EAI Endorsed Transactions
on Energy Web
Research Article EAI.EU

Short-term Electricity Load Forecasting Based on Improved Seagull Algorithm Optimized Gated Recurrent Unit Neural Network

Mengfan Xu[1] and Junyang Pan[1,*]

[1]Anhui University of Finance and Economics, Institute of Statistics and Applied Mathematics, Bengbu233030, Anhui, China

Abstract

INTRODUCTION: The complexity of the power network, changes in weather conditions, diverse geographical locations, and holiday activities comprehensively affect the normal operation of power loads. Power load changes have characteristics such as non stationarity, randomness, seasonality, and high volatility. Therefore, how to construct accurate short-term power load forecasting models has become the key to the normal operation and maintenance of power.
OBJECTIVES: Accurate short-term power load forecasting helps to arrange power consumption planning, optimize power usage and largely reduce power system losses and operating costs.
METHODS: A hybrid decomposition-optimization-integration load forecasting method is proposed to address the problems of low accuracy of current short-term power load forecasting methods.
RESULTS: The original power load time series is decomposed using the complete ensemble empirical modal decomposition method, while the correlation of power load influencing factors is analysed using Pearson correlation coefficients. The seagull optimisation algorithm is overcome to fall into local optimality by using the random adaptive non-linear adjustment strategy of manipulated variables and the differential variational Levy flight strategy, which improves the search efficiency of the algorithm. Then, the The gated cyclic unit hidden layer parameters are optimised by the improved seagull optimisation algorithm to construct a short-term electricity load forecasting model.The effectiveness of the proposed method is verified by simulation experimental analysis. The results show that the proposed method has improved the accuracy of the forecasting model.
CONCLUSION: The CEEMD method is used to decompose the original load time series, which improves the accuracy of the measurement model. The GRU prediction model based on improved SOA optimization not only has better prediction accuracy than other prediction models, but also consumes the least amount of time compared to other prediction models.

Keywords: short-term electricity load forecasting, gated cyclic unit, seagull optimization algorithm, complete ensemble empirical modal decomposition

Received on 03 March 2024, accepted on 05 April 2024, published on 15 April 2024

doi: 10.4108/ew.5282

*Corresponding author. Email: 2875177934@qq.com

1. Introduction

As China's economic level improves, industrial electricity consumption soars, electricity demand grows exponentially year by year, and the amount and complexity of electricity generated on the supply side of electricity also increases proportionally, which in turn makes the management and dispatch of complex and huge power systems increasingly tricky [1]. How to balance the power supply and demand between the grid and users is related to the safe, stable and economic operation of the grid, which is a very important aspect of power system management and has become a key concern for the power sector. In the power industry, power load forecasting not only provides useful basic data information for the power management system, but also can effectively guarantee the operation and maintenance of the power system [2]. According to the time factor, power system load forecasting is generally divided into ultra-short-term forecasting, short-term forecasting, medium-term forecasting and long-term forecasting [3]. Short-term forecasts are mainly used for generating units to be normal and to develop dispatch plans for hours or weeks for the economic allocation of the grid. Accurate short-term telephone forecasts can largely reduce power system losses and operating costs [4]. Therefore, it is an unavoidable task and an urgent and important research topic to construct a reasonable and highly accurate short-term power load forecasting model [5].

The complexity of the power network, changes in weather conditions, geographic location, and holiday activities combine to affect the normal operation of the electric load, which has the characteristics of non-smoothness, randomness, seasonality and high volatility, so how to build an accurate short-term electric load forecasting model becomes the key to the normal operation and maintenance of electricity [6]. Currently, short-term power load forecasting methods mainly include traditional forecasting methods, modern forecasting methods and combined forecasting methods [7]. Traditional forecasting methods include trend analysis [8], regression analysis [9] and time series analysis [10]. The trend analysis method is to judge the future load development trend by observing the curve fluctuation, but it does not consider the collection errors in the process and there are large differences; the regression analysis method is applied to electricity load forecasting and will be affected by different data stages; the time series analysis method is easily affected by the changes of random factors and the load forecasting results may be unsatisfactory. Modern forecasting methods include artificial neural networks [11], grey theory forecasting method [12-13], fuzzy forecasting method [14], etc. The literature [11] used an improved aspen whisker search algorithm to optimise the BP neural network to build a short-term load forecasting model for power plants, which reduced the forecasting error and provided a basis for optimal load allocation of power plant units; the literature [12] used a grey model to build a short-term load forecasting model, and the forecasting accuracy was obviously improved; the literature [13] combined the grey model and BP neural network to further improve the forecasting of a single model The accuracy of a single model was further improved by combining grey model and BP neural network. From the above literature, it can be seen that both traditional and modern forecasting methods lack generalization and do not satisfy all kinds of electric load forecasting problems simultaneously. As the electricity load presents non-smooth, random and high fluctuation characteristics, the current single modern forecasting method is less stable and cannot meet the accuracy demand, and the combined forecasting method combines multiple models, which can avoid the problems existing in a single model [15]. The literature [16] used the aspen whisker search algorithm to optimise the support vector machine parameters to enhance the training performance of the prediction model and improve the accuracy of electricity load forecasting; the literature [17] analysed the electricity load historical data by Kmeans algorithm and used BP neural network to construct the electricity load forecasting model, and the results showed that the combined forecasting method was better than the single forecasting method; the literature [18] proposed a flower pollination based algorithm to optimize the data method of variational modal decomposition, and considering the load information of historical moments and the load information of future moments, the two-way LSTM neural network was used to construct the electric load prediction model, and the simulation experiment results showed that the proposed method improved the accuracy of electric load prediction.

With the rapid development of computing technology, artificial intelligence algorithms continue to break through the limitation, especially neural networks with their powerful multivariate mapping capability are widely used in non-linear prediction problems and can obtain better prediction accuracy [19]. Recurrent neural network (RNN) [20] can update the current neuron state by using the previous moment's neuron state, solving the analysis of time series data prediction. Long short term memory (LSTM) [21], as an improved network of RNN, solves the problems caused by gradient explosion and gradient disappearance. Gated recurrent unit neural (GRU) [22] simplifies the network structure by adding gate structure control on the basis of LSTM, which can better process and mine the time series.GRU uses back propagation algorithm to update the network parameters, which is easy to fall into local optimum and makes the power load prediction model less stable.

In order to overcome the problem of GRU falling into local optimum, better learn the non-linearity, randomness and volatility of electricity load data, and further improve the short-term electricity load forecasting accuracy, this paper adopts a hybrid decomposition-optimization-integration forecasting model. (1) decompose the original electricity price data using the complete ensemble empirical modal decomposition method to improve the prediction model's extraction of the change pattern of the load sequence; (2) analyse the relevance of the factors influencing electricity load using Pearson correlation coefficients and screen the factors with strong relevance to electricity load as inputs to the prediction model; (3) combine the manipulated variable stochastic adaptive nonlinear adjustment strategy and the differential variance Levy flight strategy, the seagull optimization algorithm is improved, and a short-term electricity load forecasting method based on the improved optimization algorithm to optimize the gated recurrent unit neural network is also proposed; (4) the electricity load data are used to verify that the method in this paper has higher forecasting accuracy.

2.CEEMD-based time series decomposition of electrical loads

In order to better find the laws of power load time series and improve the accuracy of short-term load prediction, this paper adopts the fully ensemble empirical mode decomposition method to pre-process the predicted time series [23]. The classical empirical mode decomposition (EMD) method is a time signal decomposition algorithm, which is mainly widely used for non-smooth non-linear time series signals, but there are cases of confounding of eigenmodal functions. Ensemble EMD (EEMD) is an improved method of EMD algorithm, in which uniformly distributed white noise is added to the load time series several times to overcome the problem of mode overlap, but there is a problem of white noise residual. In order to solve the above problems, this paper chooses Complementary EEMD (CEEMD) to analyse and process the power load time series [24]. CEEMD makes the white noise residual smaller and the computational cost lower by adding two mutually opposite white noises to the load time series and decomposing them using the EEMD method [24]. The main process is as follows.

1) Add the positive and negative opposite white noise sequences $\xi_i^+(t)$ and $\xi_i^-(t)$ to the original load time series $y(t)$ to obtain the additive noise time series, i.e.

$$\begin{cases} Y_i^+(t) = y(t) + \xi_i^+(t) \\ Y_i^-(t) = y(t) + \xi_i^-(t) \end{cases} \tag{1}$$

where $Y_i^+(t)$ and $Y_i^-(t)$ are the time series after adding positive and negative white noise, respectively.

2) Determine the upper and lower envelope sequences of $Y_i^+(t)$ and $Y_i^-(t)$ using cubic spline interpolation to obtain the mean upper and lower envelope sequences of $Y_i^+(t)$ and $Y_i^-(t)$ $m_1^+(t)$ and $m_1^-(t)$, and the time series to be decomposed $h_1^+(t)$ and $h_1^-(t)$ are calculated as follows

$$\begin{cases} h_1^+(t) = Y_i^+(t) - m_1^+(t) \\ h_1^-(t) = Y_i^-(t) - m_1^-(t) \end{cases} \tag{2}$$

3) Decompose the time series $h_1^+(t)$ and $h_1^-(t)$ using the EMD decomposition method to obtain the first eigenmode functions $imf_1^+(t)$ and $imf_1^-(t)$, and then subtract the eigenmode functions using $Y_i^+(t)$ and $Y_i^-(t)$ to obtain the new signal series $r_i^+(t)$ and $r_i^-(t)$, calculated as follows

$$\begin{cases} r_1^+(t) = Y_i^+(t) - imf_1^+(t) \\ r_1^-(t) = Y_i^-(t) - imf_1^-(t) \end{cases} \tag{3}$$

4) Repeat step 3) to obtain a number of eigenmodal function components and a residual component for $Y_i^+(t)$ and $Y_i^-(t)$, respectively.

$$\begin{cases} y_i^+(t) = \sum_{j=1}^{J} imf_j^+(t) + res^+(t) \\ y_i^-(t) = \sum_{j=1}^{J} imf_j^-(t) + res^-(t) \end{cases} \tag{4}$$

where J is the number of eigenmodal functions, and $res^+(t)$ and $res^-(t)$ are the residuals of the positive and negative noise series decomposition, respectively.

(5) Each eigenmode function component and residual component of $Y_i^+(t)$ and $Y_i^-(t)$ are averaged to obtain the final eigenmode component and residual component of the original load time series $y(t)$. The final objective function is given by

$$\chi(t) = res(t) + \sum_{j=1}^{J} imf_j(t) \tag{5}$$

where $res(t)$ is the average sequence of the residual components of the positive and negative additive noise sequence decomposition, and $imf_j(t)$ denotes the average sequence of the *jth* eigenmodal component of the positive and negative additive noise sequence decomposition.

3. Analysis of the factors influencing the electrical load

In order to establish a more accurate short-term model of electric load, this paper uses the load history data of a customer in a certain region to analyse the correlation of the factors influencing electric load. The Pearson correlation coefficient is used to analyse the correlation of the factors influencing the electricity load, which is calculated as follows.

$$\rho(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{6}$$

Where $\mathrm{cov}(X,Y)$ is the covariance between the variables X and Y , is the standard deviation of the variable σ_X X , σ_Y is the standard deviation of the variable Y , and $\rho(X,Y)$ is the Pearson correlation coefficient between the variables X and Y, with values ranging from $[-1,1]$. When , the variables $\rho=0$ X and Y are uncorrelated; when , the variables $0<\rho\leq 1$ X and Y are positively correlated; when $-1\leq\rho<0$, the variables X and Y are negatively correlated.

In order to analyse the historical load internal correlation, this section selects a number of days before the number of relevant moments and the load at a number of moments before that day as internal influences for Pearson correlation analysis. Assuming that the load at point *t on* day *i* is $P(i,t)$, the loads at six moments ahead on day *i are* represented by $P(i,t-1)$, $P(i,t-2)$, $P(i,t-3)$, $P(i,t-4)$, $P(i,t-5)$ and $P(i,t-6)$ *respectively, and the loads at moment t* one day ahead, moment *t* two days ahead, moment *t* three days ahead and moment *t* one week ahead on day *i are* represented by $P(i-1,t)$, $P(i-2,t)$, $P(i-3,t)$ and $P(i-7,t)$ respectively. The heat map for correlation analysis between $P(i,t)$ and the selected 10 historical loads is shown in Figure 1.

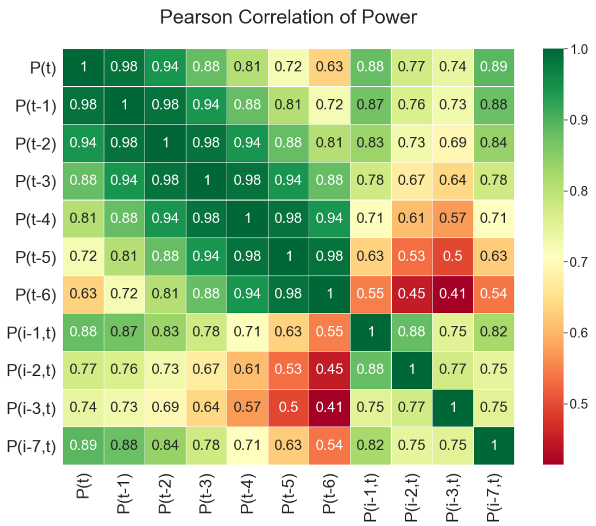


Figure 1. Correlation analysis of load at the tth point on day *i* and historical time

As can be seen from Figure 1, $P(i,t-1)$, $P(i,t-2)$, $P(i,t-3)$, and $P(i,t-4)$ $P(i,t)$ have a strong correlation, indicating that the load at time *t of* the day is similar to the load at the adjacent time with less fluctuation; $P(i-1,t)$, $P(i-2,t)$, $P(i-3,t)$, $P(i-7,t)$ and $P(i,t)$ have a strong correlation, indicating that the load at time *t of* the day is similar to the load at time *t of* the adjacent day and the load at time *t* of the adjacent week.

4. Gated cyclic cell prediction network based on improved seagull optimization algorithm

Since GRU neural networks suffer from the problem of falling into local optima, this paper uses an improved seagull optimization algorithm to search for the optimal structure of GRU neural network predictions.

4.1. Gated recurrent unit neural networks

The GRU network is an advanced version of the LSTM with a simpler structure, fewer parameters, and also solves the gradient vanishing problem of RNNs, and GRU also introduces a gate structure, consisting of update and reset gates. The model is as follows.

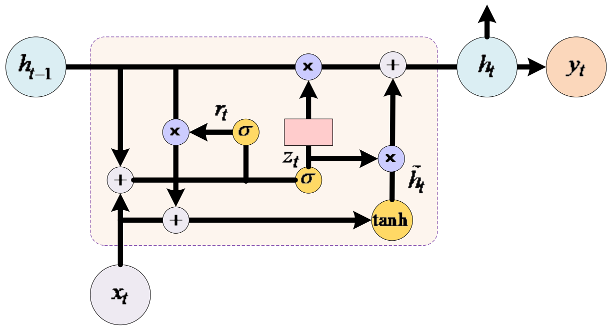


Figure 2 The GRU network

$$r_t = \sigma(W_{hr}h_{t-1} + W_{xr}x_t + b_r) \tag{7}$$

$$\tilde{h}_t = \tanh(W_{r\tilde{h}} \cdot (r_t * h_{t-1}) + W_{x\tilde{h}}x_t + b_{\tilde{h}}) \tag{8}$$

where r_t is the reset gate, which determines how much of h_{t-1} 's history memory is retained. $\tilde{h}_t$ is the latest information of the candidate hidden layer at the current moment. h_{t-1}, h_t is the hidden layer information for the cell state at $t-1$ and t respectively, $W_{r\tilde{h}}$, $W_{x\tilde{h}}$, W_{xr}, W_{hr} are the weights, and b_r and $b_{\tilde{h}}$ are the biases.

$$z_t = \sigma(W_{hz}h_{t-1} + W_{xz}x_t + b_z) \tag{9}$$

$$h_t = (1-z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{10}$$

where W_{hz} and W_{xz} are the weights and b_z is the bias. z_t is the forgetting gate, which is used to combine the input hidden layer information h_{t-1} from the previous moment, with the candidate hidden layer information $\tilde{h}_t$ from the current moment, to get the output cell hidden layer information h_t . When $z_t = 0$, the hidden layer directly outputs the previous hidden layer information h_{t-1} , when $z_t = 1$, the candidate hidden layer directly outputs the current hidden layer information h_t .

$$y_t = \sigma(W_{yt} h_t) \tag{11}$$

Where W_{yt} indicates the weight between the current hidden layer output h_t and the final output layer.

4.2 Improved seagull optimization algorithm

The Seagull optimization algorithm (SOA) [25] is an optimization algorithm inspired by animal behaviour proposed by Gaurav Dhiman in 2018.SOA simulates the migratory behaviour and aggressive behaviour of seagulls. Migratory behaviour is the behaviour of gulls approaching towards the optimal gull position while avoiding collision and searching for prey throughout space, mainly simulating the exploration process of the algorithm. Attack behaviour is the process by which gulls approach their prey using spiral moving flight manoeuvres, mainly simulating the algorithm's exploitation process. The inspired behaviour of the seagull optimisation algorithm is shown specifically in Figure 3.

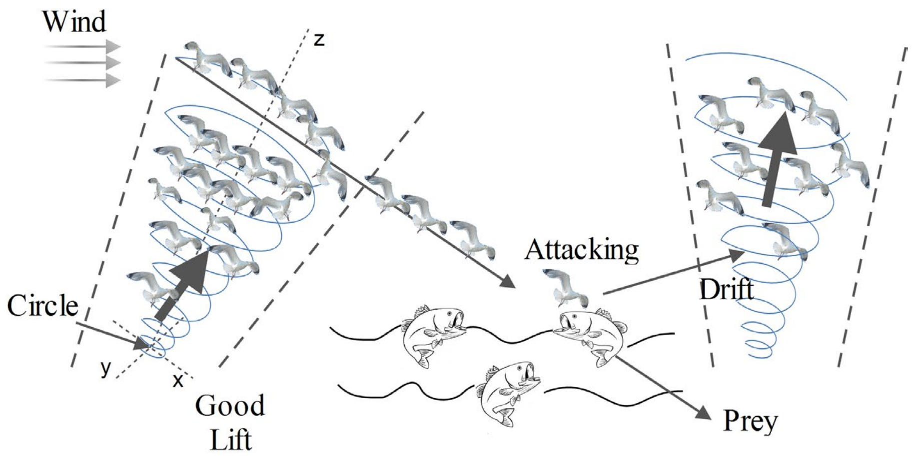


Figure 3 Seagull optimization algorithm heuristic behavior

(1) Basic seagull optimization algorithm

1) Migration behaviour

During migration, the location renewal of gull populations satisfies three main conditions: avoiding collisions between gulls, approaching towards the best individual gull, and location renewal through the best individual gull.

To avoid a collision between two gulls, calculate the position of the individual gull after it has moved using the variable A.

$$\boldsymbol{C}_s = A \times \boldsymbol{P}_s(t) \tag{12}$$

Where, $\boldsymbol{C}_s$ represents the position of the gull searching for individuals that do not collide with other individuals, $\boldsymbol{P}_s$ represents the current position of the gull individual, t represents the number of iterations, and variable A represents the movement behaviour of the individual, calculated as follows.

$$A = f_c - \left(t \cdot f_c / Max_{iteration}\right) \tag{13}$$

Where f_c is used to control the frequency of the manipulation variable A, usually set to 2. $Max_{iteration}$ indicates the maximum number of iterations. The value of variable A decreases from 2 to 0 as the number of iterations increases.

After avoiding collisions between neighbouring individuals, the search individual moves towards the optimal individual, as expressed by the following equation.

$$\boldsymbol{M}_s = B \times \left(\boldsymbol{P}_{bs} - \boldsymbol{P}_s\right) \tag{14}$$

where $\boldsymbol{M}_s$ represents the new position of the individual gull after it has moved towards the optimal gull individual $\boldsymbol{P}_{bs}$. B is a random number that is primarily used to balance the exploration and exploitation capabilities of the algorithm and is calculated as follows.

$$B = 2 \cdot A^2 \cdot rand \tag{15}$$

where $rand$ is a random number with a value between 0 and 1.

Finally, the equation for the distance the search individual position moves relative to the optimal individual is as follows.

$$\boldsymbol{D}_s = \left| \boldsymbol{C}_s + \boldsymbol{M}_s \right| \tag{16}$$

where $\boldsymbol{D}_s$ indicates the distance between the searched individual and the optimal individual.

2) Aggressive behaviour

The process of developing an optimised algorithm for simulating attack behaviour makes use of historical experience during the search process. The gull attacking prey process is mastered by wings and weight in the direction of the attack. In general, the gull attacking prey process is generally simulated using a spiral movement action, which is calculated as follows.

$$\boldsymbol{P}_s(t+1) = \left(\boldsymbol{D}_s \times x \times y \times z\right) + \boldsymbol{P}_{bs}(t) \tag{17}$$

$$x = r \times \cos(k) \tag{18}$$

$$y = r \times \sin(k) \tag{19}$$

$$z = r \times k \tag{20}$$

$$r = u \times e^{kv} \tag{21}$$

Where $\boldsymbol{P}_s(t+1)$ indicates the individual seagull after the position update. r denotes the radius of rotation of the spiral, k denotes a random number between 0 and 2π u and v denotes the constant that defines the spiral action and e is the base of the natural logarithm. The gull optimisation algorithm uses a random method to generate the population and updates the gull population position by transforming the global search and local search process to ultimately search for the best individuals.

(2) Stochastic adaptive non-linear adjustment strategy for manipulated variables

The main role of the classical SOA algorithm manipulation variable A is to regulate the balancing algorithm global exploration and local exploitation process. When $A \geq 1$, the gull population will expand the collision avoidance distance, meaning that a wider space will be explored in order to globally explore better search locations, while when $A < 1$, the gull population will reduce the lead distance, meaning that a more precise area will be exploited in order to locally search for better prey locations. The manipulation variable A of the classical SOA algorithm decreases linearly from 2 to 0 as the number of iterations increases. At the beginning of the algorithm iteration, the manipulation variable A has a larger value, making the individual gull move a larger distance, which is beneficial for global exploration; at the end of the algorithm iteration, the manipulation variable A has a smaller value, making the individual gull move a smaller distance, which is beneficial for local exploitation. In realistic optimisation problems, the algorithm search process is more responsible and the linear adjustment strategy of manipulated variables cannot be adapted to realistic problems. Therefore, in order to adapt to complex optimisation problems and better balance global exploration with local exploitation capabilities, a stochastic adaptive non-linear adjustment strategy is proposed in this paper. This strategy combines stochastic distribution methods, adaptive adjustment methods and non-linear decreasing methods to improve the calculation of the manipulated variables A , which are calculated as follows.

$$A = f_{ada} \cdot \left(1 - \sin\left(\frac{\pi}{2} \cdot \frac{t}{Max_{iteration}}\right)\right) \tag{22}$$

$$f_{c-ada} = f_{c\min} + M \cdot \left(f_{c\max} - f_{c\min}\right) + \sigma \cdot randn \tag{24}$$

$$M = \frac{f_{\max} - \overline{f}}{\overline{f} - f_{\min}} \tag{25}$$

$$\sigma = E\left(\left(\tilde{f} - E\left(\tilde{f}\right)\right)^2\right) \tag{26}$$

Where, A is the SOA algorithm ground manipulation variable. f_{c-ada} is the stochastic adaptive used to control the frequency, the calculation formula (24) is mainly divided into the adaptive part and the random distribution part [26]. $f_{c\min}$ and $f_{c\max}$ denote the minimum and maximum values of the control frequency, which generally take the values of 0 and 2. The adaptive part uses the value of the fitness function to calculate the control frequency, as shown in Equation (25), where M is the control frequency adaptive parameter, and $f_{\max}$, $f_{\min}$ and $\overline{f}$ denote the maximum, minimum and average fitness values, respectively; when M is larger and f_{c-ada} is larger at the beginning of the iteration, the algorithm performs a global exploration search; when M is smaller and f_{c-ada} is smaller at the end of the iteration, the algorithm performs a local exploration search. The random distribution is calculated using the variance of the normalised fitness value, where σ is the variance, $randn$ is the random number that follows a normal distribution, and $\tilde{f}$ represents the normalised fitness value.

(3) Differential variant Levy flight strategy

The classical SOA algorithm attack behaviour mainly simulates local mining capabilities. For complex multimodal optimization problems, the SOA algorithm suffers from the problem of falling into a local optimum in the late iteration. Therefore, in order to increase the diversity in the late search stage and avoid the population falling into local optimum, this paper uses the differential variational Levy flight strategy to improve the SOA algorithm attack behaviour by improving the specific formulation as follows.

$$\boldsymbol{P}_s(t+1)=(\boldsymbol{D}_s\times x\times y\times z)+\boldsymbol{P}_{DE-Levy}(t) \quad (27)$$

$$\boldsymbol{P}_{DE-Levy}(t)=\boldsymbol{P}_{bs}(t)+randn\cdot Levy(\boldsymbol{P}_s(t))+randn\cdot\left|\boldsymbol{P}_{r1}(t)-\boldsymbol{P}_{r2}(t)\right| \quad (28)$$

where $\boldsymbol{P}_{DE-Levy}(t)$ denotes the distance travelled based on the differential variance Levy flight strategy. $\boldsymbol{P}_{bs}(t)$ denotes the optimal individual gull position. $Levy(\boldsymbol{P}_s(t))$ denotes the current individual Levy flight distance. Levy flights have small flight steps over long periods of time, occasionally producing longer flight steps to increase flight diversity. the Levy flight model is specifically calculated as shown in (29). $randn$ denotes random numbers that obey a normal distribution. $\left|\boldsymbol{P}_{r1}(t)-\boldsymbol{P}_{r2}(t)\right|$ denotes the difference variable between individuals $\boldsymbol{P}_{r1}(t)$ and $\boldsymbol{P}_{r2}(t)$, where $r1\neq r2\neq s$.

The Levy flight model is specifically calculated as follows.

$$Levy(\boldsymbol{P}_s(t))=\alpha\cdot s\cdot(\boldsymbol{P}_{bs}(t)-\boldsymbol{P}_s(t)) \quad (29)$$

where α represents the scale factor and takes the value $[-1,1]$; s is the random wandering step, calculated as follows.

$$s=\frac{u}{|v|^{1/\beta}} \quad (30)$$

$$\sigma_u=\left[\frac{\Gamma(1+\beta)\cdot\sin(\pi\cdot\beta/2)}{\Gamma((1+\beta)/2)\cdot\beta\cdot 2^{(\beta-1)/2}}\right]^{1/\beta} \quad (31)$$

$$\sigma_v=1 \quad (32)$$

Where u and v are parameters that follow a normal distribution, i.e. $u\sim N(0,\sigma_u^2)$, $v\sim N(0,\sigma_v^2)$, $\Gamma(\cdot)$ are gamma functions.

4.3. GRU prediction method based on improved SOA algorithm

The GRU neural network prediction model based on the Improved SOA (ISOA) algorithm is mainly divided into a data module, an optimisation weight module and a gated recurrent unit neural network module (GRU module). The GRU module uses the ISOA optimisation parameters to decode into weights, thus building the GRU network; it then uses the incoming training data from the data module to train the GRU; the optimal use of the test set for prediction, to obtain the error between the expected value and the actual output value.

(1) Coding method

In order to improve the accuracy of the GRU neural network, the parameters of the GRU neural network are optimized using the improved ISOA algorithm, i.e., the weights and biases of the optimized neural network, and the parameters are encoded using the real number encoding method.

(2) Adaptability function

In order to accurately reflect the strengths and weaknesses of the trained GRU network, the root mean square error (RMSE) is used as the fitness function in this paper and is calculated as follows.

$$RMSE=\sqrt{\left(\sum_{i=1}^{M}(\hat{y}_i-y_i)^2\right)\Big/M} \quad (33)$$

Where M is the number of observed samples, $\hat{y}_i$ and y_i represent the true and predicted values of the sample respectively. i

(3) ISOA-GRU method

According to the coding method and the fitness function, the steps of the GRU neural network prediction method based on the improved seagull optimization algorithm are as follows.

Step 1: Pre-processing and normalization of the raw data into a test set and a training set.

Step 2: The improved SOA algorithm encodes the weights and bias parameters of the GRU neural network, while initializing the algorithm parameters such as population parameters and number of iterations; calculating the value of the fitness function according to equation (33).

Step 3: Calculate the manipulated variable A using the stochastic adaptive non-linear adjustment strategy and calculate the distance between the search individual and the optimal individual according to equation (16) $\boldsymbol{D}_s$;

Step 4: Calculate the distance travelled using the differential variance Levy strategy $\boldsymbol{P}_{DE-Levy}$;

Step 5: Update the individual gulls, calculate the value of the fitness function and update the global optimal solution and the individual optimal solution.

Step 6: Determine whether the termination condition is satisfied. If it is satisfied, exit the iteration, output the optimal network parameters and execute step 7, otherwise continue with step 3.

Step 7: decoding the parameters of the ISOA-based optimization network and obtaining the weights and biases of the GRU neural network.

Step 8: Build the ISOA-GRU network, train the network with the training set to get the prediction model, input the test set into the prediction model and get the prediction results.

5. Short-term load forecasting method for electricity based on improved SOA algorithm to optimize GRU

Applying the ISOA-GRU forecasting method proposed in this paper to the short-term electricity load forecasting problem.

Step 1: Decompose the original power load time series using the Complete Ensemble Empirical Modal Decomposition method (CEEMD) to obtain $K+1$ components $\{imf_1,\cdots,imf_i,\cdots,imf_k,Res\}$;

Step 2: For the decomposed components, combine the inputs of external factors affecting the variation of electricity load and construct a GRU neural network based on ISOA optimization to build a prediction model. Optimisation of the parameters of the GRU neural network using the improved SOA algorithm to select the optimal GRU neural network parameters.

Step 3: Input the test set of each component to the component prediction model, output to obtain the prediction results of each component, and obtain the final total prediction results by superposition reconstruction.

Step 4: The performance of the prediction model proposed in this paper is analysed by comparing the evaluation metrics with other methods.

6. Experimental simulation analysis

In order to verify the performance of the short-term electricity load forecasting method proposed in this paper, the forecasting of the proposed algorithm is analysed and discussed in this section by selecting a customer electricity load dataset from a region in South America.

6.1. Simulation environment setup

This paper was programmed using MATLAB 2021a with a test environment of Windows 10, a processor of AMD Ryzen 9 5900HX with Radeon Graphics and 16.0 GB of RAM.The experimental dataset was selected from the electricity load dataset of a customer in a region of South America from May 1 to June 29, 2018 data [19] as the training set of the prediction model, and the 24h electricity price on June 30, 2018 as the test set of the prediction model. The specific parameter settings of the electricity load forecasting algorithm proposed in this paper and the comparison forecasting method are shown in Table 1. CEEMD-ISOA-GRU uses the CEEMD decomposition method [24], and its parameters are set as follows: the number of white noise added is 40, and the standard deviation is taken as 0.1; EMD-ISOA-GRU uses the EMD decomposition method; the number of optimization iterations of the SOA and ISOA algorithms is set as 50.

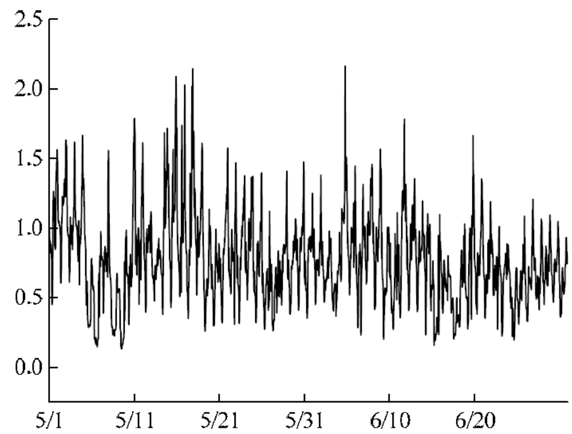


Figure 4. Raw data on electrical loads

Table 1 Power load forecasting methods parameter settings

Predictive models	Decomposition algorithm	Prediction algorithm parameter setting
LSTM	None	The number of hidden layer nodes is 50 and Adam optimally adjusts the weights
GRU	None	The number of hidden layer nodes is 50 and Adam optimally adjusts the weights
SOA-GRU	None	The number of hidden layer nodes is 50 and the SOA population is 50
ISOA-GRU	None	The number of cryptospheric nodes is 50 and the ISOA population is 50
EMD-ISOA-GRU	EMD	The number of cryptospheric nodes is 50 and the ISOA population is 50
CEEMD-ISOA-	CEEMD	The number of cryptospheric nodes is 50 and the ISOA population is 50

Predictive models	Decomposition algorithm	Prediction algorithm parameter setting
GRU		

6.2. Evaluation indicators

In order to fairly analyse the forecasting performance of each forecasting model, this paper uses the mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE), where MAE and MAPE are calculated as follows.

$$MAE = \frac{1}{M}\sum_{i=1}^{M}\left|\hat{y}_i - y_i\right| \tag{34}$$

$$MAPE = \frac{1}{M}\sum_{i=1}^{M}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{35}$$

6.3. Analysis of simulation experiments

In order to improve the accuracy of the ISOA-GRU forecasting model, this paper uses the CEEMD algorithm to decompose the original electricity load time series, and the decomposed electricity price series by the CEEMD algorithm is given in Figure 5. As can be seen from Figure 5, the volatility of the electricity load data is better decomposed by using the CEEMD algorithm for decomposition.

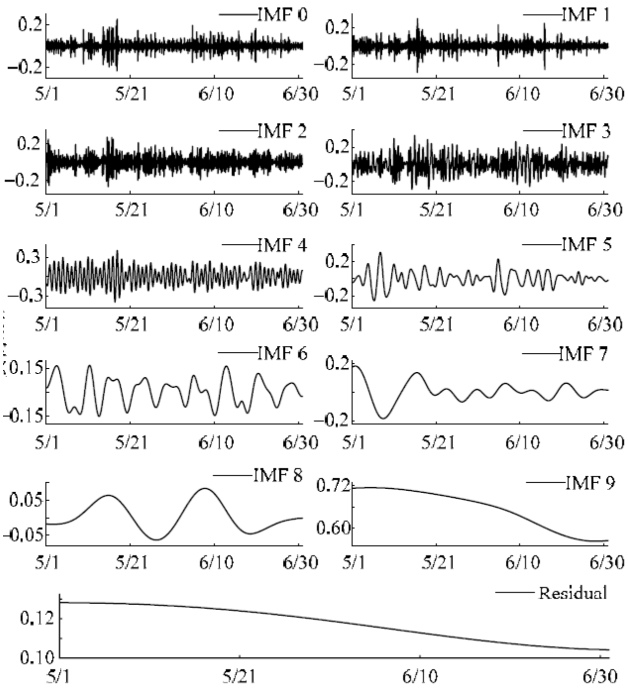
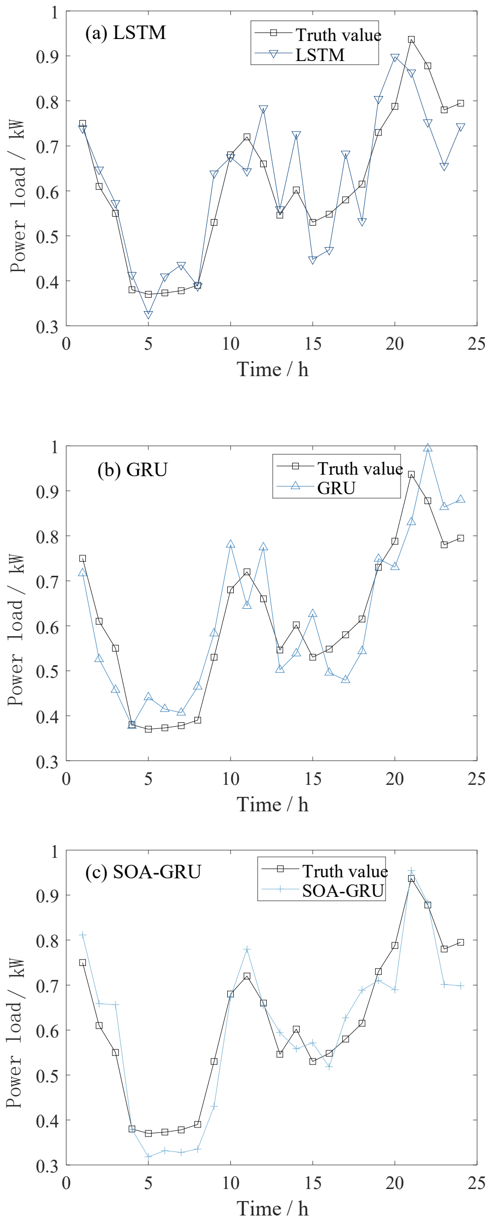


Figure 5 Sequence of electrical loads after decomposition by the CEEMD algorithm

In order to verify the effectiveness of the model proposed in this paper, CEEMD-ISOA-GRU was compared with the other five models, and the load prediction results and relative prediction errors of each model are shown in Figure 10 and Figure 11. By comparing the prediction results in Figure 6(a) and (b), the prediction results of GRU are closer to the true values, thus indicating that the prediction performance of GRU neural network is better than that of LSTM; by comparing the prediction results in Figure 6(b), (c) and (d), the prediction results of GRU neural network based on ISOA optimization are closer to the true values, thus indicating that the prediction performance of ISOA-GRU neural network better than the other two models; comparing the prediction results in Figure 6(d), (e) and (f), the CEEMD power load sequence decomposition has improved the model accuracy to a certain extent.



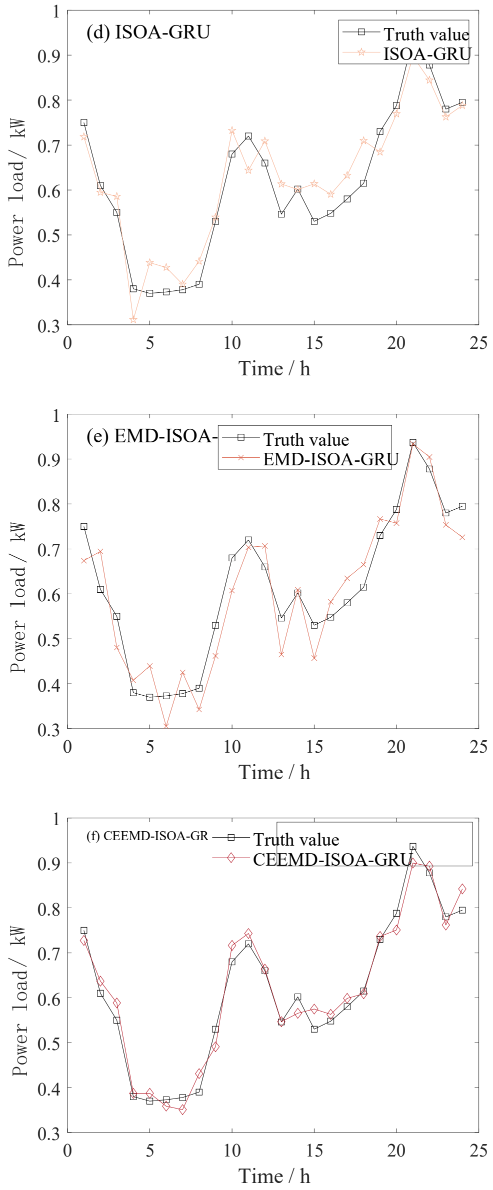


Figure 6 Comparison of the real value of the electricity load with the results of the forecasts based on each method

From Figure 7, it can be seen that the relative error between the prediction results of the CEEMD-ISOA-GRU model and the true value is smaller at 4h, 12h~13h, 18h~19h and larger at 15h, 24h. the relative error of the prediction of the EEMD-ISOA-GRU model performs worse at 1~10h, 13h, 15h, 24h, the relative error of the ISOA-GRU model relative error performed worse at 4~5h, 13h, 15h, 19h, the SOA-GRU model performed worse at 3h, 9h, 18h, 20h, 23h~24h, the GRU neural network's relative error was larger at 2h~3h, 11h~13h, 15h, 17h, 20h~24h, the LSTM neural network's relative error was larger at 9h, 11h~12h and 14h~24h. In summary, the CEEMD-ISOA-GRU model has the smallest prediction error overall.

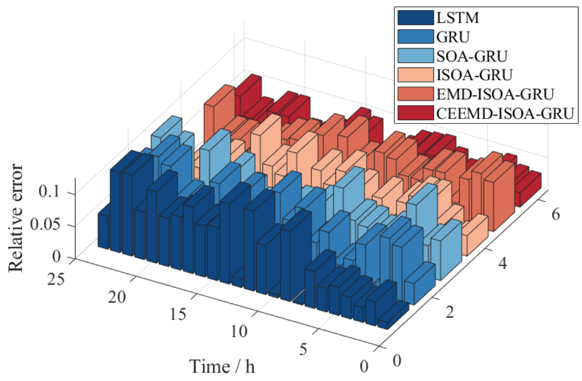


Figure 7 Relative error of short-term forecasts by method

In order to effectively compare the forecasting effectiveness of each model, the mean values of MAE, RMSE, MAPE and forecasting time (Time) for 10 independent runs of each model were counted. The prediction results of each model are given in Table 2. As shown in Table 2, in terms of MAE, RMSE and MAPE metrics, CEEMD-ISOA-GRU has the best performance, followed by EEMD-ISOA-GRU; in terms of Time metrics, ISOA-GRU has the best performance, mainly because the improved SOA algorithm is more efficient than the original algorithm in terms of optimisation, and the decomposition makes the prediction model take more time to compute. The CEEMD-ISOA-GRU MAE, RMSE and MAPE values are 0.0242, 0.0279 and 20.44% respectively, and the prediction time is 4.37e-03 seconds, with high accuracy and real-time performance to meet the prediction demand.

In terms of MAE, RMSE, MAPE, and Time metrics, CEEMD-ISOA-GRU had the best performance, followed by CEEMDAN-GWO-DELM; in terms of MAE and RMSE, IGWO-DELM had the worst results; in terms of MAPE, CEEMDAN-LSSVM had the worst results; and in terms of prediction time, CEEMDAN The MAE, RMSE and MAPE values of CEEMDAN-IGWO-DELM were 3.21, 4.30 and 21.46%, respectively, and the prediction time was 4.37e-05 seconds, which is high accuracy and good real-time performance.

Table 2. Comparison of 10 predictions across models

No.	Algorithms	MAE	RMSE	MAPE/%	Time/s
1	LSTM	0.0668	0.0779	33.01	2.32e-02
2	GRU	0.0626	0.0758	30.96	8.63e-03
3	SOA-GRU	0.0494	0.0582	29.32	4.79e-03
4	ISOA-GRU	0.0428	0.0497	28.02	**1.17e-03**
5	EMD-ISOA-GRU	0.0409	0.0447	30.02	6.57e-03
6	CEEMD-ISOA-GRU	**0.0242**	**0.0279**	**20.44**	4.37e-03

7. Conclusion

In order to further improve the accuracy of the short-term load forecasting method for electricity, this paper proposes a short-term load forecasting method based on an improved optimisation algorithm to optimise gated recurrent unit neural networks through a hybrid decomposition-optimisation-integration forecasting framework. The method decomposes the original load time series using a complete ensemble empirical modal decomposition method; uses Pearson correlation coefficients to correlate the factors influencing the electricity load; adopts a random adaptive non-linear adjustment strategy for manipulated variables and a differential variational Levy flight strategy to enhance the population diversity of the original SOA algorithm, improve the optimization-seeking accuracy, and improve the algorithm's generalized optimization-seeking capability; uses the The improved SOA algorithm is used to optimize the gated recurrent unit neural network and construct a short-term load forecasting method. Through simulation, the following conclusions are drawn.

(1) The CEEMD method was used to decompose the original load time series, which improved the accuracy of the measurement model.

(2) The GRU forecasting model based on improved SOA optimisation is not only better than other forecasting models in terms of forecasting accuracy, but also less costly than other forecasting models.

The prediction model proposed in this paper does not perform well in some moments and a heterogeneous prediction model based on decomposition-optimisation-integration is the next research focus.

References

[1] Zheng T, Dai Zemei, Yao Jiahao, et al. Integrated energy system control freedom index and its impact on economic dispatch[J]. China Electric Power, 2021(3): 1-17.

[2] Deng Y.S., Jiao F.S., Zhang R.F., et al. A review of research on electric load forecasting methods in distribution network planning[J]. Electricity and Energy Efficiency Management Technology, 2019(14): 1-7.

[3] Le Haozhe. Research on short-term power load forecasting algorithm based on neural network hybrid model[D]. Nanchang: Nanchang University, 2021.

[4] Lin Jialiang, Li Nuanqun, Wu Yongfeng. Correlation analysis of short-term load forecasting accuracy in power systems[J]. Automation Applications, 2017(12): 125-127.

[5] Zhu JC, Yang ZL, Guo YJ, et al. A review on the application of deep learning in power load forecasting[J]. Journal of Zhengzhou University (Engineering Edition), 2019, 40(05): 13-22.

[6] Li B., Kong X. Y., Zhu S. J., et al. Short-term load forecasting for regional grids considering demand response in an interconnected energy environment[J]. Power System Automation, 2021, 45(01): 71-78.

[7] Xu Qing, Zhou Chao, Zhao Shuangshang, et al. Research on short-term electric load forecasting methods based on machine learning [J]. Electricity Measurement and Instrumentation, 2019, 56(23): 70-75.

[8] Zhong Ting. Research on short-term power load forecasting method based on EEMD decomposition and its application [D]. Xiangtan: Xiangtan University, 2020.

[9] Wang Yanling, Wu Mengkai. A medium- and long-term load forecasting model based on partial least squares regression under the new economic normal[J]. Power Automation Equipment, 2018, 38(03): 133-139.

[10] Song Yanru. Research and application of multi-objective optimization-based hybrid algorithm for load forecasting and machine learning [D]. Lanzhou: Lanzhou University, 2020.

[11] Song Jefei. Research and application of short-term load forecasting methods for power plants [D]. Xi'an: Xi'an University of Architecture and Technology, 2021.

[12] Yinsheng S, Chunxiao L, Bao L, et al. Principal Component Analysis of Short-term Electric Load Forecast Data Based on Grey Forecast[C]//Journal of Physics: Conference Series. IOP Publishing, 2020, 1486(6): 062031.

[13] Wang R, Zhou C X, Lu J. Short-term electricity load forecasting based on a combined model[J]. Computer Simulation, 2018, 35(12): 396-400.

[14] Zhang Yanyu, Xiao Xi. Analysis of the current status of research on load forecasting of power systems at home and abroad [J]. Shandong Industrial Technology, 2016(11): 215-208.

[15] Yang Yun, Cai Qiu Na, Zhang Qiao Yu, et al. Research on short-term load forecasting of new energy power systems based on combined forecasting models[J]. Automation Technology and Applications, 2019, 38(08): 102-106.

[16] Yan Chongxi, Chen Hao. Research on optimizing LSSVM short-term power load forecasting method based on improved tennies whisker search algorithm[J]. Electrical Measurement and Instrumentation, 2020, 53(06): 6-11.

[17] Liu Qianying, Ruan Yingjun, Shi Xiang, et al. Time-by-time electrical load prediction for office buildings based on Kmeans clustering and BP neural network algorithm[J]. Thermal Power Engineering, 2018, 33(03): 139-144.

[18] Zhang Shuqing, Li Jun, Jiang Anqi, et al. A novel two-stage short-term power load forecasting based on FPA-VMD and

BiLSTM neural networks[J]. Power Grid Technology, 2022, 46(08): 3269-3279.

[19] Deng Zaiyu, Li Jian, Zhang Zhenyuan, et al. Short-term power load forecasting based on EEMD-GRU-MLR [J]. Power Grid Technology, 2020, 44(02): 593-602.

[20] Napoli C, Pappalardo G, Tina G M, et al. Cooperative strategy for optimal management of smart grids by wavelet RNNs and cloud computing [J]. IEEE Transactions on Neural Networks & Learning Systems, 2016, 27(8): 1672-1685.

[21] Hao Q, Srinivasan D, Khosravi A. Short-term load and wind power forecasting using neural network-based prediction intervals [J]. IEEE Transactions on Neural Networks & Learning Systems, 2017, 25(2): 303-315.

[22] Zhu L, Li C, Guo L, et al. Short-term spatial and temporal load forecasting method based on GCN-GRU [J]. Energy and Environmental Protection, 2022, 44(04): 211-215.

[23] Jiang F, He JQ, Zeng ZG, et al. Electricity price forecasting based on decomposition-optimization-integration learning method[J]. Chinese Science: Information Science, 2018, 48: 1300-1315.

[24] Li Shaojie. Research on short-term power load forecasting based on FFT-CNN-LSTM [D]. Nanchang: Nanchang University, 2021.

[25] Yang Y. Research on feature selection algorithm based on binary seagull optimization algorithm [D]. Nanchang: Jiangxi University of Finance and Economics, 2020.

[26] Xu Songjin, Long W. Improved grey wolf optimization algorithm based on stochastic convergence factor and differential variance[J]. Science, Technology and Engineering, 2018, 18(23): 252-256.