

# Power load forecasting model based on CEEMDAN-VMD multiple noise reduction decomposition optimization DBO-LTSM<sup>1</sup>

Mingshen Xu\*, Non-Member

Runji Jiang\*, Non-Member

Jingjia Tian\*, Non-Member

Jianghai Geng<sup>\*a</sup>, Non-Member

In this paper, a hybrid forecasting model is constructed to deal with the time and space fluctuation and randomness of power load data. The model is based on multilevel noise reduction, variational modal decomposition and long-term and short-term memory network (LSTM) combined with Dung beetle optimizer (DBO). Firstly, an adaptive filter is used to denoise the original data. Then, the data is first decomposed by the method of Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN), and the decomposed intrinsic mode functions (IMFs) are then integrated by the method of sample entropy and K-Means clustering to form common IMFs (Co-IMFs). The high-frequency Co-IMF0 is secondarily processed by variational mode decomposition (VMD). Finally, the dung beetle optimization algorithm is used to optimize the parameters of LSTM, and the decomposed data are predicted and superimposed. Through this composite model, multi-level noise reduction of data and optimization of the LSTM algorithm are realized, and the prediction accuracy is further improved. The experimental data shows that the RMSE (root mean square error), MAPE (mean absolute percentage error), MAE (mean absolute error) and R-Square of the model are 1.90, 3.82, 0.90 and 0.99 respectively, which verifies the effectiveness of the proposed composite model in forecasting short-term and medium-term power load.

**Keywords:** clustering; ensemble empirical mode decomposition; variational mode decomposition; dung beetle optimization algorithm; long-term and short-term memory network

## 1. Introduction

In modern electrical power technology, accurate short-term power load forecasting often provides crucial guidance for future power generation plans of electric companies. However, the prediction is highly complex due to the influence of various factors on power demand, such as weather conditions, holidays, and socio-economic situations. Traditional load forecasting models primarily rely on the time series nature of historical load data, using relatively straightforward processing methods. Common techniques are based on gradient rules to infer future trends and states[1]. In regression analysis methods, the load and its influencing factors are considered as the independent and dependent variables, respectively. The relationship between the two is analyzed to establish a regression formula for predicting future load values[2]. Among them, time series methods are the most commonly used forecasting methods, such as the Autoregressive Integrated Moving Average (ARIMA) and Seasonal Autoregressive Integrated Moving Average (SARIMA) models (Yin, C., Liu, K., 2023)[3]-[4]. Additionally, gray models and neural network models are also partially employed[5]. However, these traditional methods often view historical and forecast data as simple mathematical associations, with limited capabilities in handling non-stationary and non-linear data.

In recent years, with the advancement of data analysis techniques and deep learning algorithms, machine learning technologies have been increasingly applied to load forecasting. Notably, Support Vector Machine (SVM)[6], Back Propagation (BP) neural network[7]-[8], K-Nearest Neighbor (KNN) neural network[9], and Recurrent Neural Network (Gated Recurrent Unit)[10]-[11] are classic algorithmic models in this field and are incorporated into

most existing load forecasting methods. However, these traditional models may encounter issues like local minima and overfitting during training if hyperparameters are not properly adjusted (Feng J., Yang J., 2021)[12]. To address this, recent studies have proposed a series of combined forecasting models using intelligent optimization algorithms. Examples include the Least Squares Support Vector Machine (LSSVM) model optimized by Sparrow Search Algorithm (SSA)[13]-[14], the SVM model optimized by Improved Particle Swarm Optimization (IPSO)[15], the Kernel Extreme Learning Machine (KELM) model optimized by the Improved Hunter-Prey Algorithm (LHPO)[16], the new algorithm based on Spatial Autocorrelation and Convolutional Long Short-Term Memory (SAC-ConvLSTM)[17]-[18], the SVM algorithm optimized by Cuckoo Search Algorithm based on Singular Spectrum Analysis[19], and the Asynchronous Adaptive Federated Learning for distributed load forecasting based on smart meter data[20]. These intelligent optimization algorithms significantly enhance the processing speed and prediction accuracy of the models by automatically optimizing parameters. However, the temporal fluctuations in short-term power load data, noise and outlier data might interfere with the forecasting process, making it difficult for the model to achieve the desired accuracy. Hence, several data processing methods have been applied to load forecasting models, such as Empirical Mode Decomposition (EMD)[21]-[22], Ensemble Empirical Mode Decomposition (EEMD)[23], Complementary EEMD (CEEMD)[24]-[25], Adaptive Noise Ensemble Empirical Mode Decomposition[26], and the improved CNN model based on Encoder-Decoder[27].

Based on the above analysis, to enhance the accuracy of the short-term forecasting model, this study adopts an improved version of the

<sup>1</sup> a) Correspondence to: Jianghai Geng. E-mail: gengjianghai@163.com  
North China Electric Power University,  
Lianchi District, Baoding City, Hebei Province, 071000, China

fully adaptive noise ensemble empirical mode decomposition method for data preprocessing. Subsequently, a composite forecasting model optimized by DBO for LSTM neural network hyperparameters is constructed. Initially, the CEEMDAN-VMD method is used to denoise the raw data. The DBO intelligent optimization strategy is then utilized to automatically optimize the hyperparameters of the LSTM. The resulting model is used to predict each element individually. Finally, the individual predictive results are aggregated and combined to obtain the final forecasted data.

## 2. Data Processing

There is more noise in the wind power data selected for this paper, so in order to improve the accuracy of the prediction, this paper uses the CEEMDAN-VMD method is employed to process the raw data. The principle of the method is described as follows:

### 2.1 CEEMDAN

CEEMDAN is an advanced signal decomposition method that offers significant advantages when dealing with complex signals. Its theoretical foundation primarily builds upon empirical mode decomposition (EMD) and further evolves into complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) by introducing Gaussian white noise. This development effectively addresses the mode mixing in EMD effectively. Moreover, CEEMDAN enhances the EEMD approach by incorporating specific adaptive noise at each decomposition stage. This not only improves computational efficiency but also produces superior modal decomposition results.

The decomposition procedure of the CEEMDAN algorithm is outlined as follows:

First, the description of variables is as follows:

Let  $E_k(\cdot)$  represent the k-th intrinsic mode component obtained after EMD decomposition, while the k-th intrinsic mode component obtained through CEEMDAN decomposition is denoted as  $IMF_k$ . Let  $v_j(t)$  be a Gaussian white noise signal satisfying a standard normal distribution, where  $j = 1, 2, \dots, N$  represents the number of times white noise is added.  $x(t)$  represents the signal to be decomposed.

The specific steps are as follows:

1. First, Gaussian white noise is added to the signal to be decomposed,  $x(t)$ , creating a new signal. Subsequently, this new signal undergoes EMD decomposition to obtain the first-order intrinsic mode component. Afterward, the average of all generated intrinsic mode components is computed, resulting in the first mode component, denoted as follows:

$$\overline{IMF}_1(t) = \frac{1}{N} \sum_{n=1}^N IMF_1^n(t) \quad (n = 1, 2, \dots, N) \quad (1)$$

2. Next, compute the residual sequence  $q_1(t)$  after removing the first mode component:

$$q_1(t) = x(t) - \overline{IMF}_1(t) \quad (2)$$

3. Adding the noise component  $\mu_1 M_1[v_m(t)]$  to the second residual sequence yields the second mode component  $IMF_2(t)$ .

$$IMF_2(t) = \frac{1}{N} \sum_{n=1}^N M_1[q_1(t)] + \mu_1 M_1[v_n(t)] \quad (3)$$

4. We repeat steps 2 and 3 iteratively until the resulting residual signal becomes a monotonic function and cannot be further decomposed. At this point, the iteration is terminated. The number

of obtained intrinsic mode components is denoted as  $k$ , and the relevant calculations are as follows:

$$q_i(t) = q_{i-1}(t) - IMF_i(t) \quad (4)$$

$$IMF_{i+1}(t) = \frac{1}{N} \sum_{n=1}^N M_1(\mu_i(t) + \mu_i M_i[v_m(t)]) \quad (5)$$

$$R(t) = x(t) - \sum_{i=1}^k IMF_i(t) \quad (6)$$

## 2.2 Sample Entropy, SE

Sample Entropy (SampEn) is a statistical method used to measure the complexity of time series data. Proposed by Richman and colleagues, its purpose is to quantitatively describe the regularity and complexity of a system. SampEn's concept is built upon the self-similarity and complexity of data, enabling the quantification of irregularities in time series data.

A notable advantage of SampEn is its weak dependency on data length, which leads to higher statistical stability and adaptability when quantifying features. It is based on the probability of new patterns emerging in a time series, thus measuring its complexity. In this context, a lower SampEn value indicates a higher level of self-similarity in the time series, with a lower probability of new pattern generation, implying simplicity. Conversely, a higher SampEn value signifies lower self-similarity in the time series, indicating a higher probability of generating new pattern generation, suggesting increased complexity.

The calculation of sample entropy involves several key steps:

First, the description of variables is as follows:

The original time series is denoted as  $T$ , the embedding dimension is represented by  $e$ , the similarity tolerance parameter is denoted as  $r$ , the distance between state vectors is indicated by  $d(s_i, s_j)$ , and the quantities along with their averages are represented by  $A_i$  and  $A^q(e)$ . These steps collectively contribute to the calculation of sample entropy, which serves to quantify the complexity of the provided time series data.

Then the specific steps are as follows:

1. Construction of State Vectors:

For a given original time series  $T = \{t_1, t_2, \dots, t_N\}$ , select an appropriate embedding dimension  $q$  to construct new state vectors, denoted as  $T = \{t_i, t_{i+1}, \dots, t_{i+q-1}\}$ , where  $i = 1, 2, \dots, N - q$ .

2. Distance Calculation:

The maximum value of the element-wise difference distance between two state vectors is

$$d(s_i, s_j) = \max(|s_{i+k} - s_{j+k}|) \quad (7)$$

3. Calculation of Similarity Count:

With a given similarity tolerance parameter  $p$ , count the number of distances between state vectors  $s_i$  and  $s_j$  that are less than or equal to  $e$ , resulting in quantity  $D_i$ . Define

$$D_i^q(p) = \frac{D_i}{N-q-1} \quad (8)$$

, where  $1 \leq i \leq N - q$ .

4. Calculation of Averages:

The average value of  $D_i^q(p)$  is

$$D^q(e) = \frac{\sum_{i=1}^{N-q} D_i^q(p)}{N-q} \quad (9)$$

5. Increasing Embedding Dimension and Calculating Sample Entropy:

We increase the embedding dimension to  $q + 1$  and repeat the above steps to calculate  $A^{q+1}(e)$ . When the time series length is limited, sample entropy can be calculated using the following

formula:

$$\text{SampEn}(q, e) = -\ln \left( \frac{D^{q+1}(e)}{D^q(e)} \right) \quad (10)$$

### 2.3 Variational Mode Decomposition (VMD)

Variational Mode Decomposition (VMD) is a time-frequency analysis tool innovatively designed by Dragomiretskiy et al. in 2014. The key advantage of VMD lies in its ability to decompose multi-component signals into several single-component amplitude and frequency modulated signals simultaneously, while addressing the issues of endpoint effects and misleading components that may arise during the iterative process. As an analytical method, VMD is capable of handling nonlinear and non-stationary signals, decomposing complex multi-frequency signals into N intrinsic mode functions (IMFs). Additionally, in the implementation of the VMD algorithm, Wiener filtering is introduced to achieve effective noise reduction. By utilizing the Alternating Direction Method of Multipliers (ADMM) approach to update each mode function and its center frequency, each sub-signal acquires an independent center frequency, while estimating the bandwidth and minimum of sub-signal are minimized. This ensures that the reconstruction result of each sub-signal closely approximates the original signal.

The fundamental steps of VMD can be divided into the establishment and resolution of the variational problem. The variational constraint problem can be formulated using the following equation:

$$\begin{cases} \min_{\{\xi_2\}\{\xi_0\}} \left\{ \sum_k \left\| \partial_t \left\{ \left[ \delta(t) + \frac{j}{\pi t} \right] \cdot \rho_k(t) \right\} \cdot e^{-j\xi_d(t)} \right\|_2^2 \right\} \\ \text{s.t. } \sum_k \xi_k = h \end{cases} \quad (11)$$

Where  $\mathbf{o}_n = o_1, o_2, \dots, o_n$  represents the mode components,  $\xi_n = \xi_1, \xi_2, \dots, \xi_n$  stands for frequency centers,  $\Delta(t)$  represents the pulse function,  $v$  is the number of modes, and  $h$  denotes the original signal. To solve this variational problem, the introduction of quadratic penalty factor  $\alpha$  and the Lagrange multiplier operator  $\lambda$  is required. This transforms the aforementioned constrained problem into an unconstrained problem, achieved as follows:

$$L(\{\rho_n\}, \{\xi_n\}, \lambda) = \alpha \sum_k \left\| \partial_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) \cdot \rho_k(t) \right] \cdot e^{-j\xi_d(t)} \right\|_2^2 + \|h(t) - \sum_k \rho_k(t)\| + [\lambda(t), h(t) - \sum_k \rho_k(t)] \dots \quad (12)$$

Subsequently, the ADMM approach is employed to alternately update  $\rho_n$ ,  $\xi_n$ , and  $\lambda_0$ . The iterative formula for  $\rho_n$  is given by:

$$\hat{\rho}_k^{n+1}(\xi) = \frac{\hat{h}(\xi) - \sum_{i \neq k} \hat{\rho}_i(\xi) + \hat{\lambda}(\xi)/2}{1 + 2\alpha(\xi - \xi_k)^2} \quad (13)$$

The iterative formula for  $\xi_n$  is as follows:

$$\xi_k^{n+1} = \frac{\int_0^\infty \xi |\hat{\rho}_k^{n+1}(\xi)|^2 d\xi}{\int_0^\infty |\hat{\rho}_k^{n+1}(\xi)|^2 d\xi} \quad (14)$$

The iterative formula for  $\lambda$  is as follows:

$$\hat{\lambda}^{n+1}(\xi) = \hat{\lambda}^n(\xi) + \tau \left[ \hat{h}(\xi) - \sum_k \hat{\rho}_k^{n+1}(\xi) \right] \quad (15)$$

$$\sum_{k=1}^K \frac{\|\hat{\rho}_k^{n+1} - \hat{\xi}_k^n\|_2^2}{\hat{\xi}_k^n} < e \quad (16)$$

### 3. DBO Optimization Algorithm

Dung Beetle Optimizer (DBO) is a unique population intelligence optimization algorithm proposed by XUE J et al. in 2023. The algorithm is borrowed from the social behaviors of dung beetle populations, including ball rolling, dancing, searching for food, breeding, and stealing. In practice, the DBO algorithm optimizes the balance between global search and local exploitation, featuring fast convergence and high solution accuracy. Each dung beetle swarm consists of four different types of agent dung beetles, i.e., ball-rolling dung beetles, breeding dung beetles (breeding balls), baby dung beetles, and stealing dung beetles.

#### 3.1 Ball-Rolling Dung Beetle

Ball-rolling dung beetles have the unique behavior of making balls of dung and rolling them to a desired location. During the rolling process, dung beetles rely on astronomical cues (e.g., the position of the sun or the direction of the wind) to maintain the straightness of their rolling path. To simulate this rolling behavior, we have dung beetle beetles move in a specified direction within the search space. As the rolling progresses, the position of the dung beetle is updated and its rolling behavior can be represented by the following mathematical model:

$$x_i(m+1) = x_i(m) + \alpha \times k \times x_i(m-1) + b \times \Delta x, \Delta x = |x_i(m) - X^w| \dots \dots \dots \quad (17)$$

Here,  $t$  represents the number of iterations now,  $x_i(m)$  represents the position information of the first dung beetle at the  $t$ th iteration.  $k$  is a constant of the deflection coefficient taking values in the range of  $(0, 0.2]$ ,  $p$  is a constant belonging to  $(0, 1)$ .  $\alpha$  is a natural coefficient taking the value of -1 or 1, and  $X^w$  stands for the global worst position, and  $\Delta x$  is used to simulate the variations of light intensity.

When a dung beetle is unable to move forward (e.g., when it encounters an obstacle), it changes its direction by dancing to find a new path. We use the tangent function to model the dancing behavior of the dung beetle to get a new rolling direction. Once the dung beetle succeeds in determining the new direction, it continues to roll. Therefore, the location of the dung beetle's dancing behavior can be defined as follows:

$$x_i(m+1) = x_i(m) + \alpha \times k \times x_i(m-1) + b \times \Delta x, \Delta x = |x_i(m) - X^w| \dots \dots \dots \quad (18)$$

In this equation,  $\theta$  ranges from  $[0, \pi]$ . If  $\theta$  is equal to  $0, \pi/2$  or  $\pi$ , then the position of the dung beetle will not be updated.

#### 3.2 Dung Beetle Breeding

In nature, dung beetles provide a safe environment for their offspring by rolling dung balls and hiding them in a safe place. For dung beetles, it is very important to choose a suitable spawning place. Based on this, we design a boundary selection strategy, which simulates the behavior of female dung beetles in choosing spawning areas. The expression of the strategy is as follows

$$Lb^* = \max(B^* \times (1 - R), Lb) \quad (19)$$

$$Ub^* = \min(B^* \times (1 - R), ub) \quad (20)$$

Among them,  $B^*$  is the current local optimal position, and  $Lb^*$  and  $Ub^*$  are the lower and upper bounds of the spawning area respectively. The formula of  $R$  is  $R = 1 - t/T_{\max}$ , where  $T_{\max}$  is the maximum number of iterations, and  $Lb$  and  $Ub$  represent the minimum and maximum values of the optimization problem

respectively.

Here,  $X_i(t)$  is the position information of the  $i$  breeding ball in the  $t$  iteration,  $p_1$  and  $p_2$  are two independent random vectors with the size of  $1 \times D$ , and  $D$  is the dimension of the optimization problem. In particular, the location of breeding balls is strictly limited to a specific range, that is, the spawning area.

### 3.3 Small Dung Beetles

Some mature dung beetles crawl out of the ground in search of food, and we refer to these dung beetles "little dung beetles". In order to simulate the process of these little dung beetles searching for food, we set up an optimal feeding area to guide their behavior. Specifically, the boundary of the optimal feeding area can be defined by the following formula: In these formulas,  $B^b$  is the global optimal position at present,  $Lb^b$  and  $Ub^b$  are the lowest and highest boundaries of the optimal feeding area respectively, and other parameters remain consistent with those defined in the aforementioned formula (3). Therefore, the position of the dung beetle will be updated as follows:

$$Lb^b = \max(B^b \times (1 - R), Lb) \quad (21)$$

$$U_b^b = \min(B^b \times (1 - R), ub) \quad (22)$$

In these formulas,  $B^b A$  is the global optimal position at present,  $Lb^b$  and  $Ub^b$  are the lowest and highest boundaries of the optimal feeding area respectively, and other parameters are consistent with those defined in the aforementioned formula (21)-(22). Thus, the position of the dung beetle will be updated as follows:

In the formula,  $x_i(t)$  is the position of the  $i$ -th dung beetle in the  $T$ -th iteration,  $C_1$  is a random number with a normal distribution, and  $C_2$  is a random vector with a value between  $(0, 1)$ .

Some dung beetles make a living by stealing dung balls from other dung beetles. We call them "thief" dung beetles. At the same time, we know from formula (5) that  $X^b$  is the most ideal food source, so we can infer that the surrounding area is the main area where thieves and dung beetles compete for food. During each iteration, the position of the thief dung beetle will be updated according to the following rules:

$$x_i(m+1) = B^b + S \times g \times (|x_i(m) - X^*| + |x_i(m) - X^b|) \quad (24)$$

In this rule,  $x_i(m)$  stands for the position of the  $i$ -th thief dung beetle in the  $t$  iteration,  $G$  is a random vector with a size of  $1 \times D$  that follows a normal distribution, and  $S$  is a constant.

The above is the basic flow of the DBO algorithm.

### **3.4 Long Short-Term Memory Model (LSTM)**

Long Short-Term Memory (LSTM) is a modified Recurrent Neural Network (RNN) whose goal is to solve the problem of gradient vanishing and gradient explosion that RNNs are prone to when dealing with long sequence training. LSTM has better performance in longer time series.

The uniqueness of LSTM is that it introduces specially designed internal memory units  $r$ . These units update  $r_{\{i-1\}}$  to the current state by fusing the information of the neuron and the state  $r_t$  of the previous time step, which is then deflated by a tanh activation function. Define  $r'_t$  as the temporary state  $w_r$  and  $U_r$  of the current unit as the weight matrix, then the corresponding formula is expressed as:

$$r_t' = \tanh(w_r x_t + U_r h_{t-1}) \quad (25)$$

$$r_t = f_t r_{t-1} + i_t r'_t \quad (26)$$

In the LSTM model, the information of a neuron consists of three key components:

The input gate  $p_t$  controls the number of inputs  $x_t$  and neuron states  $h_{t-1}$  updated to the neuron information from the previous step. Let  $w_r$  and  $U_r$  be the weight matrices,  $b_t$  be the bias, and  $\sigma$  be the activation function (usually Sigmoid or ReLU), then Eq. is expressed as:

$$p_t = \sigma(w_r x_t + U_r h_{t-1} + b_t) \quad (27)$$

The Forget gate  $f_t$  controls the input  $x_t$  from the previous step and the amount of information to be forgotten in the neuron state  $h_{t-1}$ . Let  $w_f$  and  $U_f$  be the weight matrices,  $b_f$  is the bias, and the corresponding formula is expressed as:

$$f_t = \sigma(w_f x_t + U_f h_{t-1} + b_f) \quad (29)$$

$$g_t = \sigma(w_f x_t + U_f h_{t-1} + b_f) \quad (28)$$

The outgoing  $z_t$  controls the amount of information about the current state  $q_t$  to be output into the current neuron state  $h_t$ . Let  $w_z$  and  $U_z$  be the weighting matrix and  $b_z$  be the bias, the corresponding formula is expressed as:

By introducing Forget gate, we can make the model "forget" unimportant information. At the same time, compared to ordinary RNNs that only have a single memory overlay, LSTMs control the state transfer by gating the state, and show better performance in tasks that require long-term memory. However, this also brings more parameters, with a consequent increase in the difficulty of global optimization, and each gate unit relies on the memory unit weights of the previous neuron, thus typically requiring more operational resources to support training.

### 3.5 Establishment of the DBO-LSTM Model

DBO-LSTM is a short-term power load forecasting model that incorporates Dung Beetle Optimizer (DBO) and Long Short-Term Memory Network (LSTM). In the model, we take the number of iterations, the learning rate, and the number of neurons in the hidden layer of LSTM as the optimization objectives of the DBO algorithm, and the optimal solution is obtained through global search and given to the LSTM network, so as to improve the prediction accuracy. The key steps to construct the DBO-LSTM short-term electricity load forecasting model are as follows:

(1) Initialize the main parameters of the DBO. This includes determining the population size, the maximum number of iterations, the proportions of rolling dung beetles, breeding dung beetles, small dung beetles and stealing dung beetles, and the search latitude. In our model, there are three parameters to be optimized, so the search dimension is three-dimensional.

(2) Initialize the location of the population. We randomly generate a dung beetle individual, denoted as  $X_t = (\alpha, \beta, \gamma)$ , where  $\alpha$  represents the number of iterations of the LSTM network,  $\beta$  represents the learning rate, and  $\gamma$  represents the number of neurons in the hidden layer. Meanwhile, other parameters of the LSTM are determined and the prediction scheme is set as a single-step prediction.

(3) Determine the fitness function of the population. Use the initialized population parameters to construct the LSTM model and calculate the RMSE between the prediction results of the training set and the actual values as the fitness function of the population.

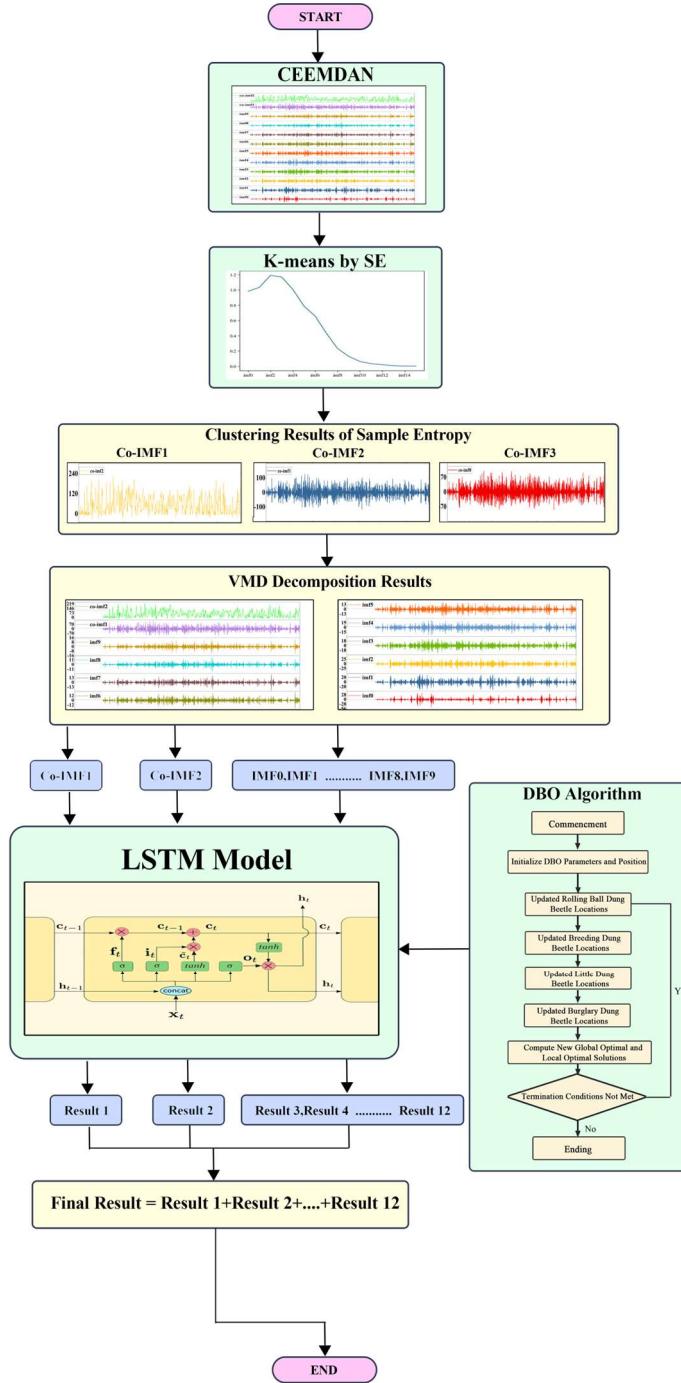


Fig. 1. The steps of this model.

#### 4. Example Analysis and Simulation:

In this paper, a total of 35,040 load data points from January 1, 2019, to December 30, 2019, along with related meteorological and temporal data of Xinjiang wind power are used for model construction and validation of power load forecasting. Among them, the power load data is sampled at a sampling interval of 15 minutes, i.e., there are 96 data sampling points per day. Meanwhile, we select the first 34960 data as the training dataset and the last 100 data as the validation dataset for short-term forecasting, while we select the

first 28040 data as the training dataset and the last 7000 data as the validation dataset for long-term forecasting.

##### 4.1 Selection of Input Variables

After analyzing the correlation between different wind speeds and directions and the load data, this paper selects the nine variables with the highest correlation as input variables to train the model and predict the future load data, and the specific data and results are as Table 1:

Table 1. Selected variables and their correlation with loads

input variable	Correlation coefficient
Anemometer tower 10m wind speed(m/s)	0.76
Anemometer tower 30m wind speed(m/s)	0.47
Anemometer tower 50m wind speed(m/s)	0.78
Anemometer tower 10m wind direction(°)	-0.34
Anemometer tower 30m wind direction(°)	-0.36
Anemometer tower 50m wind direction(°)	0.190069
Temperature(°)	0.132965
Pneumatic(hPa)	-0.127836
Humidity(%)	-0.136031

#### 4.2 Data Pre-Processing

Subsequently, we began to carry out data processing.

First of all, it is necessary to perform data noise reduction on the original utilizes therefore, this paper adopts the CEEMDAN method to decompose the original electric load data.

To reduce the computational complexity, improve the speed of model training and avoid the problem of overfitting. We find the sample entropy for each IMF and use the k-means method for clustering . Subsequently, we derived three new integrated sequences (Co-IMFs): the high-frequency sequence Co-IMF0 (containing imf1-imf4), the mid-frequency sequence Co-IMF1 (containing imf5-imf7), and the low-frequency sequence Co-IMF2 (containing imf8 -imf15). The variation patterns of all three Co-IMs are relatively stable, which facilitates us to further extract the fluctuation characteristics of each IMF and train the prediction model.

Meanwhile, in order to accurately predict high-frequency sequences, this paper implements the Variable Modal Decomposition (VMD) method for their secondary decomposition. In order to avoid the problem of under-decomposition or over-decomposition during the decomposition process, we tried several different k values and calculated and compared different center frequencies. In the end, we chose k = 10 as the parameter for the VMD model decomposition, while the default values were used for the other parameters.

In order to improve the prediction accuracy and speed, this paper normalizes the load data as follows :

$$Load_n = \frac{Load_n - Load_{nmax}}{Load_{nmax} - Load_{nmin}} \quad (30)$$

In this formula,  $Load_n$  is the initial load value of the nth sequence decomposition load;  $Load_{nmax}$  is the maximum value of the nth decomposition sequence load;  $Load_{nmin}$  is the minimum value of the nth decomposition sequence load. At this point, this paper has completed all the pre-processing of electric load data.

#### 4.3 Selection of Evaluation Indicators

In order to quantitatively assess the accuracy of the prediction model, we use three main evaluation metrics in this paper. These include the mean absolute error, which measures the average

absolute difference between the predicted value and the actual value; the mean absolute percentage error, which provides the percentage of the prediction error relative to the actual value, enabling us to understand the effect of the error in the aggregate; and the root mean square error, which is the square root of the average of the squares of the prediction errors, and is able to give the magnitude of the prediction error. The exact formula is given below:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_{estimate}(i) - p_{actual}(i))^2} \quad (31)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^n \left| \frac{p_{estimate}(i) - p_{actual}(i)}{p_{estimate}(i)} \right| \times 100 \quad (32)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_{estimate}(i) - p_{actual}(i)| \quad (33)$$

$$R = \frac{\sum_{i=1}^n (p_{estimate}(i) - \bar{p}_{estimate})(p_{actual}(i) - \bar{p}_{actual})}{\sqrt{\sum_{i=1}^n (p_{estimate}(i) - \bar{p}_{estimate})^2 \sum_{i=1}^n (p_{actual}(i) - \bar{p}_{actual})^2}} \quad (34)$$

In this formula,  $p_{estimate}(i)$  and  $p_{actual}(i)$  are the predicted and measured values, respectively;  $\bar{p}_{estimate}$  and  $\bar{p}_{actual}$  are the average of the predicted and measured values, respectively.

#### 4.4 Model Parameter Setting

The DBO optimization parameters and the original parameters for the DBO-LSTM network are shown in Table 2.

Table 2. Original network parameters

Parameter	Value
Population Size	64
Maximum Iteration	30
Number	
Rolling Cockroach Ratio	0.2
Ovipositing Cockroach Ratio	0.4
Small Cockroach Ratio	0.2
Stealing Cockroach Ratio	0.2
Learning Rate Range	[0.001, 0.01]
Neuron Number Range	[10, 100]
Fitness Function	RMSE = 1
Activation Function	selu

After using BWO to perform hyperparameter optimization for the LSTM network, the optimal hyperparameter values are shown in Table 3.

Table 3. Results after hyperparameter optimization

Parameter	Optimal Value
Fitness Value	0.001
Learning Rate	0.003
Number of Neurons in LSTM Layer 1	55
Number of Neurons in LSTM Layer 2	213

#### 4.5 Prediction Results and Comparative Analysis

##### 4.5.1 Comparison of Model Prediction Metrics Before and After Optimization

To validate the efficacy of the proposed model, this paper initially employed the same experimental data to compare the original LSTM model with the denoised DBO-LSTM model. After the experimentation, the comparative results are shown in Table 4 and Fig. 2.

Table 4. Comparison of model prediction metrics before and after optimization

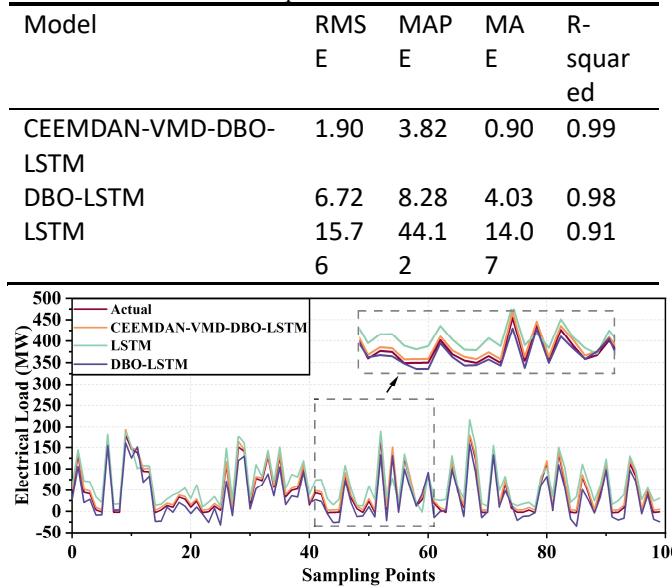


Fig. 2. Comparative analysis of model prediction results before and after optimization

##### 4.5.2 Comparative Evaluation of Predictive Metrics for Classical Models

Subsequently, this paper selects four other typical prediction models for comparison: BP Neural Network, K-Nearest Neighbors (K-NN), XGBoost, and Support Vector Machine (SVM). These are compared with the proposed CEEMDAN-VMD-DBO-LSTM model. The experimental results are in Table 5 and Fig. 3:

Table 5. Comparative Analysis of Predictive Performance Metrics for Classical Models

Model	RMS E	MAP E	MA E	R-squared
CEEMDAN-VMD-DBO-LSTM	1.90	3.82	0.90	0.99
SVM	28.99	459.51	21.92	0.71
BP	23.36	286.05	17.28	0.81
KNN	21.35	249.18	15.08	0.84
XGBoost	47.81	179.82	32.33	0.83

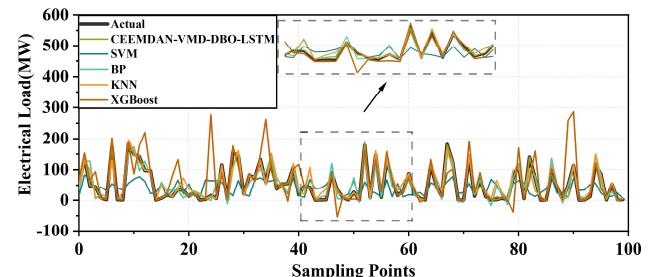


Fig.3. Comparative visualization of predictive results from classical models

##### 4.5.3 Comparative Forecasting Analysis of Optimization Algorithms

Furthermore, this paper selects three other optimization algorithms: SSA, MVO, and PSO, for comparison with the proposed CEEMDAN-VMD-DBO optimization method for LSTM. The experimental results are as Table 6 and Fig. 4:

Table 6. Comparative analysis of forecasting metrics for different optimization algorithms

MODEL	RMS E	MAP E	MA E	R-squared
CEEMDAN-VMD-DBO-LSTM	1.90	3.82	0.9	0.99
SSA—LSTM	11.50	11.97	5.60	0.95
MVO—LSTM	12.30	12.89	6.44	0.94
PSO—LSTM	9.046	10.79	5.49	0.97

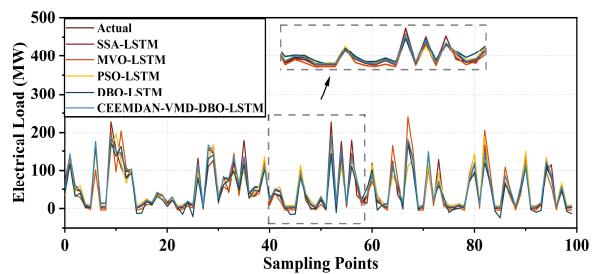


Fig.4. Comparative visualization of forecasting results for different optimization algorithm models

After consolidating the experimental results, this paper computed the data improvement of the proposed model in comparison to other benchmark models in terms of RMSE, MAPE, and MAE. The visualization of these improvements is presented as Fig.5:

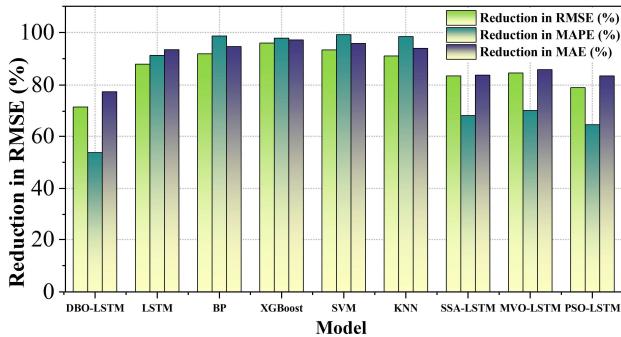


Fig.5. Data visualization of predictive accuracy Enhancement for CEEMDAN-VMD-DBO-LSTM

It can be observed that the prediction model proposed in this study achieves a reduction of over 40% in terms of RMSE, MAPE, and MAE. This further attests to the efficacy of the model.

#### 4.5.4 Validation of Model Performance Over Extended Time Durations

To validate the generality of the proposed model, specifically its accuracy over extended periods, this paper utilizes the subsequent 7000 data points as a validation set. The experimental results are as Table 7 and Fig.6:

Table 7. Long-Term forecasting metrics data for CEEMDAN-VMD-DBO-LSTM

RMSE	MAPE	MAE	R-squared
4.0962	11.3937	2.2349	0.9893

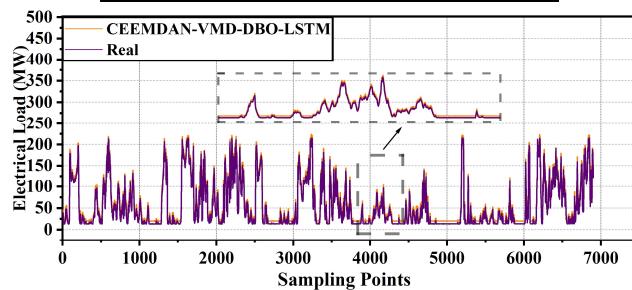


Fig.6. Long-Term Load Forecasting Curve for CEEMDAN-VMD-DBO-LSTM

The experimental results indicate that, in a long-term prediction scenario, the model continues to maintain high accuracy. This suggests that the model proposed in this paper is versatile and applicable to forecasting tasks across various time durations.

In summary, the experimental results clearly demonstrate the precision improvement of the CEEMDAN-VMD-DBO-LSTM model in various metrics compared to traditional prediction models and optimization algorithms. This directly reflects its superiority and generality in electricity forecasting tasks over different time spans. The model exhibits notable advantages in prediction accuracy and error reduction, providing a robust foundation for our future research and model enhancements.

## 5. Conclusion

Addressing the challenges of short-term electric power load forecasting in modern power systems, this study uses the annual load data of Xinjiang in 2019, along with related meteorological and temporal data as a benchmark. The research introduces a composite model based on CEEMDAN-VMD multi-level noise reduction and DBO-LSTM to forecast mid-short term electric power loads. This

model is further validated against multiple network models and optimization algorithm models. The main conclusions are as follows:

(1) Unlike classical forecasting models, this study employs the CEEMDAN-VMD method to repeatedly decompose the wind power time series of wind farms into noise-reduced components. This approach facilitates the integration of features from newly established wind farms with surrounding wind farms under the same modality. Subsequently, a forecasting model is developed for each component, significantly enhancing the prediction accuracy of the composite model.

(2) Based on data comparisons with other models, the superiority of the CEEMDAN-VMD-DBO-LSTM model in electricity forecasting tasks is evident. The model exhibits pronounced advantages in terms of prediction accuracy and error reduction, providing a robust foundation for our future research and model enhancements.

(3) This model is adaptable to various environments. In forecasting water resource management, it aids in the rational planning of watershed water demands. Simultaneously, in the power grid load forecasting, the model efficiently assists nations in addressing challenges posed by extreme weather conditions such as severe droughts, typhoons, and heavy rainfall. This allows different sectors to more swiftly forecast and regulate the power grid.

## References

- (1) Sousa, J.C., Jorge, H.M., and Neves, L.P.: "Short-term load forecasting based on support vector regression and load profiling", Energy Reports, Vol.7, No. Supplement 6, pp.487-492 (2013).
- (2) Hosking, J.R.M. and Natarajan, R.: "Short-term forecasting of the daily load curve for residential electricity usage in the Smart Grid", Applied Stochastic Models in Business and Industry, Vol.29, No.5, pp.429-438 (2013).
- (3) Zhou, Z., Jing, X., and Li, Y.: "Prediction of Electricity Consumption in Anhui Province Based on HP-ARIMA-BP", Proceedings of SPIE - The International Society for Optical Engineering, Vol.12594, No.1, pp.125940Y (2023).
- (4) Yin, C., Liu, K., and Zhang, Q.: "SARIMA-Based Medium- and Long-Term Load Forecasting", Strategic Planning for Energy and the Environment, Vol.42, No.2, pp.283-306 (2023).
- (5) Feng, J., Yang, J., and Li, Y.: "Load forecasting of electric vehicle charging station based on grey theory and neural network", Energy Reports, Vol.7, No. Supplement 6, pp.487-492 (2021).
- (6) Aasim, S.N. and Singh, A.: "Data driven day-ahead electrical load forecasting through repeated wavelet transform assisted SVM model", Applied Soft Computing, Vol.111, No.1, pp.107730 (2021).
- (7) Sun, H., Wang, S., and Liu, C.: "Load Forecasting of Electric Vehicle Charging Station Based on Power Big Data and Improved BP Neural Network", Lecture Notes on Data Engineering and Communications Technologies, Vol.153, No.1, pp.410-418 (2023).
- (8) Xiao, Y., He, X., and Huang, R.: "Prediction of Distributed Photovoltaic Users' Electric Energy Data Based on BP Neural Network Algorithm", 2022 12th International Conference on Power and Energy Systems, ICPES 2022, Vol.1, No.1, pp.816-820 (2022).
- (9) Hong, H., Wu, K., and Zhang, Y.: "Study on the BL Early Warning Model of Transmission Line Tower Based on KNN-SVM Algorithm", 2021 2nd International Conference on Artificial Intelligence, Computer Networks and Communications, AICNC 2021, Vol.2146, No.1, Article number: 012003 (2022).
- (10) Aseeri, A.O.: "Effective RNN-Based Forecasting Methodology Design for Improving Short-Term Power Load Forecasts: Application to Large-Scale Power-Grid Time Series", Journal of Computational Science, Vol.68, No.1, pp.101984 (2023).
- (11) Singh, S. and Tripathi, M.M.: "Short-Term Electricity Demand Forecast Using Deep RNN and Stacked LSTM", Advances in Manufacturing Technology and Management - Proceedings of 6th International Conference on Advanced Production and Industrial Engineering, ICAPIE 2021, Vol.1, No.1, pp.578-588 (2023).
- (12) Fan, G.-F., Li, Y., and Zhang, X.-Y.: "Short-term load forecasting based on a generalized regression neural network optimized by an improved sparrow search algorithm using the empirical wavelet decomposition method", Energy

- Science&Engineering, Vol.11, No.1, pp.e1465 (2023).
- (13) Feng, J., Yang, J., and Li, Y.: "Load forecasting of electric vehicle charging station based on grey theory and neural network", Energy Reports, Vol.7, No.Supplement 6, pp.487-492 (2021).
- (14) Yang, H., Tian, F., and Zhang, P.: "Short-term load forecasting based on CEEMD-FE-AOA-LSSVM", Dianli Xitong Baohu yu Kongzhi/Power System Protection and Control, Vol.50, No.13, pp.126-133 (2022).
- (15) Chen, Y., Yang, Y., and Liu, C.: "A hybrid application algorithm based on the support vector machine and artificial intelligence: An example of electric load forecasting", Applied Mathematical Modelling, Vol.39, No.9, pp.2617-2632 (2015).
- (16) Shan, J.-n., Wang, H.-z., and Pei, G.: "Research on short-term power prediction of wind power generation based on WT-CABC-KELM", Energy Reports, Vol.8, No.Supplement 8, pp.800-809 (2022).
- (17) Liu, J. and Yin, Y.: "Power Load Forecasting Considering Climate Factors Based on IPSO-Elman Method in China", Energies, Vol.15, No.3, Article 1236 (2022).
- (18) Jalalifar, R., Delavar, M.R., and Ghaderi, S.F.: "SAC-ConvLSTM: A Novel Spatio-Temporal Deep Learning-based Approach for a Short Term Power Load Forecasting", Expert Systems with Applications, Vol.1, No.1, pp.121487 (2023).
- (19) Zhang, X., Wang, J., and Zhang, K.: "Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by Cuckoo search algorithm", Electric Power Systems Research, Vol.146, No.1, pp.270-285 (2017).
- (20) Fekri, M.N., Grolinger, K., and Mir, S.: "Asynchronous adaptive federated learning for distributed load forecasting with smart meter data", International Journal of Electrical Power & Energy Systems, Vol.153, No.1, pp.109285 (2023).
- (21) Mounir, N., Ouadi, H., and Jrhilifa, I.: "Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system", Energy and Buildings, Vol.288, No.1, pp.113022 (2023).
- (22) Mounir, N., Ouadi, H., and Jrhilifa, I.: "Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system", Energy and Buildings, Vol.288, No.1, pp.113022 (2023).
- (23) Yuan, J., Wang, L., and Qiu, Y.: "Short-term electric load forecasting based on improved Extreme Learning Machine Model", Energy Reports, Vol.7, No.Supplement 7, pp.1563-1573 (2021).
- (24) Yang, W., Liu, Y., and Shu, Q.: "A Short-term Load Forecasting Model Based on CEEMD", Dianwang Jishu/Power System Technology, Vol.46, No.9, pp.3615-3622 (2022).
- (25) Yang, H., Tian, F., and Zhang, P.: "Short-term load forecasting based on CEEMD-FE-AOA-LSSVM", Dianli Xitong Baohu yu Kongzhi/Power System Protection and Control, Vol.50, No.13, pp.126-133 (2022).
- (26) Xiao, B., Gao, W., and Li, D.: "Spatial load forecasting method based on  $3\sigma$ -CEEMDAN-LSTM", Dianli Zidonghua Shebei/Electric Power Automation Equipment, Vol.43, No.3, pp.159-165 (2023).
- (27) Suxann Lim, Jun-Ho Huh, Seok-hoon Hong, Chul-Young Park, and Jong-Chan Kim: "Solar Power Forecasting Using CNN-LSTM Hybrid Model", Energy, Vol.15, No.21, pp.8233 (2022).



**Jingjia Tian** (Non-Member) was born in Guizhou, China, in 2003. Since 2021, he has been pursuing her B.S. degree in Automation in the Department of Automation at North China Electric Power University (Baoding). He current research interests include the energy internet, power generation system modeling, simulation, and optimization settings.



**Jianghai Geng** (Non-Member) born in May 1980, is of Han ethnicity and holds the title of Senior Engineer. From 2000 to 2004, he studied High Voltage and Insulation Technology at North China Electric Power University (Baoding), earning a Bachelor of Engineering degree. He continued his studies there from 2004 to 2007, obtaining a Master of Engineering degree. From 2007 to 2012, he worked at the Electric Power Research Institute of Hebei Province Electric Power Company. Since August 2012, he has been with the High Voltage Research Department at North China Electric Power University (Baoding). His current research focuses on ultra-high voltage power transmission technology, electrical equipment monitoring and fault diagnosis, and electromagnetic compatibility in power electronic systems.



**Mingshen Xu** (Non-Member) was born in 2004. Since 2022, he has been pursuing his B.S. degree in Information and Computational Science in the Department of Mathematics and Physics at North China Electric Power University (Baoding). His current research interests include computational mathematics, energy big data analysis, and load forecasting.



**Runji Jiang** (Non-Member) born in 2004. Since 2022, he has been pursuing his B.S. degree in Electrical Engineering and Automation at North China Electric Power University (Baoding). His current research interests include Extra-High Voltage, electric automation and power grid dispatch.