16135098>
- [6] Wu, Z., & Huang, N. E. (2009). Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method. *Advances in Adaptive Data Analysis*, 1(1), 1–41.
- [7] Nguyen, R. K., & Seong, J. H. (2014). Variational Mode Decomposition. *IEEE Transactions on Signal Processing*, 62(12), 3294–3302.
- [8] Wu, Z., Huang, H., & Huang, N. E. (2010). Complete Ensemble Empirical Mode Decomposition with Adaptive Noise. *Proceedings of the Royal Society A*, 466(2117), 243–281.
- [9] Wang, C., Liao, Z., & Chen, P. (2017). Improved CEEMDAN for Noise-Assisted Data Analysis and Decomposition. *Journal of Sound and Vibration*, 388, 127–136.
- [10] Dragomiretskiy, K., & Zosso, D. (2014). Variational Mode Decomposition. *IEEE Transactions on Signal Processing*,

- 62(3), 531-544. <https://doi.org/10.1109/TSP.2013.2288675>
- [11] Guo, J. (2024). Research and application of short-term power data forecasting based on deep learning. Linyi University. <https://doi.org/10.44252/d.cnki.glydx.2024.000030>
- [12] Guo, X., Hua, D., Bao, P., et al. (2024). A short-term electricity price forecasting method based on improved VMD-PSO-CNN-LSTM. *Journal of Electric Power Science and Technology*, 39(02), 35-43. <https://doi.org/10.19781/J.ISSN.16781>
- [13] Lai, Y., et al. (2025). Short-Term Power Load Prediction Method Based on VMD and EDE-BiLSTM. *IEEE Access*, 13, 10481-10488. <https://doi.org/10.1109/ACCESS.2024.3522656>
- [14] Zhang, J. (2011). Study on short-term electricity price hybrid forecasting model in electricity market environment. North China Electric Power University (Beijing).
- [15] Zhou, M., Yan, Z., Ni, Y., et al. (2004). A new ARIMA electricity price forecasting method with error prediction correction. *Journal of China Electrical Engineering*, (12), 67-72. <https://doi.org/10.13334/J.S>
- [16] Guo, Q., Lin, H., Li, Y., et al. (2024). Short-term electricity price forecast based on NGO-VMD-SSA-ESN. *Electrotechnical Technology*, (02), 130-136. <https://doi.org/10.19768/j.cnki.dgjs.2024.02.038>
- [17] Ji, X., Zeng, R., Zhang, Y., et al. (2022). CNN-LSTM Short-term Electricity Price Forecasting Based on Attention Mechanism. *Power System Protection and Control*, 50(17), 125-132. <https://doi.org/10.19783/j.cnki.pspc.211472>
- [18] Wang, P., et al. (2022). An Online Electricity Market Price Forecasting Method Via Random Forest. *IEEE Transactions on Industry Applications*, 58(6), 7013-7021. <https://doi.org/10.1109/TIA.2022.3198393>
- [19] Srivastava, A. K., Singh, D., Pandey, A. S., & Maini, T. (2019). A Novel Feature Selection and Short-Term Price Forecasting Based on a Decision Tree (J48) Model. *Energies*, 12(19), 3665. <https://doi.org/10.3390/en12193665>
- [20] Keles, D., Scelle, J., Paraschiv, F., & Fichtner, W. (2016). Extended forecast methods for day-ahead electricity spot prices applying artificial neural networks. *Applied Energy*, 162, 218-230. <https://doi.org/10.1016/j.apenergy.2015.09.087>
- [21] Bakir, A., & Rami, A. (2025). Enhanced electricity price forecasting in smart grids using an optimized hybrid convolutional Multi-Layer Perceptron deep network with Marine Predators Algorithm for feature selection. *Energy Sources, Part B: Economics, Planning, and Policy*, 20(1).
- [22] Al-Musaylh, M. S., Deo, R. C., Adarnowski, J. F., & Li, Y. (2018). Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia. *Advances in Engineering Informatics*, 35, 1-16. <https://doi.org/10.1016/j.aei.2017.11.002>
- [23] Lucas, A., Pegios, K., Kotsakis, E., & Clarke, D. (2020). Price Forecasting for the Balancing Energy Market Using Machine-Learning Regression. *Energies*, 13(20), 5420. <https://doi.org/10.3390/en13205420>
- [24] Loizidis, S., Kyprianou, A., & Georghiou, G. E. (2024). Electricity market price forecasting using ELM and Bootstrap analysis: A case study of the German and Finnish Day-Ahead markets. *Applied Energy*, 363, 123058. <https://doi.org/10.1016/j.apenergy.2024.123058>
- [25] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541-551.
- [26] Yang, Y., Ran, Q., Luo, D., et al. (2024). Short-term Electricity Price Forecasting Based on CNN-BIGRU with Attention Mechanism. *Journal of Power Systems and Automation*, 36(3), 22-29. <https://doi.org/10.19635/j.cnki.csu-epsa.001257>
- [27] Zhang, Y., Sun, M., Ji, X., et al. (2024). Short-term electricity price prediction method based on the quadratic decomposition and ATT-CNN-LSTM-MLR combined model. *Smart Electric Power*, 52(12), 65-72. <https://doi.org/10.20204/j.sp.2024.12009>
- [28] Kumar, K., Prabhakar, P., & Verma, A. (2024). Forecasting wind power using Optimized Recurrent Neural Network strategy with time-series data. *Optimization and Control Applications and Methods*, 45(4), 1798-1814. <https://doi.org/10.1002/oca.3122>
- [29] Ji, X. (2019). Research on Short-term Electricity Price Forecasting in Power Market Based on Deep Learning. North China Electric Power University. (In Chinese)
- [30] Zhou, Y., Su, Z., Gao, K., Wang, Z., Ye, W., & Zeng, J. (2024). A short-term electricity load forecasting method

- integrating empirical modal decomposition with SAM-LSTM. *Frontiers in Energy Research*, 12. <https://doi.org/10.3389/fenrg.2024.1423692>
- [31] Guohua, Q. (2023). Research on short-term electricity price forecasting model based on feature engineering and deep learning. Guangxi University. (In Chinese)
- [32] Shengke, H., Feihu, H., Zhiteng, C., et al. (2022). Day-ahead market marginal electricity price forecasting based on GCN-LSTM. *Proceedings of the CSEE*, 42(9), 3276-3286. <https://doi.org/10.13334/j.0258-8013.pcsee.202548>
- [33] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 5998-6008.
- [34] Xin, S., Simin, W., Jingdong, X., et al. (2024). Improved Transformer-PSO Short-term Electricity Price Forecasting Method Considering Multi-dimensional Factors. *Journal of Shanghai Jiaotong University*, 58(09), 1420-1431. <https://doi.org/10.16183/j.cnki.jsjtu.2023.065>
- [35] Cantillo-Luna, S., Moreno-Chuquen, R., Lopez-Sotelo, J., & Celeita, D. (2023). An Intra-Day Electricity Price Forecasting Based on a Probabilistic Transformer Neural Network Architecture. *Energies*, 16(19), 6767. <https://doi.org/10.3390/en16196767>
- [36] Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., & Long, M. (2024). iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. *arXiv Preprint arXiv:2310.06625*.
- [37] Holland, J. H. (1992). Genetic algorithms. *Scientific American*, 267(1), 66-73.
- [38] Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*. Perth, WA, Australia: IEEE, 1942-1948.
- [39] Dorigo, M., Caro, G. D., & Gambardella, L. M. (1999). Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2), 137-172.
- [40] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, 179, 2232-2248.
- [41] Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303-315.
- [42] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67-82.
- [43] Faramarzi, A., Heidarinejad, M., Mirjalili, S., et al. (2020). Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Systems with Applications*, 152, 113377.
- [44] Braik, M. S. (2021). Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems. *Expert Systems with Applications*, 174, 114685.
- [45] Hashim, F. A., Hussain, K., Houssein, E. H., et al. (2021). Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Applied Intelligence*, 51, 1531-1551.
- [46] Mohammadi-Balani, A., Nayeri, M. D., Azar, A., et al. (2021). Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Computers & Industrial Engineering*, 152, 107050.
- [47] Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 80, 20-34.
- [48] Khishe, M., & Mosavi, M. R. (2020). Chimp Optimization Algorithm. *Expert Systems with Applications*, 149, 113338.
- [49] Li, S., Chen, H., Wang, M., et al. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer System*, 111, 300-323.
- [50] Zhao, S. J., Zhang, T. R., Ma, S. L., Chen, M. (2022). Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. *Engineering Applications of Artificial Intelligence*, 114, 105075.
- [51] Abdollahzadeh, B., Gharehchopogh, F. S., & Mirjalili, S. (2021). African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Computers & Industrial Engineering*, 158, 107408.

- [52] Eslami, N., Yazdani, S., Mirzaei, M., et al. (2022). Aphid-Ant Mutualism: A novel nature-inspired metaheuristic algorithm for solving optimization problems. *Mathematics and Computers in Simulation*, 201, 362-395.
- [53] Zhong, C. T., Li, G., & Meng, Z. (2022). Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowledge-Based Systems*, 251, 109215.
- [54] Jafari, M., Salajegheh, E., & Salajegheh, J. (2021). Elephant clan optimization: A nature-inspired metaheuristic algorithm for the optimal design of structures. *Applied Soft Computing*, 113, 107892. <https://doi.org/10.1016/j.asoc.2021.107892>
- [55] Kazemi, M. V., & Veysari, E. F. (2022). A new optimization algorithm inspired by the quest for the evolution of human society: Human felicity algorithm. *Expert Systems with Applications*, 193, 116468. <https://doi.org/10.1016/j.eswa.2022.116468>
- [56] Pan, J. S., Zhang, L. G., Wang, R. B., et al. (2022). Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems. *Mathematics and Computers in Simulation*, 202, 343–373. <https://doi.org/10.1016/j.matcom.2022.01.024>
- [57] Yan, L., Yan, Z., Li, Z., Ma, N., Li, R., & Qin, J. (2023). Electricity market price prediction based on quadratic hybrid decomposition and THPO algorithm. *Energies*, 16(13), 5098. <https://doi.org/10.3390/en16135098>
- [58] Xu, Y., Huang, X., Zheng, X., Zeng, Z., & Jin, T. (2024). VMD-ATT-LSTM electricity price prediction based on grey wolf optimization algorithm in electricity markets considering renewable energy. *Renewable Energy*, 236, 121408. <https://doi.org/10.1016/j.renene.2024.121408>
- [59] Ahmadianfar, A., Heidari, A. A., Noshadian, S., Chen, H., & Gandomi, A. H. (2022). INFO: An efficient optimization algorithm based on weighted mean of vectors. *Expert Systems with Applications*, 195, 116516. <https://doi.org/10.1016/j.eswa.2022.116516>