IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 July 2023, accepted 1 October 2023, date of publication 5 October 2023, date of current version 11 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3322272

RESEARCH ARTICLE

Short-Term Power Load Forecasting Based on ICEEMDAN-GRA-SVDE-BiGRU and Error Correction Model

LIANBING LI[1], RUIXIONG JING[2], (Graduate Student Member, IEEE), YANLIANG ZHANG[2], LANCHAO WANG[2], AND LE ZHU[2]

[1]State Key Laboratory of Reliability and Intelligence of Electrical Equipment, Hebei University of Technology, Tianjin 300130, China
[2]School of Artificial Intelligence and Data Science, Hebei University of Technology, Tianjin 300130, China

Corresponding author: Ruixiong Jing (jingwork1999@foxmail.com)

ABSTRACT The significance of short-term power load forecasting extends to grid dispatching and future planning. To address the issues of nonlinear characteristics and poor prediction accuracy of original power load, a hybrid short-term power load forecasting method was proposed based on Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (ICEEMDAN), Grey Relation Analysis (GRA), Improved Secondary Variation Differential Evolution Algorithm (SVDE), Bidirectional Gated Recurrent Unit (BiGRU) and Error Correction Model. Firstly, ICEEMDAN decomposition is used to divide the sequence into Intrinsic Mode Functions (IMF) and a residual component (Res), and GRA is used to reconstruct the partial component sequences to improve the model operation efficiency and anti-interference ability. Then, an Improved Secondary Variation Differential Evolution Algorithm (SVDE) is proposed to perform hyperparameter optimization on BiGRU neural networks to predict the processed component sequences. Finally, an Error Correction Model based on SVDE-BiGRU is established by the processed mode components and factors such as temperature and holiday weekends to further increase the accuracy of its load prediction. The experimental results show that the RMSE, MSE, and MAPE of the prediction method are 89.72, 60.56, and 0.55% on average, respectively. Compared with the common BiGRU prediction method its MAE value is reduced by 79.02%. Compared with several mainstream methods, its MAE value is reduced by 70.88% at maximum and 40.62% at minimum, which proves the effectiveness and accuracy of the proposed method in short-term power load forecasting.

INDEX TERMS Differential evolutionary algorithm, short-term power load forecasting, error correction, bidirectional gated recurrent unit (BiGRU), ICEEMDAN, GRA.

I. INTRODUCTION

In recent years, with the continuous development of intelligent grid technology, accurate and efficient power load forecasting can provide a reliable basis for grid scheduling, thus ensuring the stable and efficient operation of the power system [1]. This has significant practical implications for energy utilization and the planning and scheduling of power systems [2]. According to the time horizon, power load forecasting can be classified into three categories: Long-term Load Forecasting [3], Medium-term Load Forecasting, and Short-term Load Forecasting [4], among which short-term load forecasting has better guidance for grid economic dispatch, power consumption planning and resource utilization [5].

According to the uncertainty of the grid load and nonperiodic factors such as weather and holidays, short-term power load forecasting methods are divided into three main categories: traditional forecasting methods [6], modern forecasting methods, and combined forecasting methods [7].

1) Traditional forecasting methods: traditional forecasting methods are based on mathematical statistical models, such as the peak load model method, regression analysis [8], the least squares method [9], and the gray model method. In [10],

The associate editor coordinating the review of this manuscript and approving it for publication was Behnam Mohammadi-Ivatloo.

fuzzy multiple linear regression was used for short-term forecasting of holiday loads. In [11], an improved autoregressive moving combination forecasting model was used for seasonal short-term and medium-term load forecasting, which is a simple and fast computational model, but the model lacks autonomous learning capability and has low prediction accuracy for complex nonlinear systems [12].

2) Modern forecasting methods: modern forecasting methods based on artificial intelligence, such as Support Vector Machine (SVM) [13], extreme learning machines, and Artificial Neural Networks (ANN) [14], [15], [16], mainly include Recurrent Neural Networks (RNN), Long Short-Term Memory neural networks (LSTM), Gated Recurrent Unit (GRU) neural networks, and support vector regression (SVR) and improved algorithms [17], [18]. In [19], a chaotic particle swarm algorithm was used to optimize support vector machine parameters with some improvement in short-term prediction accuracy. In [20], wavelet analysis and LSTM neural networks are combined in a deep learning architecture for short-term prediction. In [21], a combination of genetic algorithm and GRU neural network is used for prediction, which has a faster training speed and can be applied in load prediction with poor smoothness [22].

3) Combinatorial prediction methods: they are weighted and integrated by different models and algorithms into a combined forecasting model, decompose the load into multiple subseries before model prediction, and finally superimpose the prediction results of each series to achieve the final prediction value [23].

For nonlinear, nonstationary, and nonperiodic original power load, many scholars have started to combine data decomposition algorithms with neural networks for prediction as a way to enhance the prediction accuracy of forecasting models [24]. In [25], the data decomposition algorithm and neural network were combined to first decompose the original power load into individual subcomponents by Empirical Mode Decomposition (EMD), and then the LSTM neural network was used to predict the different components separately, and the forecasting accuracy of their combined model was verified by example to be high, but the LSTM neural network has many parameters and the training speed is slow [26]. In [27], the EMD algorithm and GRU neural network were composed into a hybrid prediction model for short-term load forecasting, and it was verified that the EMD-GRU hybrid model has greater prediction accuracy than a single model. However, the EMD algorithm exists issues such as pattern confounding and endpoint effects when decomposing the original power load, and it does not optimize the key parameters of the prediction model, which will affect the prediction accuracy and effectiveness when performing load forecasting [28], so there is still room for improvement in this regard. The GRU neural network only considers the influence of historical period influencing factors on the forecasted load, while the electric load is not only determined by the load influencing factors in the historical period but also associated with the influencing factors in the future period. Therefore, the GRU neural network cannot fully extract the effective potential relationship between the load characteristics data [29].

In addition to these, meteorological factors such as temperature are crucial for load forecasting. In recent years, the development of the weather forecasting (NWP) model has improved the accuracy and reliability of forecasting [30]. Meanwhile, some scholars have proposed prediction models using weather forecast correction neural networks [31], but the hybrid prediction effect of these weather forecast models and the original prediction models is not excellent.

The contributions and innovations of this paper are as follows:

- A novel forecasting method combining a hybrid prediction method and an error correction model is proposed, which can take advantage of the high accuracy of the combined forecasting method and consider the errors brought by external factors such as weather on the forecasts.
- Proposes an improved quadratic variational differential evolution algorithm (SVDE) to optimize the hyperparameters of the bidirectional gated recurrent unit (BiGRU) to enhance the prediction accuracy and speed.
- Verify the effectiveness and applicability of the method through example experiments and error criteria.

The remainder of this paper is organized as follows: Section II describes original power load decomposition and reconstruction methods. Section III presents the SVDE optimization algorithm and is tested for performance. Then, Section IV outlines the prediction method and the prediction steps. Next, Section V includes example validation and analysis of prediction results. Finally, section VI concludes the paper.

II. LOAD DATA DECOMPOSITION AND RECONSTRUCTION

A. ICEEMDAN DECOMPOSITION

The actual power load has a large number of nonlinear, non-stationary, and strong random fluctuation characteristics. Empirical Mode Decomposition (EMD) is a data mining method with the adaptive capability to decompose signals based on the time-scale features of the data itself, which can handle complex nonlinear load prediction problems without any predefined basis functions. The Ensemble Empirical Mode Decomposition (EEMD) is proposed to effectively suppress the mode mixing in the EMD decomposition, but the reconstruction error is large in the decomposition process [32]. The Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) decomposition effectively reduces the reconstruction error, and the decomposition is more efficient.

Based on the decomposition algorithms of EMD and CEEMDAN, the Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (ICEEMDAN) decomposes the original signal $x(t)$ into several Intrinsic Mode Functions (IMF) components and a residual component by introducing white Gaussian noise with mean

0 and unit variance in the decomposition process [33]. The ICEEMDAN decomposition method effectively solves the problems of mode aliasing in the EMD algorithm, possible residual noise in the EEMD algorithm, and pseudo-mode in the CEEMDAN algorithm. The decomposition process of the ICEEMDAN algorithm is as follows.

1) The $E_k(\cdot)$ operator in the EMD decomposition method is introduced to decompose the k_{th} mode component, whose component is denoted as $\overrightarrow{IMF_k}$, and the $M(\cdot)$ operator is used to generate the local mean value of the signal.

2) Add I groups of white Gaussian noise to the original power load sequences, construct a partial mean decomposition sequences $x^i(t) = x(t) + \varepsilon_0 E_1\left(\xi^i(t)\right)$ by the EMD, and obtain the first-order residuals as equation (1).

$$\gamma_1 = \left\langle M\left(x^i(t)\right)\right\rangle \tag{1}$$

where $\varepsilon_0 = \delta_0 std(x)/std\left(E_1\left(\xi^i(t)\right)\right)$ is used to remove residual noise in the initial stage, δ_0 is the inverse of the expected signal-to-noise ratio between the first group of white noise (ξ^i) and the analyzed signal, i denotes the times of adding noise, std denotes the standard deviation, and $\langle\cdot\rangle$ denotes averaging over the whole.

3) Calculate the first-stage ($k = 1$) mode components, as follows:

$$\overrightarrow{IMF_1}(t) = x(t) - \gamma_1(t) \tag{2}$$

4) Using the average of the local means of $\gamma_1(t) + \varepsilon_1 E_2\left(\xi^i(t)\right)$, the second-order residuals and the second-order mode components IMF are obtained as equations (3)-(4).

$$\gamma_2(t) = \left\langle M\left(\gamma_1(t) + \varepsilon_1 E_2\left(\xi^i(t)\right)\right)\right\rangle \tag{3}$$

$$\overrightarrow{IMF_2}(t) = \gamma_1(t) - \left\langle M\left(\gamma_1(t) + \varepsilon_1 E_2\left(\xi^i(t)\right)\right)\right\rangle \tag{4}$$

5) For the remaining stages $k = 3, \ldots, k$, calculate the $k_t h$ order residuals and IMF components as equations (5)-(6).

$$\gamma_k(t) = \left\langle M\left(\gamma_{k-1}(t) + \varepsilon_{k-1} E_k\left(\xi^i(t)\right)\right)\right\rangle \tag{5}$$

$$\overrightarrow{IMF_k}(t) = \gamma_{k-1}(t) - \left\langle M\left(\gamma_{k-1}(t) + \varepsilon_{k-1} E_k\left(\xi^i(t)\right)\right)\right\rangle \tag{6}$$

6) Repeat the previous step to find all residuals and components.

B. GREY RELATION ANALYSIS

The principle of Grey Relation Analysis (GRA) is to determine the degree of similarity between different influencing factors and the system by calculating the grey correlation between them and analyzing the influence degree of different influencing factors on the development of the system [34]. It is divided into four main steps:

1) Determine the analysis sequences. Let the parent sequence is $X_0 = (X_0(1), X_0(2), \cdots, X_0(n))$, the subsequence is $X_i = (X_i(1), X_i(2), \cdots, X_i(k), \cdots, X_i(n))$, where $i = k = 1, 2, \cdots, n$.

2) Dimensionless preprocessing of the subseries. The mean value of the subseries data is first found, and then that mean value is removed using each data of the subseries. The formula is as follows:

$$X_i' = X_i \Big/ \tilde{X}_i \tag{7}$$

3) Calculate the grey relational coefficient. The formula is as equations (8)-(9).

$$\gamma(X_0(k), X_i(k)) = \frac{\min_i \min_k \Delta_i(k) + \xi \max_i \max_k \Delta_i(k)}{\Delta_i(k) + \xi \max_i \max_k \Delta_i(k)} \tag{8}$$

$$\Delta_i(k) = \left|X'_o(k) - X'_i(k)\right| \tag{9}$$

4) Calculate the gray relational degree between them. The formula is as follows:

$$\gamma(X_0, X_i) = \frac{1}{n}\sum_{k=1}^{n} \gamma(X_0(k), X_i(k)) \tag{10}$$

III. SVDE ALGORITHM AND ALGORITHM PERFORMANCE

A. SECONDARY VARIATION DIFFERENTIAL EVOLUTION ALGORITHM

Differential Evolution (DE) is a random search algorithm based on population differences, which is essentially an evolutionary algorithm based on real number encoding with the idea of preservation of superiority [35]. The algorithm is inspired by the criterion of "survival of the fittest" in nature and selects the next generation by initializing the population, evaluating individual fitness value, differential variation operation, hybridization operation, and selection operation. It has the characteristics of few control parameters, strong robustness, and strong global optimization capability. The four steps of the standard differential evolution algorithm are as follows.

1) POPULATION INITIALIZATION

A random initialization in the solution space produces a parent population with population size NP and dimension D. The $j_t h$ dimensional vector of each individual ($\overrightarrow{x_i}$) is expressed as:

$$\begin{cases} x_{i,j} = x_{\min xj} + rand(0,1) \cdot \left(x_{\max xj} - x_{\min xj}\right) \\ i = 1, 2, \cdots, NP; j = 1, 2, \cdots, D \end{cases} \tag{11}$$

2) VARIATION OPERATION

Through the difference strategy, the difference vector is generated by subtracting two different parents in the parent population, and the vector is synthesized with the individuals to be mutated to obtain the variation vector [36]. Its common variation strategies are as equations (12)-(15).

$$DE/rand/1 : v_{i,G+1} = x_{r1,G} + F \cdot \left(x_{r2,G} - x_{r3,G}\right) \tag{12}$$

$$DE/best/1 : v_{i,G+1} = x_{best,G} + F \cdot \left(x_{r1,G} - x_{r2,G}\right) \tag{13}$$

$$DE/current-to-rand/1 :$$

$$v_{i,G+1} = x_{i,G} + F \cdot \left(x_{r1,G} - x_{i,G}\right) + F \cdot \left(x_{r2,G} - x_{r3,G}\right) \tag{14}$$

$DE/current-to-best/1$:

$$v_{i,G+1} = x_{i,G} + F \cdot \left(x_{best,G} - x_{i,G}\right) + F \cdot \left(x_{r1,G} - x_{r2,G}\right) \tag{15}$$

where $v_{i,G+1}$ denotes the individual after variation, $x_{i,G}$ denotes the i_th individual in the G_{th} generation; $x_{r1,G}$,$x_{r2,G}$ and $x_{r3,G}$ denote the different individuals randomly selected in the current population, $x_{best,G}$ denotes the best individual in the G_{th} generation, and F is the scaling factor, whose value range is generally [0,1]. After many tests, the $DE/current-to-best/1$ variation strategy has the shortest search time under the same optimal solution when the value of the scaling factor is 0.55.

3) CROSSOVER OPERATION

After the variation operation, the experimental individuals were obtained by the binomial crossover of the G_{th} generation individuals and the mutant individuals, as equations (16)-(17).

$$u_{i,G+1} = \left(u_{i1,G+1}, u_{i2,G+1}, \cdots, u_{iD,G+1}\right) \tag{16}$$

$$u_{ij,G+1} = \begin{cases} v_{ij,G+1} & if\ rand\ (0,\ 1) \le CR, or\ j = j_c \\ x_{ij,G} & others \end{cases} \tag{17}$$

where CR is the crossover factor in the crossover operation, whose value range is generally [0,1], and the cross effect is better when the value of 0.35 is tested. j_c denotes a random integer in the interval [1, D].

4) THE SELECTION OPERATION

Differential evolutionary algorithms usually use a greedy selection strategy to select the better fitness among the child individual $u_{i,G+1}$ and current individual $x_{i,G}$ as members of the next generation.

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & f_i\left(u_{i,G+1}\right) \le f_i\left(x_{i,G}\right) \\ x_{i,G} & others \end{cases} \tag{18}$$

$$f_i\left(\cdot\right) = \left|f\left(\cdot\right) - f\left(x_{best}\right)\right| \tag{19}$$

where $u_{i,G+1}$ denotes the child individual after the selection operation, $x_{i,G}$ is the contemporary individual, x_{best} is the optimal solution of the test function. $f\left(\cdot\right)$ is the value of the test function, and $f_i\left(\cdot\right)$ denotes the value of the fitness function. When the value of the fitness function of the offspring individual is less than the value of the parent individual, the child individual will replace the parent individual into the next generation to complete the renewal of the population.

The traditional differential evolutionary algorithm uses a single variation strategy to produce variation individuals, which makes its algorithm parameters remain fixed in the optimization process, resulting in the algorithm search easily falling into the local optimum and premature phenomenon, which affects the solution accuracy and convergence speed of the algorithm [37]. Therefore, the initial period of the differential evolution algorithm should have a large optimization range to ensure the diversity of the population. Gaussian variation, as a variation operation method to improve the genetic algorithm, can produce a large variation step, which makes the algorithm a better global search ability, so it is suitable for exploration in the early period of the algorithm. In the middle and later periods of the algorithm, the local optimum should be jumped out as soon as possible, and the local search should be conducted for the region near the current optimal individual [38]. The $DE/current-to-best/1$ variation strategy can search for better individuals faster while ensuring a certain population diversity. Compared with $DE/rand/1$ and $DE/best/1$, it has a faster speed of convergence and the capability to escape from the local optimum, so it is suitable for the later development stage of the algorithm.

Aiming at the different characteristics of the above two variation strategies, a differential evolution algorithm (SVDE) with a secondary variation strategy is proposed, combining Gaussian variation and $DE/current-to-best/1$ variation, adding a dynamic weight factor SA to balance the weights of the two strategies, guiding the algorithm to select the appropriate variation strategy at different evolutionary times, and improving the optimization ability of the algorithm. The specific strategies are as equations (20)-(21).

$$M_i = \begin{cases} v_{i,G+1} = N\left(\dfrac{x_{best,G}+x_{i,G}}{2}, \left|x_{best,G} - x_{i,G}\right|\right) \\ \qquad rand\ (0,\ 1) \le SA_i \\ v_{i,G+1} = x_{i,G} + F \cdot \left(x_{best,G} - x_{i,G}\right) + F \cdot \left(x_{r1,G} - x_{r2,G}\right) \\ \qquad others \end{cases} \tag{20}$$

$$SA_i = 1 - e^{-\left|f_i(x_{i,G}) - f_i(x_{best,G})\right|} \tag{21}$$

where M_i denotes the variation strategy performed for the individual $x_{i,G}$, and SA_i is the dynamic weight factor corresponding to the individual $x_{i,G}$. The value of the current individual $x_{i,G}$ is closer to the value of the optimal individual $x_{best,G}$ in the G_{th} generation, SA_i has a smaller value, and the probability of the algorithm using the $DE/current-to-best/1$ variation strategy is larger, and vice versa the probability of the algorithm using the Gaussian variation strategy is larger. In the evolutionary process, the value of SA_i is adaptively adjusted with the evolutionary process, which is able to explore the unknown search region, but also able to accelerate the mining of the neighborhood of the known excellent individual, which in turn improves the algorithm's ability to find the optimal.

In order to further improve the problem of premature convergence of the difference algorithm, it should be ensured that the global and local search capabilities of the

TABLE 1. Each test function and related parameters.

Test function name	Function	Range of values	Optimal solution
Sphere	$f_1 = \sum_{i=1}^{n} {x_i}^2$	[-100,100]	0
Schwefel 2.22	$f_2 = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
Rastrigin	$f_3 = \sum_{i=1}^{n} \left[{x_i}^2 - 10\cos(2\pi x_i) + 10 \right]$	[-5.12,5.12]	0
Ackley	$f_4 = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} {x_i}^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right) + 20 + e$	[-32,32]	0
Hybrid function	$f_5 = \left(\sin^2\sqrt{{x_1}^2 + {x_2}^2} - 0.5\right) \Big/ \left[1 + 0.001({x_1}^2 + {x_2}^2)\right]^2 + 0.5$	[-100,100]	0

algorithm collaborate with each other. Therefore, a stagnation perturbation strategy is added and a parametric population update ratio r^G is introduced. According to its value and the size of the set threshold, adjust the value of the dynamic weight factor SA_i, so as to change the evolutionary direction of the next iteration of individuals, and ensure that the algorithm jumps out of this local extreme value region in a timely and limited manner. Their population renewal ratios r^G and SA_i are related as follows.

$$SA_{i,G+1} = \begin{cases} 0.5 & r^G \le \delta \\ SA_{i,G} & others \end{cases} \tag{22}$$

where if the value of δ is too large, it will affect the evolution of the population towards the favorable trend in the early stage of evolution; while if the value of δ is too small, it will miss the better time of jumping out of the local extreme value. By trying continuously from 0 to 1 values, good optimization performance for different test functions is shown when the value of δ is 0.35.

The specific process of the SVDE algorithm is shown in Fig. 1.

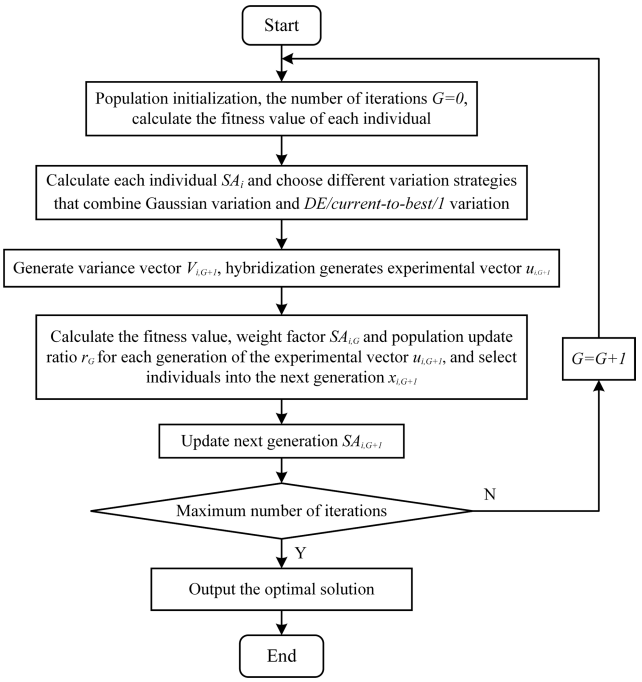

FIGURE 1. Flowchart of SVDE optimization algorithm.

B. SVDE FUNCTION PERFORMANCE TEST

In order to evaluate the performance of the overall optimization of the SVDE algorithm, the Particle Swarm Optimization (PSO), Differential Evolution algorithm (DE), Genetic Algorithm (GA), Chaos Sparrow Search Optimization Algorithm (CSSOA) [39], Whale Optimization Algorithm with Hybrid Reverse Learning (MWOA) [40] and SVDE algorithm were subjected to performance comparison experiments on five benchmark test functions, and each function and the associated parameters are shown in Table 1. For ensuring fairness, the population size of the six algorithms NP is uniformly set to 100, the dimensionality is 20, the maximum number of function evaluations be 20,000, each algorithm is run 30 times independently [41], and the specific parameters of each algorithm are established as in Table 2, and the optimal mean solution (the first row of each function) and standard deviation (the second row of each function) are obtained as in Table 3.

TABLE 2. Parameter settings of each algorithm.

Algorithm	Parameter settings
PSO	c1=c2=1.5, ω=0.8
GA	Pc=0.55, Pm=0.35
DE	F=0.55, CR=0.35
CSSOA	$R_2 \in [0,1]$, $ST \in [0.5,1]$
MWOA	$r_1 \in [0,1]$, $r_2 \in [0,1]$, b=1
SVDE	F=0.55, CR=0.35

In the same dimension, compared with the other three algorithms, the SVDE algorithm has an absolute advantage in convergence accuracy and the ability to find the best (Table 3). Except for the difference between the SVDE algorithm and other algorithms on function 5, the optimum mean and standard deviation of the optimal solution of SVDE on the rest of the functions reach the optimal level, followed

TABLE 3. Optimal mean solution and standard deviation of functions.

Function	evaluations	PSO	GA	DE	CSSOA	MWOA	SVDE
f_1	optimal solution	3.9721e-04	1.7296e+03	3.3050e-06	2.4579e-14	6.6247e-16	**1.5517e-18**
	standard deviation	1.2563e-02	1.6153e+01	1.4998e-06	1.1498e-15	4.5787e-16	**3.5114e-18**
f_2	optimal solution	6.6214e-02	7.5064e+03	1.5153e-08	3.8965e-12	5.2794e-10	**1.9343e-14**
	standard deviation	1.1158e-01	2.2284e+03	4.2429e-09	4.0571e-12	8.4163e-11	**4.4661e-14**
f_3	optimal solution	2.2936e-01	-1.5235e+00	1.6688e-01	1.7425e-10	5.2147e-07	**2.8554e-11**
	standard deviation	1.4094e-00	2.8878e+00	1.3281e-01	3.4789e-11	4.4734e-08	**1.8334e-12**
f_4	optimal solution	3.6751e-02	2.11367e-01	1.4658e-11	3.7412e-17	5.4784e-14	**1.4635e-19**
	standard deviation	5.5009e-01	4.8260e-02	2.9005e-12	1.7846e-17	7.8941e-14	**2.4421e-20**
f_5	optimal solution	2.2769e-13	1.1494e-12	2.5980e-14	7.0984e-12	6.2478e-14	**1.1243e-14**
	standard deviation	5.4083e-13	8.3746e-12	2.1649e-13	3.4721e-13	8.4789e-14	**2.3812e-14**

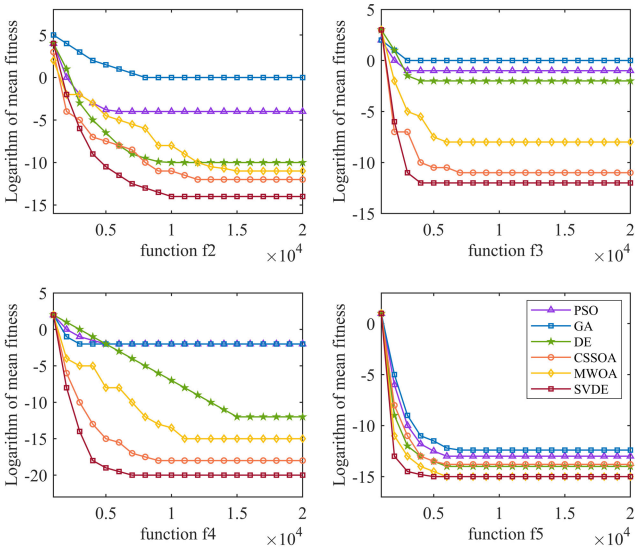

FIGURE 2. Logarithmic curve of fitness value of f_2-f_5 function.

by DE and PSO algorithms, and the standard GA algorithm has the worst performance in finding the best.

Within the maximum number of evaluations (20,000 times), 20 points are taken on average as the observation points for function evaluation. In order to show the convergence speed of the SVDE algorithm more intuitively, the convergence curves of f_2-f_5 four more complex test functions are shown in Fig. 2, with the horizontal coordinate being the number of function evaluations and the vertical coordinate being the logarithm of the mean fitness of each algorithm run independently for 30 times at each fixed observation point.

As shown in Fig. 2, compared with three traditional optimization algorithms and two mainstream improvement algorithms, the SVDE algorithm has strong convergence ability and optimization seeking effect. Since the dynamic weight factor *SA* in the SVDE algorithm can change the proportion of Gaussian variation and $DE/current-to-best/1$ variation in real-time according to the population update ratio, it makes the algorithm jump out of local optimum based on having a large range of seeking advantages more quickly to achieve better convergence speed and convergence effect.

IV. LOAD FORECASTING METHOD BASED ON ERROR CORRECTION MODEL AND ICEEMDAN-GRA-CVDE-BIGRU

A. BIGRU

A Recurrent Neural Network (RNN) is suitable for analyzing and processing time series data because the recurrent unit structure is introduced in an RNN network, which can transfer the information between layers in both directions, and then form a memory of the information, so it has some memory ability. However, the RNN neural network has some shortcomings, when the sequence is long, its learning ability and the ability of memory will be reduced, and there are problems of gradient disappearance and gradient explosion [42].

The LSTM neural network is a special kind of RNN, which was proposed to solve the problem of RNN gradient disappearance [43]. It has been widely used in the territory of predicting time series data, and many variants have evolved in recent years according to different needs. GRU is a variation of LSTM that uses a gated recurrent neural network topology that has fewer training parameters and faster training speed compared to LSTM and maintains the prediction effect of LSTM [44]. The inner unit of the GRU is very familiar to that of the LSTM, with the difference that the GRU integrates the input and forgetting gates in the LSTM into a single update gate. As a result, there are only two gate structures in GRU, which are the update gate and the reset gate. The update gate is used to indicate the degree to which the feature information from before the moment is retained in the current moment, and the reset gate is used to control the degree to which the message from the current moment is combined with that from the previous moment. The GRU neural network with the structure shown in Fig. 3.

The calculation formulas of the GRU neural network are as equations (23)-(26).

$$r_t = \sigma\left(W_r\left[h_{t-1}, x_t\right]\right) \tag{23}$$

$$z_t = \sigma\left(W_z\left[h_{t-1}, x_t\right]\right) \tag{24}$$

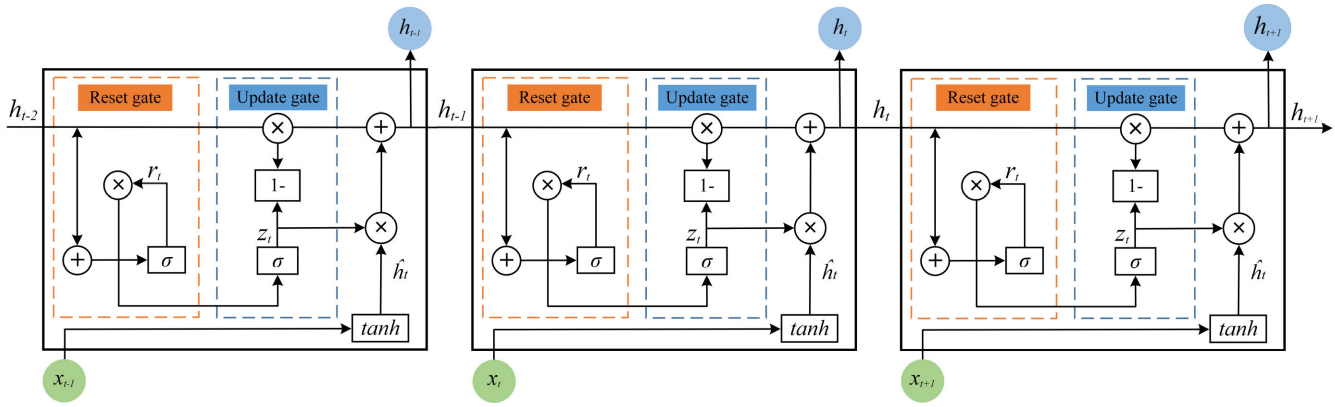

FIGURE 3. Structure and connection of gated recurrent units.

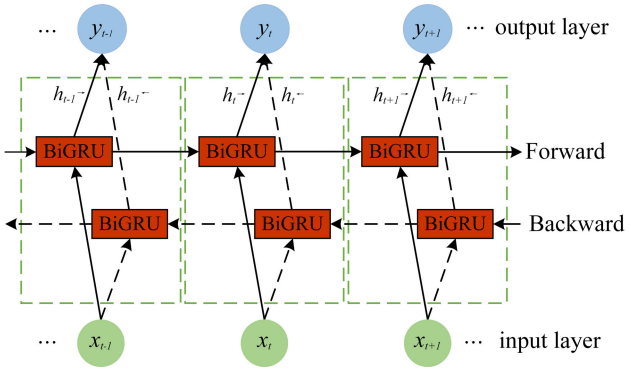

FIGURE 4. Structure of BiGRU neural network model.

$$\hat{h}_t = \tanh\left(W_{\hat{h}}\left[r_t * h_{t-1}, x_t\right]\right) \tag{25}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \tag{26}$$

where x_t is the input at moment t, z_t and r_t indicate the output of the update gate and reset gate at the moment of t; h_t and h_{t-1} are the state information of the current moment and the previous moment, respectively; $\hat{h}_t$ indicates the candidate hidden state, $W_{\hat{h}}$, W_r and W_z are the weight matrices of the candidate hidden state, update gate, and reset gate, respectively; σ indicates the sigmoid activation function, [] indicates the connection of two vectors, and $*$ indicates the product between matrices.

The information is propagated in a unidirectional neural network in a sequence from front to back, and the power load is associated with both historical periods as well as future periods, and a single GRU neural network cannot further extract the deeper features of the data [45]. Aiming at the shortcomings of one-way information transfer, the model prediction performance is improved by constructing a Bi-directional Gated Recurrent Unit (BiGRU). The established BiGRU model is shown in Fig. 4.

BiGRU has two independent hidden layers with the same internal structure and opposite directions, relying on both directions to provide data information to the network simultaneously. The GRU of the first layer calculates the forward propagation information of the data in the temporal order, while the GRU of the second layer reads the time series in the reverse direction and calculates the backward propagation information of the data in the reverse chronological order. These two layers of GRUs do not interfere with each other in computing their hidden states, and the final BiGRU network output information is determined jointly by the two. The output equation is as follows:

$$\vec{h}_t = GRU(x_t, \vec{h}_{t-1}) \tag{27}$$

$$\overleftarrow{h}_t = GRU(x_t, \overleftarrow{h}_{t-1}) \tag{28}$$

$$H_t = \omega_t \vec{h}_t + v_t \overleftarrow{h}_t + b_t \tag{29}$$

where $GRU(\cdot)$ denotes the nonlinear change obtained after learning by a single GRU network, ω_t and v_t denote the weights corresponding to the forward GRU and the backward GRU at time t, and b_t is the bias term. $\vec{h}_t$ is the output of the forward GRU and $\overleftarrow{h}_t$ is the output of the backward GRU, and through the forward transmission hiding layer and the backward transmission hiding layer to obtain the hidden state H_t at the moment.

B. ERROR CORRECTION MODEL

Due to the influence of objective factors such as temperature and holidays, the load can change significantly during the periods such as sudden temperature drops, sudden temperature rises, and holiday weekends. On the other hand, the daily higher load periods and lower load periods in the short-term load have periodicity, and BiGRU neural network training cannot quickly make accurate predictions of these anomalies when the load values are directly used as outputs. According to the six-dimensional data in the example dataset, the correlation between its temperature, humidity, precipitation, holiday weekend type, high or low load periods and original power load are calculated using the grey correlation analysis (GRA), and the specific correlation is shown in Table 4.

As can be seen in Table 4, humidity and precipitation variables have a low correlation with original power load of less than 0.6. Temperature, holiday weekend type, high or low

TABLE 4. Correlation of six-dimensional data variables with original power load.

Variants	Temperature	Humidity	Precipitation	Holiday or weekend	Non-holiday or workday	High load period	Low load period
Correlation	0.879	0.572	0.418	0.948	0.618	0.833	0.754

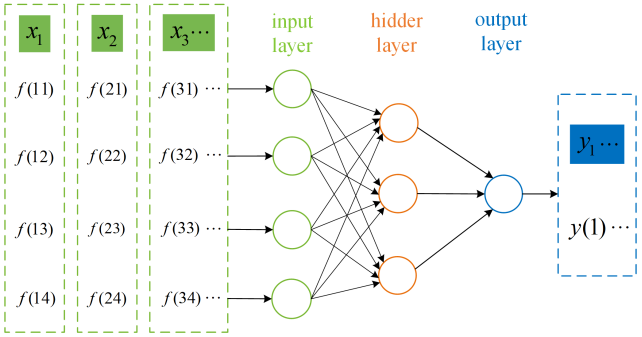

FIGURE 5. Neural network structure of error correction model.

load periods have higher correlation with raw electricity load and they can be used as inputs to the error correction model.

The ICEEMDAN decomposed feature components ($IMF_1, \ldots, IMF_k$), temperature, holiday weekend type, and daily high or low load types are selected as inputs $f(ii)$ of the BiGRU neural network. The prediction residual sequences $y(i)$ obtained by subtracting the original load sequences from the initial prediction sequences of the ICEEMDAN-GRA-SVDE-BiGRU model is used as outputs, and the SVDE-BiGRU neural network is used to train its hyperparameters so as to obtain load prediction residual sequences in the test set. The neural network structure of the error correction model is shown in Fig. 5.

C. ICEEMDAN-GRA-CVDE-BIGRU-ECM PREDICTION PROCESS

In order to reduce load forecasting errors, a combined load forecasting model based on ICEEMDAN-GRA-CVDE-BiGRU and Error Correction Model (ECM) is proposed, and its specific process is shown in Fig. 6. The specific steps are as follows.

Step 1. The ICEEMDAN method is used to decompose the power load history data into two categories: multiple eigenmode components (IMF) and residual components (Res).

Step 2. The grey correlation analysis (GRA) is used to analyze the correlation of the decomposed components, and the sequences of components with similar correlation are fused to form Intrinsic Mode Functions ($IMF_1', \ldots, IMF_m'$).

Step 3. The multiple processed mode components are input to the BiGRU neural network for training, and the four hyperparameters of the number of hidden layer neurons (L_1, L_2), learning rate lr, and $batch\_size$ of BiGRU are optimized by the SVDE algorithm, to obtain multiple initial prediction sequences.

Step 4. The error correction model is built by using five variables in the training set: prediction residual series

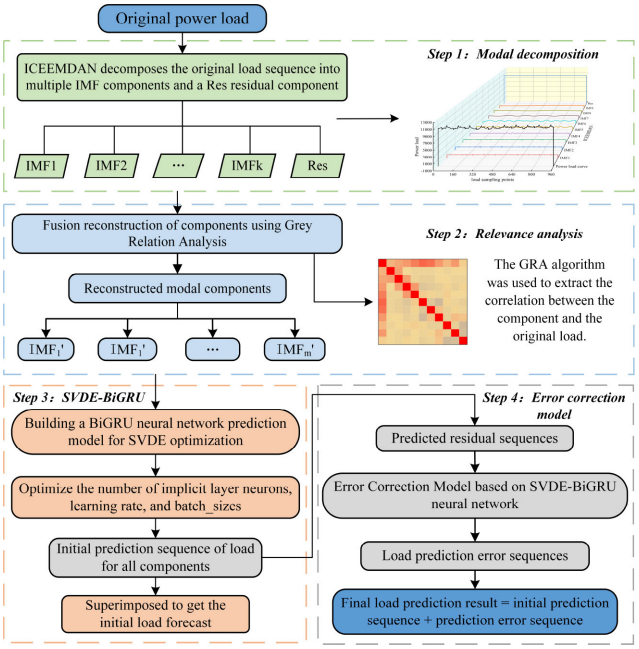

FIGURE 6. ICEEMDAN-GRA-SVDE-BiGRU-ECM prediction method flow.

(the load series minus the initial prediction series), mode components ($IMF_1', \ldots, IMF_m'$), temperature, holiday weekend type, and daily high or low load type, which leads to the prediction error series. Finally, the initial prediction series and the prediction error series are superimposed to obtain the final load prediction results after error correction.

V. ANALYSIS AND VERIFICATION OF ALGORITHMS

A. INTRODUCTION OF ALGORITHMS AND EVALUATION CRITERIA

To verify the accuracy and applicability of this load forecasting method, the experimental data were selected from a loaded dataset of an area in Ningxia City, China, which included load, temperature, humidity, precipitation, holiday weekend type, high or low load periods in six dimensions. The data time was selected from January 1, 2020, to January 20, 2020, and from July 10 to July 30 of the same year, which respectively represent the experimental conditions under high and low temperature environments. The data sampling interval is 15 minutes, and each part of the data set has 1920 sampling points, for a total of 3840 sampling points. To improve the correlation between different factors, four-dimensional data of load sequences, temperature, holiday weekend type, high or low load periods with correlation are selected as inputs to the BiGRU neural network, and these data are normalized to [-1,1], and the test set is back-normalized according to the training set. According to

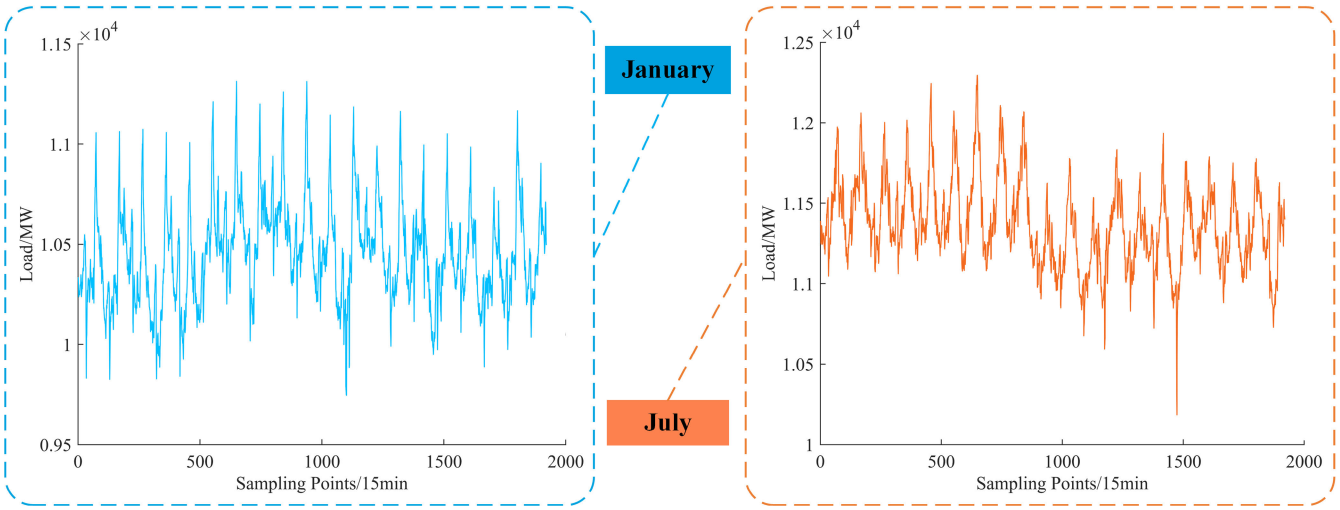

FIGURE 7. Original power load.

the data division criteria of the training and test sets, the first 1728 sampling points data of each part of the data set are used as the training set, which contains the data for the validation part of the model, and the last 192 sampling points data are used as the test set. The original power load is shown in Fig 7.

Due to the uncertainty and volatility of electric load, root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) are proposed to verify the accuracy of model prediction in order to evaluate the prediction performance of each model. In addition, the standard deviation (S) and correlation coefficient (R) are used as auxiliary indicators of prediction accuracy. The specific formulas are as equations (30)-(34).

$$RMSE = \left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2\right)^{\frac{1}{2}} \tag{30}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{31}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{32}$$

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{33}$$

$$R = \frac{\sum_{i=1}^{N}\left(\hat{y}_i - \bar{\hat{y}}\right)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}\left(\hat{y}_i - \bar{\hat{y}}\right)^2 \cdot \sum_{i=1}^{N}(y_i - \bar{y})^2}} \tag{34}$$

where y_i represents the actual value of the power load and $\hat{y}_i$ represents the forecast value of the power load. $\bar{y}$ is the average of the actual values and $\bar{\hat{y}}$ is the average of the predicted values, and $\bar{y}=\sum_{i=1}^{N} y_i, \bar{\hat{y}}=\sum_{i=1}^{N} \hat{y}_i/N$.

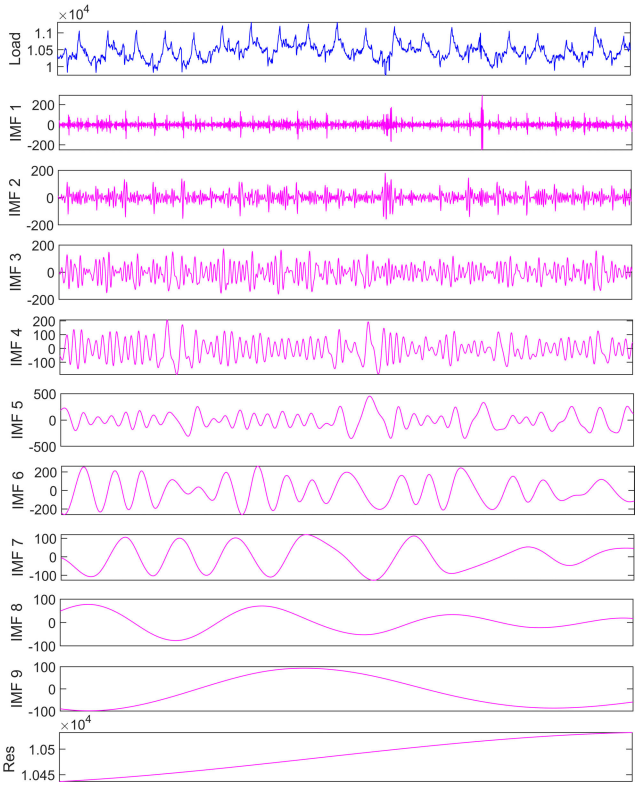

FIGURE 8. Decomposition results of ICEEMDAN power load.

B. ICEEMDAN DECOMPOSITION AND RECONSTRUCTION

ICEEMDAN decomposition can calculate the optimal number of modal components adaptively according to the complexity and local characteristics of the data sequences, and ensure the accuracy of the reconstruction operation on the components. As shown in Fig. 8, the load sequences of this part in January are decomposed, the noise standard deviation and signal average are set to 0.2 and 500 times, and the maximum number of iterations is 1000, and nine sets of mode

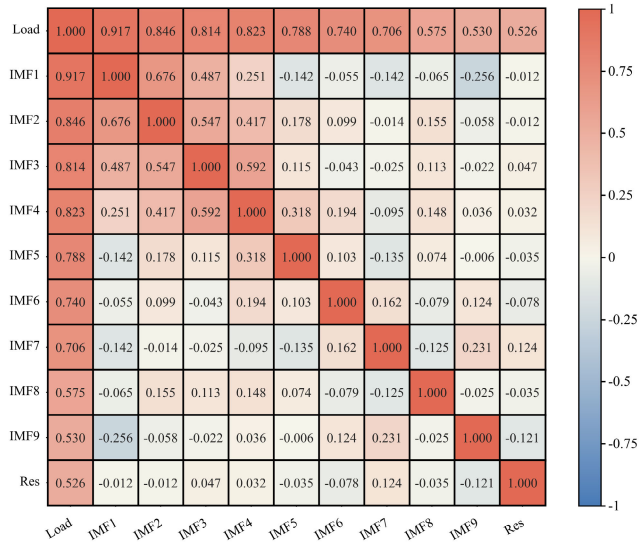

FIGURE 9. Correlation heat map of sequences and components.

TABLE 5. Decomposition reconstruction of each component seque nce.

IMF1'	IMF2'	IMF3'	IMF4'	IMF5'	IMF6'	IMF7'	IMF8'
IMF1	IMF2	IMF3	IMF4	IMF5	IMF6	IMF7	IMF8+IMF9+Res

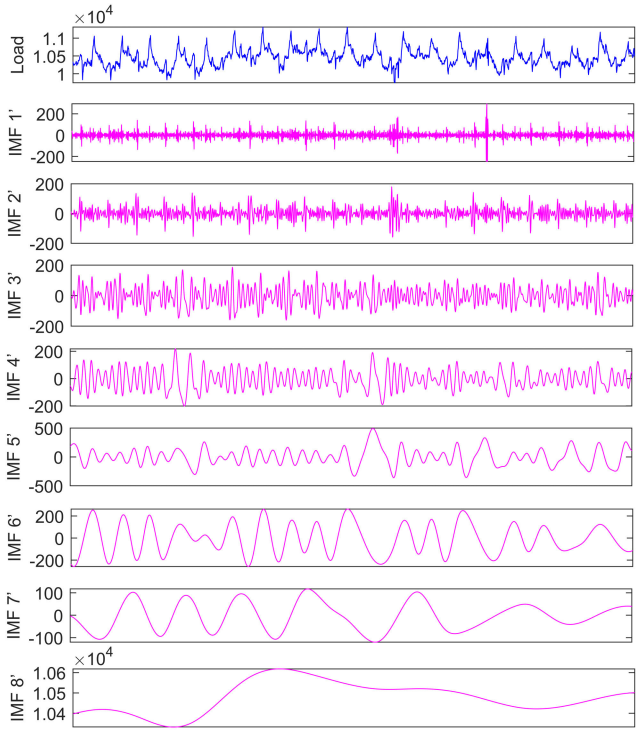

FIGURE 10. Decomposition results of ICEEMDAN-GRA power load.

components (IMF) and residuals (Res) are obtained from high to low frequency.

From the decomposition results, the frequency volatility of each high-frequency component of ICEEMDAN is small, and the fluctuation trends of IMF8 and IMF9 components of ICEEMDAN decomposition in the low-frequency component are close to each other, and the phenomenon of excessive decomposition appears. On the whole, the modal components decomposed by ICEEMDAN do not show obvious modal mixing and pseudo-mode phenomena.

In order to decrease the dimensionality of the input data of the neural network while reducing the number of feature components, the sequence components with higher relevance to the original load sequences are selected for training. The grey correlation of each component of the ICEEMDAN decomposition and the original load sequences were calculated separately using the GRA method, and the correlation heat map is shown in Fig. 9.

As shown in Fig. 9, the grey correlation values of IMF_8, IMF_9, and Res components with the original load sequences of are less than 0.6; and the correlation values are similar, so the combinations are reconstructed into a new sequence component. The combination of each sequence component is shown in Table 5, and the sequences after the component reconstruction are shown in Fig. 10.

C. ICEEMDAN-GRA-CVDE-BIGRU-ECM SHORT-TERM LOAD PREDICTION EXPERIMENT

In order to verify the accuracy and feasibility of the proposed combinatorial model and optimization algorithm in this paper, two combinatorial comparison experiments are used for validation.

The first combination uses internal ablation experiments to verify the effectiveness of the combined prediction method proposed in this paper. The four models, BiGRU, SVDE-BiGRU, ICEEMDAN-SVDE-BiGRU, and ICEEMDAN-GRA-SVDE-BiGRU, are selected as the comparison models, and the present model has experimented with the same parameter settings. The number of input layer neurons of BiGRU in the case analysis is 24, the number of hidden layers is 2, the number of output layer neurons is 1, and the optimizer is Adam. In the ICEEMDAN-GRA-SVDE-BiGRU-ECM model, the four variables of modal component, temperature, holiday weekend type, high or low load type of the uncorrected model decomposition are used as the inputs, and the predicted residual values as the output for the error correction model, in which the number of input neurons for BiGRU is 64, the number of implied layers is 4, the number of training epochs is 10, and the optimizer is AdaGrad. The load training results of the training set (including validation part) of the proposed model are shown in Fig. 11. The final error evaluations and time cost of the five models are shown in Table 6 and Table 7, the prediction results of the five models are shown in Fig. 12 and Fig. 13, and the violin plot of the error results is shown in Fig. 14.

As shown in Table 6-7 and Fig. 12-13, the single BiGRU neural network has the worst prediction effect, and the MAE of SVDE-BiGRU is 112.718 MW lower than that of BiGRU under the same BiGRU parameters in the July experiment. The prediction time cost of the SVDE-BiGRU

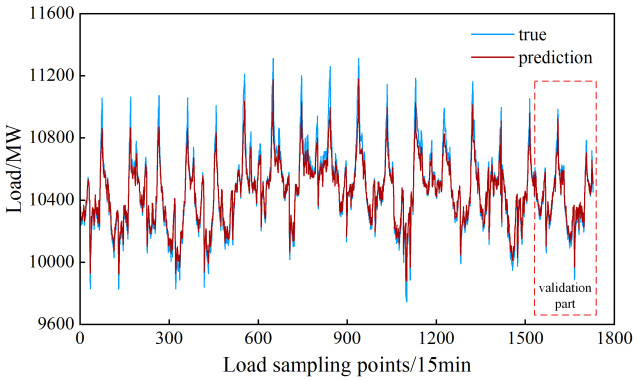

FIGURE 11. Load training results for the training set in January.

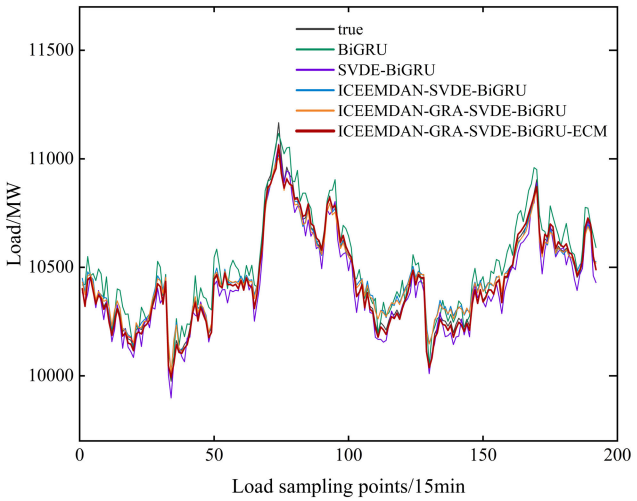

FIGURE 12. Prediction results of the first combination.

TABLE 6. Internal ablation experiment error evaluation results in January.

Models	RMSE/MW	MAE/MW	MAPE/%	time cost/s
BiGRU	325.174	263.338	2.447	7.35
SVDE-BiGRU	278.248	192.709	1.719	11.48
ICEEMDAN-SVDE-BiGRU	201.425	149.425	1.387	18.47
ICEEMDAN-GRA-SVDE-BiGRU	126.488	97.751	0.918	12.89
ICEEMDAN-GRA-SVDE-BiGRU-ECM	91.752	64.487	0.606	14.76

TABLE 7. Internal ablation experiment error evaluation results in July.

Models	RMSE/MW	MAE/MW	MAPE/%	time cost/s
BiGRU	451.274	323.353	3.062	9.68
SVDE-BiGRU	264.895	210.635	1.892	13.04
ICEEMDAN-SVDE-BiGRU	175.442	140.192	1.261	19.88
ICEEMDAN-GRA-SVDE-BiGRU	123.347	88.457	0.879	14.29
ICEEMDAN-GRA-SVDE-BiGRU-ECM	87.689	56.642	0.504	15.38

model is 15.38 seconds, which is 16.49 seconds less than that of the BiGRU model, which illustrates the significant improvement effect of the proposed SVDE algorithm on the model performance. After ICEEMDAN decomposition and SVDE-optimized hyperparameters, the MAPE of the two parts of the experimental data decreased by 0.482% on average, indicating the feasibility of signal

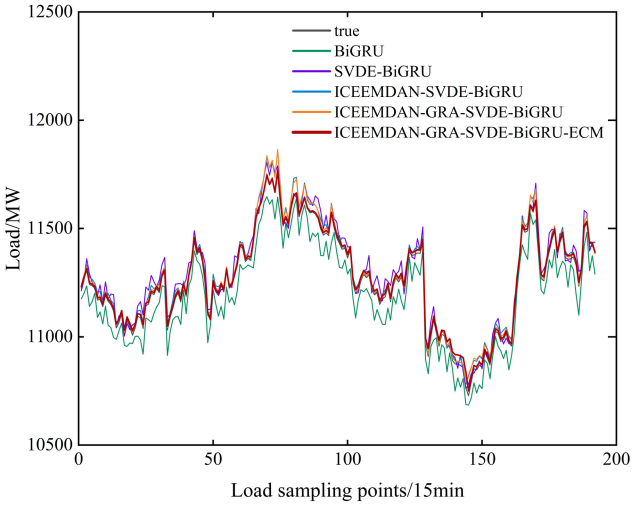

FIGURE 13. Prediction errors of the first combination.

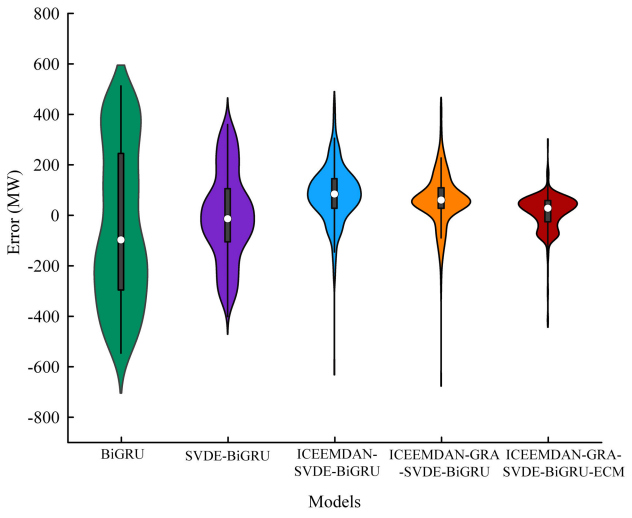

FIGURE 14. Prediction errors of the first combination.

decomposition. The ICEEMDAN-GRA-SVDE-BiGRU-ECM model has the smallest prediction error, with MAE and MAPE averaging 60.565 MW and 0.555%, respectively. The RMSE is 35.197 MW lower than that of the ICEEMDAN-GRA-SVDE-BiGRU model. As shown in Fig. 14, the proposed model in this paper has the smallest error distribution range and a more concentrated distribution, which shows that its model has higher prediction accuracy and stability.

In order to further verify the accuracy of the SVDE optimization algorithm proposed in this paper for the hyperparameters of the BiGRU neural network to find the optimal capability. To achieve the comparative analysis of various optimization algorithms, the population size and the maximum number of iterations of PSO, GA, DE, and SVDE algorithms are set as 20 and 500, respectively, and the hyperparameter hidden layer neuron number L_1, L_2, learning rate lr, and *batch_size* of BiGRU are in the optimization seeking ranges of [1, 100], [1, 100], [0.001,

TABLE 8. Hyperparameter search results.

Optimization algorithm	Hyperparameters				Optimization time
	L_1	L_2	lr	$batch\_size$	
PSO	92	42	0.0086	128	28.4s
GA	53	8	0.0124	64	36.2s
DE	68	24	0.0104	88	21.2s
CSSOA	90	43	0.0547	108	15.6s
MWOA	75	36	0.0189	96	20.9s
SVDE	88	48	0.0078	156	11.6s

TABLE 9. Evaluation results of the second combination error.

Models	RMSE/MW	MAE/MW	MAPE/%
ICEEMDAN-GRA-PSO-BiGRU-ECM	129.148	89.557	0.798
ICEEMDAN-GRA-GA-BiGRU-ECM	172.458	124.532	1.076
ICEEMDAN-GRA-DE-BiGRU-ECM	127.124	90.214	0.804
ICEEMDAN-GRA-CSSOA-BiGRU-ECM	104.579	75.849	0.712
ICEEMDAN-GRA-MWOA-BiGRU-ECM	120.784	84.178	0.796
ICEEMDAN-GRA-SVDE-BiGRU-ECM	87.689	56.642	0.504

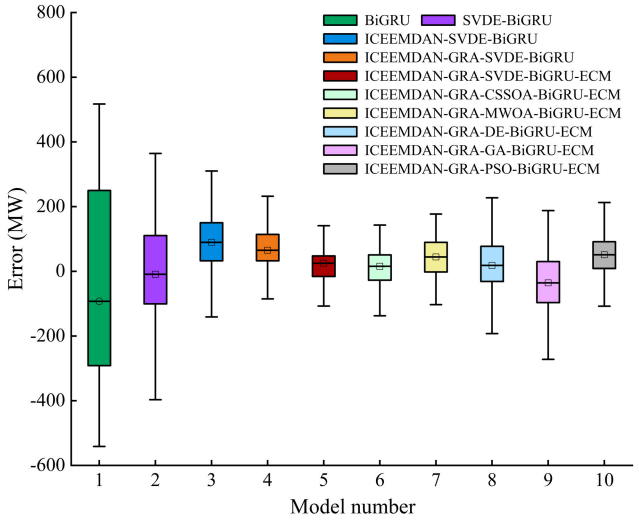

FIGURE 15. Prediction results of the second combination.

0.02], [1, 300]; the specific parameters of each algorithm are consistent with Table 2 above. According to the above parameters, six models of ICEEMDAN-GRA-PSO-BiGRU-ECM, ICEEMDAN-GRA-GA-BiGRU-ECM, ICEEMDAN-GRA-DE-BiGRU-ECM, ICEEMDAN-GRA-CSSOA-BiGRU-ECM, ICEEMDAN-GRA-MWOA-BiGRU-ECM, and ICEEMDAN-GRA-SVDE-BiGRU-ECM are established for comparative experiments.

The six hyperparameters obtained from the optimization of each model are shown in Table 8, the prediction error evaluation of the four models in the July experiment is shown in Table 9, and the overall prediction error accuracy of all models in the first group of comparison experiments is shown in Fig. 15.

As shown in Table 8, the model hyperparameters of the SVDE seeking BiGRU are L_1 is 88, L_2 is 48,

TABLE 10. Second combination of experimental error evaluation results in January.

Models	RMSE/MW	MAE/MW	MAPE/%
EEMD-GRU-MLR	184.566	149.232	1.412
CNN-BiGRU-NN	187.414	145.248	1.374
VMD-CNN-BiGRU	152.148	105.489	0.992
CNN-BiLSTM-Attention	204.564	174.178	1.652
MIC-LightGBM-XGBoost	249.154	217.574	2.056
ICEEMDAN-GRA-SVDE-BiGRU-ECM	91.752	64.487	0.606

TABLE 11. Second combination of experimental error evaluation results in July.

Models	RMSE/MW	MAE/MW	MAPE/%
EEMD-GRU-MLR	163.248	135.457	1.225
CNN-BiGRU-NN	167.463	139.185	1.253
VMD-CNN-BiGRU	141.259	98.479	0.911
CNN-BiLSTM-Attention	196.586	162.748	1.456
MIC-LightGBM-XGBoost	235.658	198.542	21.792
ICEEMDAN-GRA-SVDE-BiGRU-ECM	87.689	56.642	0.504

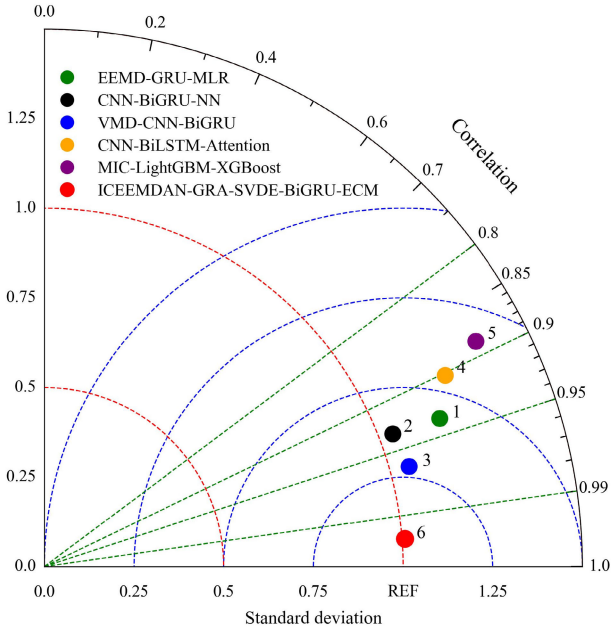

FIGURE 16. Prediction error accuracy comparison of the second combined experiment.

lr is 0.0078, $batch\_size$ is 156, and its algorithm has the shortest optimization time. As shown in Table 9 and Fig. 15, the ICEEMDAN-GRA-SVDE-BiGRU-ECM has the lowest RMSE, MAE, and MAPE values and the smallest error range. The MAE is 32.915 MW lower than the PSO optimization algorithm, 67.892 MW lower than the GA optimization algorithm, 33.572 MW lower than the DE optimization algorithm, 19.207 MW lower than the CSSOA optimization algorithm and 27.536 MW lower than the MWOA optimization algorithm. In summary, the model prediction effect is better after hyperparameter optimization

TABLE 12. Specific parameters of the comparison models.

Models	Parameter	Parameter value
EEMD-GRU-MLR	IMF number	9
	input layer	48
	hidden layer	2
	L_1	50
	L_2	32
	output layer	1
CNN-BiGRU-NN	convolutional layer	2
	K_1	64
	K_2	32
	input layer	8
	hidden layer	2
	L_1	16
	L_2	32
	output layer	34
VMD-CNN-BiGRU	IMF number	8
	convolutional layer	1
	K_1	32
	input layers	32
	hidden layer	2
	L_1	64
	L_2	128
	output layer	64
CNN-BiLSTM-Attention	convolutional layer	1
	K_1	32
	input layer	8
	hidden layer	2
	L_1	32
	L_2	64
	output layer	16
MIC-LightGBM-XGBoost	max_depth_1	12
	Learing_rate_1	0.1
	n_estimators_1	310
	num_leaves	40
	max_depth_2	10
	Learing_rate_2	0.3
	n_estimators_2	210

by the SVDE algorithm, and the SVDE algorithm has the strongest optimization-seeking ability.

The second combined experiment uses several current mainstream short-term load prediction methods as comparison models for the experiments. Five models, EEMD-GRU-MLR [46], CNN-BiGRU-NN [47], VMD-CNN-BiGRU [48], CNN-BiLSTM-Attention [49], and MIC-LightGBM-XGBoost [50], are selected as comparison models, and the model proposed in this paper have experimented in the same dataset. The specific parameters of the comparison models are shown in Table 12. The experimental error evaluations of each model are shown in Table 10 and Table 11. Taylor diagram is chosen to evaluate the prediction accuracy of several models, where the correlation coefficient (R), standard deviation (S), and root mean square error (RMSE) can demonstrate the trend and error distribution of the predicted and true values of the load, and it is shown in Fig. 16. The linear regression scatter graph for the prediction results of the model proposed in this paper is shown in Fig. 17.

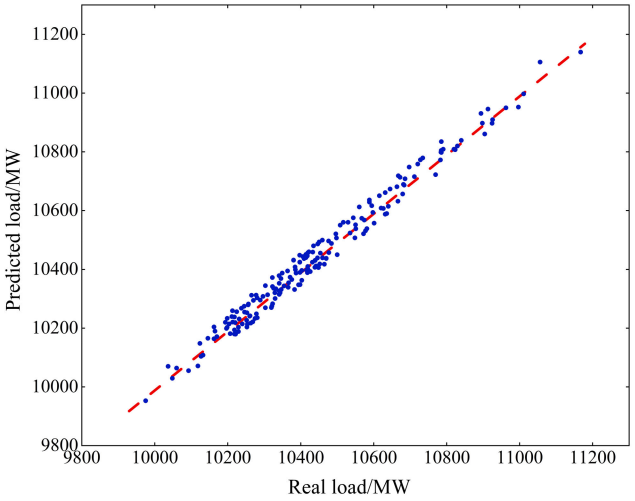

FIGURE 17. Linear regression scatter plot of prediction results.

As shown in Table 10-11 and Fig. 16, the MIC-LightGBM-XGBoost model has the largest error in the second combined experiment, the CNN-BiLSTM-Attention model has a large error, and the RMSE and S of the three models, EEMD-GRU-MLR, CNN-BiGRU-NN, and VMD-CNN-BiGRU, are at the same level. The ICEEMDAN-GRA-SVDE-BiGRU-ECM model proposed in this paper has the smallest standard deviation S as well as RMSE, and its correlation coefficient R is also closest to 1. It shows higher consistency with the trend of the true value of the load, and it can be seen that the ICEEMDAN-GRA-SVDE-BiGRU-ECM model has the highest prediction accuracy. As shown in Fig. 17, the prediction result of the model tends to be linear compared with the true value, and the combination of Fig. 12 to Fig. 17 shows that the model has good stability and robustness, which further verifies the validity and superiority of the ICEEMDAN-GRA-SVDE-BiGRU-EC model proposed in this paper.

VI. CONCLUSION

Aiming at the volatility and periodicity of power load, load forecasting accuracy is further improved. This paper proposed a short-term electric load forecasting method based on the combination of ICEEMDAN-GRA-SVDE-BiGRU and the Error Correction Model.

The ICEEMDAN decomposition method is used to decompose the original power load into modal components of different frequencies, extract the effective fluctuation information inside the sequences, and reduce the nonlinearity of the original power load. GRA is used to calculate the correlation of the decomposed components, and the feature components can reduce the dimensionality of the input data of the neural network after reconstructing them according to the correlation, and reduce the non-essential neural network training. Finally, the initial load prediction values are obtained by optimizing the hyperparameters of

BiGRU for neural network training by the SVDE algorithm with high merit-seeking capability. At the same time, the error correction model (ECM) is combined to reduce the influence of temperature and other related factors, and the residual values of load prediction obtained from training are superimposed with the initial load prediction values to obtain the final prediction values, which further improves the prediction effect and prediction accuracy.

The example validation under different conditions shows that the ICEEMDAN-GRA-SVDE-BiGRU-ECM model reduces MAE by 232.78 MW (79.02%) and MAPE by 2.21% (79.85%) on average relative to the common BiGRU model. Compared with several mainstream short-term load forecasting models, this model has a maximum MAE reduction of 147.49 MW (70.88%) and a minimum reduction of 41.42 MW (40.62%). In summary, the model proposed in this paper has a high forecasting accuracy and stability in short-term load forecasting by using the ''decomposition-reconstruction-optimization-error correction'' forecasting framework. It can effectively reduce the prediction error when the load is highly volatile, and the prediction trend has a high consistency with the real load value. However, when the model is applied to different capacity power systems, it is necessary to adjust the parameters of each part of the forecasting framework several times according to the different parameters of the application scenarios to ensure the normal use of the model. Therefore, the adaptive or tuning functions of the model parameters need to be further optimized and improved to enhance the applicability and fast responsiveness of the model.

REFERENCES

[1] P. Zeng, M. Jin, and M. F. Elahe, ''Short-term power load forecasting based on cross multi-model and second decision mechanism,'' *IEEE Access*, vol. 8, pp. 184061–184072, 2020.

[2] J. G. Jetcheva, M. Majidpour, and W.-P. Chen, ''Neural network model ensembles for building-level electricity load forecasts,'' *Energy Buildings*, vol. 84, pp. 214–223, Dec. 2014.

[3] W. Charytoniuk, M. S. Chen, and P. Van Olinda, ''Nonparametric regression based short-term load forecasting,'' *IEEE Trans. Power Syst.*, vol. 13, no. 3, pp. 725–730, Aug. 1998.

[4] J. Zheng, C. Xu, Z. Zhang, and X. Li, ''Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network,'' in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6.

[5] Y. Wang, Q. Chen, N. Zhang, and Y. Wang, ''Conditional residual modeling for probabilistic load forecasting,'' *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 7327–7330, Nov. 2018.

[6] L. Ekonomou, C. A. Christodoulo, and V. Mladenov, ''A short-term load forecasting method using artificial neural networks and wavelet analysis,'' *Int. J. Power Syst.*, vol. 1, pp. 64–68, Jul. 2016.

[7] W. R. Christiaanse, ''Short-term load forecasting using general exponential smoothing,'' *IEEE Trans. Power App. Syst.*, vol. PAS-90, no. 2, pp. 900–911, Mar. 1971.

[8] K.-B. Song, Y.-S. Baek, D. H. Hong, and G. Jang, ''Short-term load forecasting for the holidays using fuzzy linear regression method,'' *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 96–101, Feb. 2005.

[9] R. Zhang, W. Cai, L. Ni, and G. L. Lebby, ''Power system load forecasting using partial least square method,'' in *Proc. 40th Southeastern Symp. Syst. Theory (SSST)*, Mar. 2008, pp. 169–173.

[10] T. Haida and S. Muto, ''Regression based peak load forecasting using a transformation technique,'' *IEEE Trans. Power Syst.*, vol. 9, no. 4, pp. 1788–1794, Nov. 1994.

[11] C.-M. Lee and C.-N. Ko, ''Short-term load forecasting using lifting scheme and ARIMA models,'' *Exp. Syst. Appl.*, vol. 38, no. 5, pp. 5902–5911, May 2011.

[12] H. Mori and H. Kobayashi, ''Optimal fuzzy inference for short-term load forecasting,'' *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 390–396, Feb. 1996.

[13] M. Barman and N. B. Dev Choudhury, ''Season specific approach for short-term load forecasting based on hybrid FA-SVM and similarity concept,'' *Energy*, vol. 174, pp. 886–896, May 2019.

[14] G.-C. Liao, ''A novel particle swarm optimization approach combined with fuzzy neural networks for short-term load forecasting,'' in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, Jun. 2007, pp. 1–6.

[15] G. Cerne, D. Dovžan, and I. Skrjanc, ''Short-term load forecasting by separating daily profiles and using a single fuzzy model across the entire domain,'' *IEEE Trans. Ind. Electron.*, vol. 65, no. 9, pp. 7406–7415, Sep. 2018.

[16] P. Liyun, Z. Wenjun, W. Sining, and H. Lu, ''Short-term load forecasting based on DenseNet-LSTM fusion model,'' in *Proc. IEEE Int. Conf. Energy Internet (ICEI)*, Sep. 2021, pp. 84–89.

[17] J. Che and J. Wang, ''Short-term load forecasting using a kernel-based support vector regression combination model,'' *Appl. Energy*, vol. 132, pp. 602–609, Nov. 2014.

[18] W.-C. Hong, ''Hybrid evolutionary algorithms in a SVR-based electric load forecasting model,'' *Int. J. Electr. Power Energy Syst.*, vol. 31, nos. 7–8, pp. 409–417, Sep. 2009.

[19] Q. Liu, Y. Shen, L. Wu, J. Li, L. Zhuang, and S. Wang, ''A hybrid FCW-EMD and KF-BA-SVM based model for short-term load forecasting,'' *CSEE J. Power Energy Syst.*, vol. 4, no. 2, pp. 226–237, Jun. 2018.

[20] C. Tian, J. Ma, C. Zhang, and P. Zhan, ''A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network,'' *Energies*, vol. 11, no. 12, p. 3493, Dec. 2018.

[21] M. Q. Raza and A. Khosravi, ''A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings,'' *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, Oct. 2015.

[22] K. Ke, S. Hongbin, Z. Chengkang, and C. Brown, ''Short-term electrical load forecasting method based on stacked auto-encoding and GRU neural network,'' *Evol. Intell.*, vol. 12, no. 3, pp. 385–394, Jan. 2019.

[23] H. J. Sadaei, P. C. D. L. E. Silva, F. G. Guimarães, and M. H. Lee, ''Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series,'' *Energy*, vol. 175, pp. 365–377, May 2019.

[24] S. M. Elgarhy, M. M. Othman, A. Taha, and H. M. Hasanien, ''Short term load forecasting using ANN technique,'' in *Proc. 19th Int. Middle East Power Syst. Conf. (MEPCON)*, Dec. 2017, pp. 1385–1394.

[25] Y. Zhang, C. Li, Y. Jiang, L. Sun, R. Zhao, K. Yan, and W. Wang, ''Accurate prediction of water quality in urban drainage network with integrated EMD-LSTM model,'' *J. Cleaner Prod.*, vol. 354, Jun. 2022, Art. no. 131724.

[26] Z. Kong, C. Zhang, H. Lv, F. Xiong, and Z. Fu, ''Multimodal feature extraction and fusion deep neural networks for short-term load forecasting,'' *IEEE Access*, vol. 8, pp. 185373–185383, 2020.

[27] X. Gao, X. Li, B. Zhao, W. Ji, X. Jing, and Y. He, ''Short-term electricity load forecasting model based on EMD-GRU with feature selection,'' *Energies*, vol. 12, no. 6, p. 1140, Mar. 2019.

[28] S. Zhang, J. Luo, S. Wang, and F. Liu, ''Oil price forecasting: A hybrid GRU neural network based on decomposition-reconstruction methods,'' *Exp. Syst. Appl.*, vol. 218, May 2023, Art. no. 119617.

[29] Z. Wu, G. Luo, Z. Yang, Y. Guo, K. Li, and Y. Xue, ''A comprehensive review on deep learning approaches in wind forecasting applications,'' *CAAI Trans. Intell. Technol.*, vol. 7, no. 2, pp. 129–143, Jan. 2022.

[30] Y. Zhang, Y. Li, and G. Zhang, ''Short-term wind power forecasting approach based on Seq2Seq model using NWP data,'' *Energy*, vol. 213, Dec. 2020, Art. no. 118371.

[31] M. A. R. Suleman and S. Shridevi, ''Short-term weather forecasting using spatial feature attention based LSTM model,'' *IEEE Access*, vol. 10, pp. 82456–82468, 2022.

[32] M. Sibtain, X. Li, S. Saleem, M. S. Asad, T. Tahir, and H. Apaydin, ''A multistage hybrid model ICEEMDAN-SE-VMD-RDPG for a multivariate solar irradiance forecasting,'' *IEEE Access*, vol. 9, pp. 37334–37363, 2021.

[33] Y. Gao, Y. Hang, and M. Yang, ''A cooling load prediction method using improved CEEMDAN and Markov chains correction,'' *J. Building Eng.*, vol. 42, Oct. 2021, Art. no. 103041.

[34] M. Malinda and J.-H. Chen, "The forecasting of consumer exchange-traded funds (ETFs) via grey relational analysis (GRA) and artificial neural network (ANN)," *Empirical Econ.*, vol. 62, no. 2, pp. 779–823, Feb. 2022.
[35] W. Deng, J. Xu, Y. Song, and H. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Appl. Soft Comput.*, vol. 100, Mar. 2021, Art. no. 106724.
[36] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and H. Faris, "MTDE: An effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 106761.
[37] S. Li, Q. Gu, W. Gong, and B. Ning, "An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models," *Energy Convers. Manag.*, vol. 205, Feb. 2020, Art. no. 112443.
[38] X. Yu, C. Li, and J. Zhou, "A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios," *Knowl.-Based Syst.*, vol. 204, Sep. 2020, Art. no. 106209.
[39] J. Wang, H. Zhu, Y. Zhang, F. Cheng, and C. Zhou, "A novel prediction model for wind power based on improved long short-term memory neural network," *Energy*, vol. 265, Feb. 2023, Art. no. 126283.
[40] Y. Sun, X. Wang, Y. Chen, and Z. Liu, "A modified whale optimization algorithm for large-scale global optimization problems," *Exp. Syst. Appl.*, vol. 114, pp. 563–577, Dec. 2018.
[41] P. Civicioglu and E. Besdok, "Bernstain-search differential evolution algorithm for numerical function optimization," *Exp. Syst. Appl.*, vol. 138, Dec. 2019, Art. no. 112831.
[42] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.
[43] C. Li, G. Li, K. Wang, and B. Han, "A multi-energy load forecasting method based on parallel architecture CNN-GRU and transfer learning for data deficient integrated energy systems," *Energy*, vol. 259, Nov. 2022, Art. no. 124967.
[44] T. Chen, W. Huang, R. Wu, and H. Ouyang, "Short term load forecasting based on SBiGRU and CEEMDAN-SBiGRU combined model," *IEEE Access*, vol. 9, pp. 89311–89324, 2021.
[45] Y. Su and X. Jiang, "Prediction of tide level based on variable weight combination of LightGBM and CNN-BiGRU model," *Sci. Rep.*, vol. 13, no. 1, p. 5, Jan. 2023.
[46] D. Y. Deng, J. Li, Z. Y. Zhang, Y. F. Teng, and Q. Hhuang, "Short-term electric load forecasting based on EEMD-GRU-MLR," *Power Syst. Technol.*, vol. 44, no. 2, pp. 593–602, Feb. 2020.
[47] D. Niu, M. Yu, L. Sun, T. Gao, and K. Wang, "Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism," *Appl. Energy*, vol. 313, May 2022, Art. no. 118801.
[48] C. Zhang, T. Peng, and M. S. Nazir, "A novel integrated photovoltaic power forecasting model based on variational mode decomposition and CNN-BiGRU considering meteorological variables," *Electric Power Syst. Res.*, vol. 213, Dec. 2022, Art. no. 108796.
[49] T. Ma, G. Xiang, Y. Shi, and Y. Liu, "Horizontal in situ stresses prediction using a CNN-BiLSTM-attention hybrid neural network," *Geomechanics Geophys. Geo-Energy Geo-Resour.*, vol. 8, no. 5, p. 152, Sep. 2022.
[50] X. Yao, X. Fu, and C. Zong, "Short-term load forecasting method based on feature preference strategy and LightGBM-XGboost," *IEEE Access*, vol. 10, pp. 75257–75268, 2022.

LIANBING LI received the bachelor's degree in electrical automation from the Hebei University of Technology, Tianjin, in 1993, the M.S. degree in power electronics and power transmission from Tianjin University, in 1999, the Ph.D. degree in electric motors and appliances from the Hebei University of Technology, in 2003, and the Ph.D. degree from The University of Tokyo, Japan, in 2005. He is currently a Professor and a Master Supervisor with the Hebei University of Technology and also the Director of the China Electrotechnical Society, the Vice Chairman and a General Secretary of the Hebei Electrotechnical Society, and the Vice Chairman of the Tianjin Power Supply Society. He is mainly engaged in new energy generation, digital active distribution networks, power storage systems, machine learning, and data mining.

RUIXIONG JING (Graduate Student Member, IEEE) received the bachelor's degree in automation from the Tianjin University of Commerce, Tianjin, China, in 2021. He is currently pursuing the M.S. degree in electronic information with the Hebei University of Technology, Tianjin. His research interests include distribution network monitoring platform, power quality management, large data analysis, and machine learning. He is mainly working on the data decomposition and processing.

YANLIANG ZHANG received the bachelor's degree in automation from Liaoning Technical University, Liaoning, China, in 2021. He is currently pursuing the M.S. degree in electronic information with the Hebei University of Technology, Tianjin, China. His research interests include wind power operation and maintenance management, intelligent monitoring platform, and data analysis and forecast.

LANCHAO WANG received the bachelor's degree in automation from the Qilu University of Technology, Jinan, China, in 2021. He is currently pursuing the Graduate degree with the Hebei University of Technology. His research interests include real-time simulation of power systems, photovoltaic power generation and energy storage systems, power electronic systems, and battery management systems.

LE ZHU received the bachelor's degree in automation from the Hunan University of Science and Technology, Xiangtan, China, in 2021. He is currently pursuing the M.S. degree in control science and engineering with the Hebei University of Technology, Tianjin, China. His research interests include energy storage technology, battery energy management, and machine learning. He is mainly working on the organization and analysis of data.