

Article

# Research and Application of a Novel Hybrid Model Based on a Deep Neural Network for Electricity Load Forecasting: A Case Study in Australia

Kailai Ni, Jianzhou Wang \*, Guangyu Tang and Danxiang Wei

School of Statistics, Dongbei University of Finance and Economics, Dalian 116025, China

\* Correspondence: wangjz@dufe.edu.cn; Tel.: +86 13009480823

Received: 17 May 2019; Accepted: 17 June 2019; Published: 26 June 2019



**Abstract:** Electricity load forecasting plays an essential role in improving the management efficiency of power generation systems. A large number of load forecasting models aiming at promoting the forecasting effectiveness have been put forward in the past. However, many traditional models have no consideration for the significance of data preprocessing and the constraints of individual forecasting models. Moreover, most of them only focus on the forecasting accuracy but ignore the forecasting stability, resulting in nonoptimal performance in practical applications. This paper presents a novel hybrid model that combines an advanced data preprocessing strategy, a deep neural network, and an avant-garde multi-objective optimization algorithm, overcoming the defects of traditional models and thus improving the forecasting performance effectively. In order to evaluate the validity of the proposed hybrid model, the electricity load data sampled in 30-min intervals from Queensland, Australia are used as a case to study. The experiments show that the new proposed model is obviously superior to all other traditional models. Furthermore, it provides an effective technical forecasting means for smart grid management.

**Keywords:** electricity load forecasting; hybrid model; data preprocessing strategy; multi-objective optimization algorithm; deep neural network

## 1. Introduction

With the development of productivity and society, the demand for electricity for production and living is growing constantly, which has also led to an increased difficulty in power system management. Against this background, electricity load forecasting is of great help for the decision-making process of power market participants and regulators [1,2]. However, affected by many potential factors [3], it is a challenging task to conduct significant work in this field. Exaggerated forecasting can lead to excessive electricity production, which increases unnecessary operating costs and wastes energy. On the other hand, inadequate forecasting can lead to a shortage in energy production, posing political, economic, and security threats to a country or a region.

For decades, many models have been proposed in the field of load forecasting, which can be divided into three general types: statistical models, artificial intelligence (AI) models, and hybrid models.

In statistical models, a potential dynamic relationship between current information and historical data is deemed to exist, and this relationship is described using mathematical statistics methods under strict assumptions. Models of this category, such as the Auto Regressive (AR) model [4], the Auto Regressive Moving Average (ARMA) model [5], the Auto Regressive Integrated Moving Average (ARIMA) model [6], and the Seasonal Model (SM) [7], have been applied to electricity load forecasting for many years. In 2011, Li et al. [8] proposed an improved Grey Model (GM) for use in short-term load forecasting. This model adopted a second-order, univariate structure, which overcame the problem

of the GM (1,1) being weak in forecasting time series with strong randomness. In 2016, Dudek [9] proposed a univariate, short-term load forecasting framework based on the Linear Regression (LR) and a periodic pattern that was able to filter out trends and seasonal factors longer than the daily cycle, thus eliminating the non-stationarity of the mean and variance and simplifying the forecasting problem.

From the end of the 20th century until now, owing to the rapid development of computer technology, Artificial Intelligence (AI) forecasting methods have received unprecedented attention and rapidly spread in a short time. In the past two decades, many models with different structures based on AI have been designed and employed in the field of load forecasting, for example, the Artificial Neural Network (ANN) [10], Self-Organizing Map (SOM) [11], and Adaptive Network-based Fuzzy Inference System (ANFIS) [12]. In 2008, Lauret et al. [13] constructed a model on the basis of the Bayesian Neural Network (BNN) with obvious advantages over traditional neural networks and applied it to the forecasting of short-term load data. In 2017, a model based on Support Vector Regression (SVR) was proposed by Chen et al. [14], where the previous environment temperature of two hours before demand response events was utilized as an input variable to conduct load forecasting of office buildings, thereby determining the load baseline. Many scientific studies and practical applications indicate that, in a wide variety of cases of time series forecasting, AI technology tends to have better performance than traditional statistical models.

In recent years, with the invention of a variety of forecasting techniques, many hybrid models have been put forward and utilized in various fields. More specifically, it is reasonable to put hybrid forecasting models into two categories. The first category is usually based on an individual forecasting method with the addition of a data preprocessing strategy or an intelligent optimization algorithm or both, forming a model with a multi-layer structure [15]. Examples of the application of such models in load forecasting are given below. In 2018, Barman et al. [16] proposed a hybrid short-term load forecasting model based on the Support Vector Machine (SVM), which employs the Grasshopper Optimization Algorithm (GOA) to optimize network parameters to achieve high precision. Li et al. [17] proposed a hybrid model based on the Extreme Learning Machine (ELM), which incorporates a classical data preprocessing strategy. Rana et al. [18] proposed a hybrid model called the Advanced Wavelet Neural Network (AWNN). The model firstly decomposes the raw data with a modified wavelet-based strategy and then uses a neural network to forecast. More examples are presented in [19–23]. In addition, models of this category are also widely used in other fields such as wind speed forecasting [24,25], air pollution forecasting [26], and forecasting in some high-dimensional data [27,28]. Through combinations of different data preprocessing strategies, simple statistical or artificial intelligence forecasting modules, and intelligent optimization algorithms, various hybrid models of this category have been invented. Models in the second category are also called combined forecasting models. The combined forecast theory was initially expounded by Bates and Granger in 1969 [29], whose core idea was to merge the forecasting results of multiple sub-models in a weighted manner. In [30,31], combined forecasting models were applied to wind speed forecasting. In [32], Shen et al. applied a combined forecasting model to international tourism demand forecasting. In [33], Jiang et al. employed a combined model for the forecasting of carbon emissions. In the field of electricity load forecasting, Xiao et al. [34] constructed a model based on multiple neural networks in 2015 and compared it with ARIMA. The comparison showed the advantages of the combined model in terms of the forecasting ability.

**A review of various models proposed in previous literature showed that they have many insurmountable problems, which are summarized below.**

(1) Due to the overly strict assumptions of statistical models that linear relationships exist within the time series, it is difficult for data in real life to fully meet the required conditions. Therefore, in a lot of fields, bad results are often obtained, especially for nonlinear and nonstationary data with high noise and fluctuations [35].

(2) It is worth mentioning that although AI technology can better extract the nonlinear characteristics of data, it also has some disadvantages that are difficult to overcome. For example, AI forecasting methods are prone to fall into local optimization and generate an overfitting phenomenon [36].

(3) To some extent, hybrid models are able to take full advantage of each module, but at the same time, they may produce new defects, which deserve special attention.

First, most studies emphasize the forecasting accuracy, thus underestimating the significance of forecasting stability. It can be found that most of the hybrid models use single-objective optimization algorithms including Particle Swarm Optimization (PSO) [37], the Genetic Algorithm (GA) [38], the Evolutionary Algorithm (EA) [39], the Firefly Algorithm (FA) [40], or the Cuckoo Search Algorithm (CSA) [41,42]. These algorithms can help to improve the forecasting accuracy only but are unable to improve the forecasting stability simultaneously. However, forecasting accuracy and stability are equally important for a model [43]. The obsession with the former and the neglect of the latter may lead to confusing security problems in applications.

Secondly, many individual forecasting methods used in hybrid models have a limited ability to learn the data features comprehensively. It can be found that a large number of hybrid models use statistical methods or AI methods with the simple structures mentioned above. The application of these methods makes the models lack sufficient global learning ability, which will result in nonoptimal forecasting performance.

Finally, the data preprocessing strategies mainly including Empirical Mode Decomposition (EMD) [44–46], Wavelet Transform (WT) [47,48], and the Singular Spectral Analysis (SSA) [49] are not powerful enough to effectively remove outliers and noise in data, thus affecting the results.

**Therefore, it is urgent to propose a novel electricity load forecasting model which contains the advantages of each module and overcome the disadvantages mentioned above.**

Hopefully, more and more multi-objective optimization algorithms will be invented to solve Multi-Objective Problems (MOPs) in various fields. There are quite a few examples, like the Multi-Objective Particle Swarm Optimization (MOPSO), which is applied in micro-grid system management [50]; the Non-dominated Sorting Genetic Algorithm-II (NSGA-II), which is applied in redundancy allocation problems [51]; the Multi-Objective Whale Optimization Algorithm (MOWOA), which is applied in wind speed forecasting [52]; and the Multi-Objective Evolutionary Algorithm (MOEA), which is utilized in optimizing traffic flow and vehicle emission planning through urban traffic lights [53]. Multi-objective optimization algorithms can effectively solve problems among multiple conflicting objectives, making the results more in line with the actual needs.

As a popular term, deep neural networks have been successfully used in engineering, economy, security, and other fields. In [54], a model based on the Convolutional Neural Network (CNN) was applied in facial expression recognition. In [55], a model based on the Deep Belief Network (DBN) was applied in the field of medical X-ray image analysis. In [56], a model based on the Long Short-Term Memory network (LSTM) was applied in financial market forecasting. In 2017, a short-term electricity load forecasting model based on deep neural networks was proposed and good experimental results were obtained [57]. In summary, compared with other methods, deep neural networks have more powerful nonlinear mapping abilities and can extract the deeper characteristics of data. Therefore, when deep neural networks solve nonlinear modeling problems, surprising results may be achieved.

In addition, with the development of signal processing research, researchers have invented some novel and effective denoising strategies and applied them to the data preprocessing of time series. For example, strategies such as the Wavelet Packet Transform (WPT) [58], Improved Empirical Mode Decomposition (IEMD) [59], and Ensemble Empirical Mode Decomposition (EEMD) [60] have been successfully employed in the field of electricity load forecasting to reduce the random disturbance of original data, thus obtaining a better forecasting performance.

In this paper, a novel hybrid model for electricity load forecasting based on a deep neural network is successfully proposed. The model is improved by a multi-objective optimization algorithm and an advanced data preprocessing strategy. In the proposed model, DBN is used

as the core module of data feature learning and forecasting. Meanwhile, the Multi-Objective Grey Wolf Optimizer (MOGWO) is employed to search for the optimal initial weights and thresholds of DBN. In addition, the Complementary Ensemble Empirical Mode Decomposition (CEEMD), an advanced signal processing strategy, is applied in the data preprocessing procedure to remove noise existing in the load series. Finally, scientific and reasonable evaluation methods including various metrics are employed to conduct a comprehensive assessment.

The proposed model successfully introduces a deep neural network into electricity load time series forecasting. In terms of the construction of datasets, this paper divides data sampled in 30-min intervals from Queensland into seven datasets corresponding to Monday to Sunday, respectively. Meanwhile, this paper takes the previous 16 real data samples of each forecasting time point as the input variable of the proposed model, and the benchmark models also follow the above principles when constructing their input variables. The model learns each dataset separately and outputs the results of one-step and multi-step rolling forecasting. The fine results of the proposed model show its excellent forecasting accuracy and stability in modeling data with complex components like load series.

**The highlights of the study are as follows:**

**(1) Based on an emerging deep neural network and improved by an avant-garde multi-objective optimization algorithm as well as an effective data preprocessing strategy, a complex and systematic hybrid forecasting model is constructed.** The proposed model can effectively combine the advantages of each module in the structure and thus has better forecasting performance than individual models and hybrid models composed of other simple structures. As it turns out, the proposed model is superior to all compared traditional models.

**(2) An algorithm for MOPs is utilized in the proposed model to help to determine the initial network weights and thresholds, thereby promoting the forecasting accuracy and stability simultaneously.** This algorithm is an intelligent heuristic optimizer, which iterates according to Pareto's theory and the bionics principle of the preying behavior of wolves, thus successfully converging to the Pareto optimal fronts of the MOPs and searching for the optimal network parameters.

**(3) A powerful denoising strategy is utilized in the preprocessing of electricity load data, which can effectively identify high-frequency noise and remove it to reduce the impact of fluctuations on the forecasting performance.** This strategy decomposes and reconstructs the original load series into several sub-sequences, so as to filter out the high-frequency fluctuations in information in the original series and avoid them entering the subsequent data learning process.

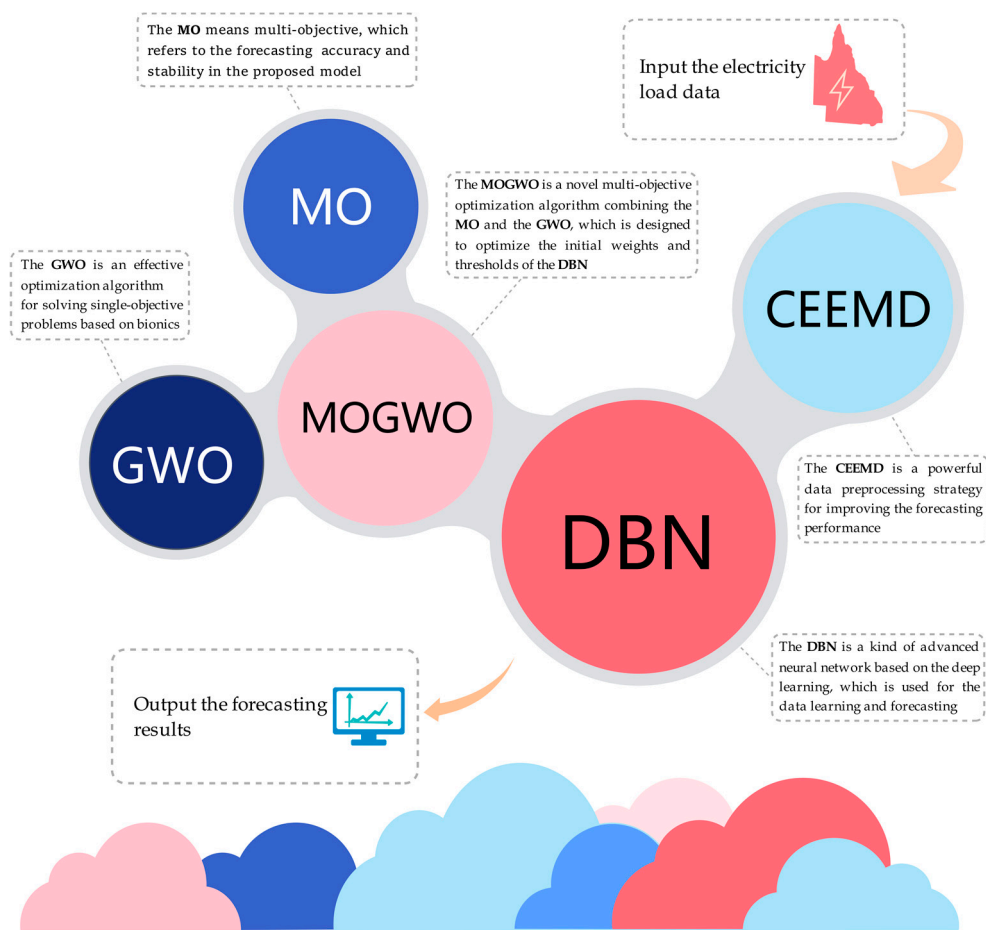
**(4) The core of the proposed model is a deep neural network, which has a stronger nonlinear mapping and characterization ability than traditional neural networks and statistical methods, due to its special structure and principles.** This module is able to conduct comprehensive learning and training for the characteristics and patterns contained in the electricity load series, thus contributing to the satisfying forecasting performance of the proposed model.

**(5) The forecasting results are evaluated reasonably and comprehensively by multiple metrics. Meanwhile, in-depth and rigorous discussions are carried out in this paper.** Six of the metrics selected are adopted to assess forecasting errors, and the remaining one is used to evaluate the convergence performance of algorithms for MOPs. Moreover, the results of the experiments are further dissected from several perspectives to validate the superiority of the model that is proposed in the study.

The rest is arranged below. The framework of the proposed model is introduced in Section 2. More details of the methodology are presented in Section 3. The ideas and steps for effective hypothesis testing are expounded in Section 4. Section 5 analyzes the results of the three experiments. In Section 6, six discussions based on the experimental results are presented. Finally, Section 7 gives the conclusion of this paper.

## 2. The Framework of the Proposed Model

The framework of the proposed model is shown in Figure 1. It can be described as follows:



**Figure 1.** The framework of the proposed model.

(1) In general, the model consists of two parts. The first part contains one module: an advanced data preprocessing strategy, and the second part contains two modules: the one is a data learning and forecasting procedure, and the other is an optimizer for network parameters.

(2) In the first part, the advanced signal denoising strategy, CEEMD, is applied as a data preprocessing module to eliminate noise to avoid it having an adverse impact on the forecasting results. The raw data is decomposed and reconstructed into a finite number of Intrinsic Mode Functions (IMFs), and afterwards, each IMF is used separately as an independent dataset input for the next part.

(3) In the second part, DBN is employed to learn the data characteristics and output forecasting values of each IMF, while MOGWO is utilized to optimize the parameters of DBN. The outputs of IMFs are then merged, forming the final forecasting results of the proposed model. It should be stressed that, in the merging process, several IMFs may not be included because they contain too much noise.

### 3. Methodology

In this section, more details are expounded according to the modules of the proposed model, as mentioned above. In turn, the concepts and implementations of CEEMD, DBN, and MOGWO are introduced.

#### 3.1. Complementary Ensemble Empirical Mode Decomposition

CEEMD, first put forward by Yeh et al. [61], is an improved strategy of EEMD, proposed by Wu et al. [62]. It is applicable to the decomposition of non-linear and non-stationary data with high-frequency noise. The procedure is below:



**Step 1.** Add  $m$  groups of Gaussian white noise with the same amplitude and the opposite phase to the original data:

$$\begin{bmatrix} P_i(t) \\ N_i(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} S_0(t) \\ G_i(t) \end{bmatrix}, \quad i = 1, 2, 3, \dots, m \quad (1)$$

where  $S_0(t)$  denotes the raw data, and  $G_i(t)$  denotes the Gaussian white noise sequence of group  $i$ .

**Step 2.**  $P_i(t)$  and  $N_i(t)$  are decomposed by the EMD strategy:

$$\begin{cases} P_i(t) = \sum_{j=1}^n imf_{ij}(t) \\ N_i(t) = \sum_{j=1}^n imf_{-ij}(t) \end{cases}, \quad j = 1, 2, 3, \dots, n \quad (2)$$

where  $imf_{ij}(t)$  refers to the  $j$ th IMF after  $P_i(t)$  is decomposed by EMD. Accordingly,  $imf_{-ij}(t)$  refers to the  $j$ th IMF after  $N_i(t)$  is decomposed by EMD.

**Step 3.** Calculate the mean value of the  $j$ th IMF for all groups of  $P_i(t)$  and  $N_i(t)$ :

$$IMF_j(t) = \frac{1}{2m} \sum_{i=1}^m (imf_{ij}(t) + imf_{-ij}(t)), \quad \begin{cases} i = 1, 2, 3, \dots, m \\ j = 1, 2, 3, \dots, n \end{cases} \quad (3)$$

where  $IMF_j(t)$  represents the final outputs of CEEMD, namely the  $j$ th IMF after the raw load data has been decomposed by CEEMD.

### 3.2. Deep Belief Network

The concept of the DBN was initially put forward by Hinton et al. [63] in 2006. The DBN consists of two components: The Restricted Boltzmann Machine (RBM) and the Back Propagation (BP) algorithm.

#### 3.2.1. Restricted Boltzmann Machine

The RBM is a kind of unsupervised neural network. Each RBM has a structure with two layers: a visible one and a hidden one. Internally, no links exist inside each layer, while full connections are adopted between layers.

The RBM can be explained using stochastic neural network theory. It is an energy-based model inspired by statistical mechanics. The energy of the joint configuration of the visible variable  $\mathbf{v}$  and the hidden variable  $\mathbf{h}$  can be expressed as

$$E(\mathbf{v}, \mathbf{h}; \mathbf{S}) = - \sum_{i,j} W_{ij} v_i h_j - \sum_i q_i v_i - \sum_j p_j h_j \quad (4)$$

where  $\mathbf{S}$  stands for RBM's parameter  $\{\mathbf{W}, \mathbf{q}, \mathbf{p}\}$ . Thereinto,  $\mathbf{W}$  refers to the weight vector between  $\mathbf{v}$  and  $\mathbf{h}$ , and  $\mathbf{q}$  and  $\mathbf{p}$  refer to the biases of  $\mathbf{v}$  and  $\mathbf{h}$ , accordingly.

Then, the joint probability distribution of  $\mathbf{v}$  and  $\mathbf{h}$  is established by Boltzmann distribution, which can be formulized as

$$P_S(\mathbf{v}, \mathbf{h}) = \frac{1}{Z(\mathbf{S})} \exp(-E(\mathbf{v}, \mathbf{h}; \mathbf{S})) = \frac{1}{Z(\mathbf{S})} \prod_{i,j} \exp(W_{ij} v_i h_j) \prod_i \exp(q_i v_i) \prod_j \exp(p_j h_j) \quad (5)$$

where  $Z(\mathbf{S})$  is the normalization factor, which can be expressed as:

$$Z(\mathbf{S}) = \sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \mathbf{S})). \quad (6)$$

The learning goal of RBM is to maximize  $P_S(\mathbf{v})$ , which refers to the marginal distribution of  $P_S(\mathbf{v}, \mathbf{h})$ :

$$P_S(\mathbf{v}) = \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h}; \mathbf{S}) = \frac{1}{Z(\mathbf{S})} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \mathbf{S})). \quad (7)$$

Usually, there are several RBMs in a DBN which are stacked vertically. In the training of the DBN, the RBM of each layer is separately trained without supervision. This step is called pre-training in deep learning.

### 3.2.2. Back Propagation Algorithm

The last layer of DBN is set to the BP algorithm. In this layer, the output vector of the RBM is used as the input vector for supervised learning. The BP algorithm is applied in DBN to propagate errors backward to the RBM of each layer and adjust the whole network, thus making it a complete system. This step is called fine tuning in deep learning.

### 3.3. Multi-Objective Grey Wolf Optimizer

The MOGWO was put forward by Mirjalili et al. [64] to cope with the optimization problems of multiple conflicting objectives, which is on the basis of leadership in society and the predation behavior of grey wolves. Generally, MOGWO is a modification of the Grey Wolf Optimizer (GWO) [65]. The main contents of GWO, MOP, and new mechanisms are described below. In addition, the pseudo-codes describing the process of how the MOGWO optimizes the DBN are shown in Algorithm 1.

#### 3.3.1. Grey Wolf Optimizer

The GWO is a single-target algorithm that was developed by drawing inspiration from the behaviors of grey wolves. The details of the GWO are as follows.

**Definition 1:** *Hierarchy.*

*There is a strict hierarchy in a grey wolf population. Let us assume that there are wolves of four types:  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  in a population, where the predation behavior is led by  $\alpha$ ,  $\beta$ , and  $\gamma$ , while the remaining  $\delta$  wolves must submit to their leadership.*

**Definition 2:** *Encircling the prey.*

*Let  $\mathbf{M}$  be the distance between the predators and the prey, which can be formulized as*

$$\mathbf{M} = |\mathbf{A} \cdot \mathbf{X}_0^t - \mathbf{X}^t| \quad (8)$$

*where  $\mathbf{X}_0^t$  denotes the location of the current prey objective,  $\mathbf{X}^t$  denotes the location of the current predator, and  $\mathbf{A}$  is the wobble coefficient.*

*The grey wolf then updates its position based on the distance between itself and the prey:*

$$\mathbf{X}^{t+1} = \mathbf{X}_0^t - \mathbf{B} \cdot \mathbf{M} \quad (9)$$

*where,  $\mathbf{X}^{t+1}$  represents the position of a predator in the next iteration, and  $\mathbf{B}$  is the convergence coefficient vector.*

*When all the grey wolves update their positions according to the above equations, they have encircled the prey once.*

**Definition 3:** *Hunting.*

*To hunt more effectively, the locations of the three best-positioned grey wolves (with optimal fitness) are used to locate the remaining  $\delta$  wolves:*

$$\mathbf{M}_\alpha = |\mathbf{A}_1 \cdot \mathbf{X}_\alpha^t - \mathbf{X}^t| \quad (10)$$

$$\mathbf{M}_\beta = \left| \mathbf{A}_2 \cdot \mathbf{X}_\beta^t - \mathbf{X}^t \right| \quad (11)$$

$$\mathbf{M}_\gamma = \left| \mathbf{A}_3 \cdot \mathbf{X}_\gamma^t - \mathbf{X}^t \right| \quad (12)$$

$$\mathbf{X}_1 = \mathbf{X}_\alpha^t - \mathbf{B}_1 \cdot \mathbf{M}_\alpha \quad (13)$$

$$\mathbf{X}_2 = \mathbf{X}_\beta^t - \mathbf{B}_2 \cdot \mathbf{M}_\beta \quad (14)$$

$$\mathbf{X}_3 = \mathbf{X}_\gamma^t - \mathbf{B}_3 \cdot \mathbf{M}_\gamma \quad (15)$$

$$\mathbf{X}^{t+1} = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3} \quad (16)$$

where  $\mathbf{X}_\alpha^t$ ,  $\mathbf{X}_\beta^t$ , and  $\mathbf{X}_\gamma^t$  represent the current positions of wolves  $\alpha$ ,  $\beta$ , and  $\gamma$ , respectively, and  $\mathbf{X}^t$  represents the current position of a certain  $\delta$  grey wolf.  $\mathbf{M}_\alpha$ ,  $\mathbf{M}_\beta$ , and  $\mathbf{M}_\gamma$  represent the distances from wolf  $\alpha$ ,  $\beta$ , and  $\gamma$  to wolf  $\delta$ , respectively. Then,  $\mathbf{X}^{t+1}$  defines the final position of wolf  $\delta$ . In addition,  $\mathbf{A}_1$ ,  $\mathbf{A}_2$ , and  $\mathbf{A}_3$  are vectors between 0 and 2, and  $\mathbf{B}_1$ ,  $\mathbf{B}_2$ , and  $\mathbf{B}_3$  are vectors between  $-1$  and  $1$ .

---

**Algorithm 1: MOGWO-DBN**


---

**Input:**

$x_i^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(p))$ —a sequence of training data

$x_f^{(0)} = (x^{(0)}(p+1), x^{(0)}(p+2), \dots, x^{(0)}(p+l))$ —a sequence of testing data

**Output:**

$\hat{y}_f^{(0)} = (\hat{y}_f^{(0)}(p+1), \hat{y}_f^{(0)}(p+2), \dots, \hat{y}_f^{(0)}(p+l))$ —a sequence of forecasting data

**Parameters:**

$Iter_{Max}$ —the maximum number of iterations

$n$ —the number of grey wolves

$t$ —the current iteration number

$\mathbf{X}_i$ —the position of wolf  $i$

$\mathbf{a}$ —the random vector in  $[0,1]$

$\mathbf{b}$ —the constant vector in  $[0,2]$

$\mathbf{c}$ —the random vector in  $[0,1]$

$F_i$ —the fitness function of wolf  $i$

```

1: /*Set the parameters of the MOGWO and the DBN*/
2: /*Initialize the grey wolf population  $\mathbf{X}_i$  ( $i = 1, 2, \dots, n$ ) randomly*/
3: /*Initialize  $\mathbf{b}$ ,  $\mathbf{B}$ , and  $\mathbf{A}$ */
4: /*Define the archive size*/
5: FOR EACH  $i: 1 \leq i \leq n$  DO
6: Evaluate the corresponding fitness function  $F_i$  for each search agent
7: END FOR
8: /*Find the non-dominated solutions and initialized the archive with them*/
9:  $\mathbf{X}_\alpha = \text{SelectLeader}(\text{archive})$ 
10: /*Exclude alpha from the archive to avoid selecting the same leader*/
11:  $\mathbf{X}_\beta = \text{SelectLeader}(\text{archive})$ 
12: /*Exclude beta from the archive to avoid selecting the same leader*/
13:  $\mathbf{X}_\gamma = \text{SelectLeader}(\text{archive})$ 
14: /*Add back alpha and beta to the archive*/
15: WHILE ( $t < Iter_{Max}$ ) DO
16: FOR EACH  $i: 1 \leq i \leq n$  DO
17: /*Update the position of the current search agent*/
18:  $\mathbf{M}_j = |\mathbf{A}_j \cdot \mathbf{X}_j - \mathbf{X}|, i = 1, 2, 3; j = \alpha, \beta, \gamma$ 
19:  $\mathbf{X}_i = \mathbf{X}_j - \mathbf{B}_j \cdot \mathbf{M}_j, i = 1, 2, 3; j = \alpha, \beta, \gamma$ 
20:  $\mathbf{X}(t+1) = (\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3) / 3$ 
21: END FOR
22: /*Update  $\mathbf{b}$ ,  $\mathbf{B}$ , and  $\mathbf{A}$ */
23:  $\mathbf{B} = 2 \cdot \mathbf{b} \cdot \mathbf{c} - \mathbf{b}; \mathbf{A} = 2 \cdot \mathbf{a}$ 
24: /*Evaluate the corresponding fitness function  $F_i$  for each search agent*/

```

---



---

```

25: /*Find the non-dominated solutions*/
26: /*Update the archive regarding the obtained non-dominated solutions*/
27: IF the archive is full DO
28: /*Delete one solution from the current archive members*/
29: /*Add the new solution to the archive*/
30: END IF
31: IF any newly added solutions to the archive are outside the hypercubes DO
32: /*Update the grids to cover the new solution(s)*/
33: END IF
34:  $X_\alpha = \text{SelectLeader}(\text{archive})$ 
35: /*Exclude alpha from the archive to avoid selecting the same leader*/
36:  $X_\beta = \text{SelectLeader}(\text{archive})$ 
37: /*Exclude beta from the archive to avoid selecting the same leader*/
38:  $X_\gamma = \text{SelectLeader}(\text{archive})$ 
39: /*Add back alpha and beta to the archive*/
40:  $t = t + 1$ 
41: END WHILE
42: RETURN archive
43: OBTAIN  $X^* = \text{SelectLeader}(\text{archive})$ 
44: Set  $X^*$  as the initial weights and thresholds of DBN
45: Use  $X^*$  to train and update the weights and thresholds of DBN
46: Input the historical data into DBN to forecast the future changes

```

---

**Definition 4:** *Attacking.*

*Attacking is the final stage of hunting, in which the wolf pack catches the prey and the prey stops moving. The process is determined by  $\mathbf{B}$ . The grey wolves will continue to hunt when  $|\mathbf{B}| < 1$ , and the wolves are forced to leave the prey when  $|\mathbf{B}| > 1$ .*

## 3.3.2. Multi-Objective Problem

It is believed that the MOP was first proposed by Italian economist Vilfredo Pareto in 1896. Generally, an MOP refers to a problem of simultaneously optimizing multiple objective functions under multiple constraint conditions.

Let  $\mathbf{D}$  be the decision vector, a MOP can be formulized as follows:

$$\min F(\mathbf{D}) = \{f_1(\mathbf{D}), f_2(\mathbf{D}), \dots, f_n(\mathbf{D})\} \quad (17)$$

$$\text{s.t.} \begin{cases} p_i(\mathbf{D}) \leq 0 (\text{or } q_i(\mathbf{D}) \geq 0), & i = 1, 2, \dots, k \\ h_j(\mathbf{D}) = 0, & j = 1, 2, \dots, l \end{cases} \quad (18)$$

Unlike problems containing single-objective optimization, there are multiple objective functions and constraints in MOPs. So, it is not desirable to evaluate a solution only based on whether a single objective is optimal or not.

Several definitions of MOP are given below:

**Definition 5:** *Pareto dominance.*

*Let  $\mathbf{v}_1$  and  $\mathbf{v}_2$  be two solutions in the feasible domain.  $\mathbf{v}_1$  dominates  $\mathbf{v}_2$  (or  $\mathbf{v}_1 > \mathbf{v}_2$ ), if and only if these two conditions are met simultaneously:*

$$\forall i \in [1, n], f_i(\mathbf{v}_1) \leq f_i(\mathbf{v}_2) \quad (19)$$

$$\exists j \in [1, n], f_j(\mathbf{v}_1) < f_j(\mathbf{v}_2) \quad (20)$$

**Definition 6:** *Pareto optimality.*

$\mathbf{u}$  is defined as the feasible region.  $\mathbf{v}_1 \in \mathbf{u}$  is the Pareto optimality if and only if the following condition is met:

$$\nexists \mathbf{v}_2 \in \mathbf{u}, F(\mathbf{v}_2) > F(\mathbf{v}_1) \quad (21)$$

**Definition 7:** Pareto optimal set.

The Pareto optimal set is the set formed by Pareto optimal solutions, which can be expressed as shown below:

$$P = \{ \mathbf{v}_1, \mathbf{v}_2 \in \mathbf{u} | \exists \mathbf{v}_2 > \mathbf{v}_1 \} \quad (22)$$

**Definition 8:** Pareto optimal front.

The set consisting of function values calculated according to solutions in the Pareto optimal set and defined to be the Pareto optimal front is formulized below:

$$PF = \{F(\mathbf{v}) | \mathbf{v} \in P\} \quad (23)$$

**Definition 9:** The fitness function of the proposed model.

$std(\mathbf{g}^f - \mathbf{g}^o)$  and MSE are set as two sub-objective functions in the proposed model, which respectively represent the forecasting stability and accuracy. More specifically, in this study, the objective function of MOGWO is

$$\min_{F(\mathbf{D})} \begin{cases} sof_1(\mathbf{D}) = std(\mathbf{g}^f - \mathbf{g}^o) \\ sof_2(\mathbf{D}) = MSE = \frac{1}{N_g} \sum_{i=1}^{N_g} (g_i^f - g_i^o)^2, \quad i = 1, 2, \dots, N_g \end{cases} \quad (24)$$

where,  $\mathbf{g}^f$  and  $\mathbf{g}^o$  represent the forecasting outputs and actual observations respectively, and  $\mathbf{D}$  is the decision vector. In the proposed model,  $\mathbf{D}$  refers to the initial weights and thresholds of the DBN.

### 3.3.3. New mechanisms

Compared with the GWO, two new mechanisms are introduced in the MOGWO: One is to use an archive to store nondominant Pareto optimal solutions, and the other is a leader selection strategy. More details are as follows.

**Definition 10:** Pareto archive.

The Pareto archive is a simple storage unit that holds the nondominant solutions. The working principles of this structure can be summarized as four points:

- New solutions dominated by at least one solution in storage will not be archived.
- New solutions that dominate at least one solution in storage will be archived, and the dominated one will be deleted.
- If there is no domination relationship between a new solution and stored solutions, the new one will be archived.
- If the size is beyond the maximum storage limit, the elimination mechanism will be enabled according to the degree of crowding.

**Definition 11:** Leader selection strategy.

Considering that the optimal individual locations under the current number of iterations have been stored in the Pareto archive, the MOGWO will search the least crowded segments of the Pareto archive and select three wolves as leaders using the roulette wheel method with probability:

$$P_i = \frac{c}{N_i^s} \quad (25)$$

where,  $P_i$  denotes the probability that each element in the  $i_{th}$  segment of Pareto archive is selected, and  $N_i^s$  denotes the number of solutions in that segment. In addition,  $c$  is a constant greater than 1.

#### 4. Hypothesis Test

In this paper, the Diebold-Mariano (DM) test is employed to verify the statistical significance of the difference in forecasting accuracy between two models [66].

The null hypothesis  $H_0$  denotes that the difference in the forecasting performance of these two models is not statistically significant under the significance level  $\alpha$ , and the alternative hypothesis  $H_1$  is the opposite, as described below:

$$H_0 : E[L(e_{1i})] = E[L(e_{2i})] \quad (26)$$

$$H_1 : E[L(e_{1i})] \neq E[L(e_{2i})] \quad (27)$$

where,  $L$  is the loss function. Now, define

$$d_i = L(e_{1i}) - L(e_{2i}) \quad (28)$$

$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i \quad (29)$$

$$\gamma_k = \frac{1}{n} \sum_{i=k+1}^n (d_i - \bar{d})(d_{i-k} - \bar{d}) \quad (30)$$

$$DM = \frac{\bar{d}}{\sqrt{(\gamma_0 + 2 \sum_{k=1}^{h-1} \gamma_k) / n}} \quad (31)$$

In general, the appropriate value for  $h$  is:  $h = \sqrt[3]{n} + 1$ .

Accept  $H_1$  and reject  $H_0$  if and only if the following condition is met:

$$|DM| > Z_{\alpha/2} \quad (32)$$

where,  $Z_{\alpha/2}$  refers to the two-tailed critical value at the significance level  $\alpha$  of the standard normal distribution.

#### 5. Experiments

This section objectively presents the process, results, and corresponding analysis of the three experiments. In addition, the data description, the performance metrics used, and the setup of the experiments are explained in detail.

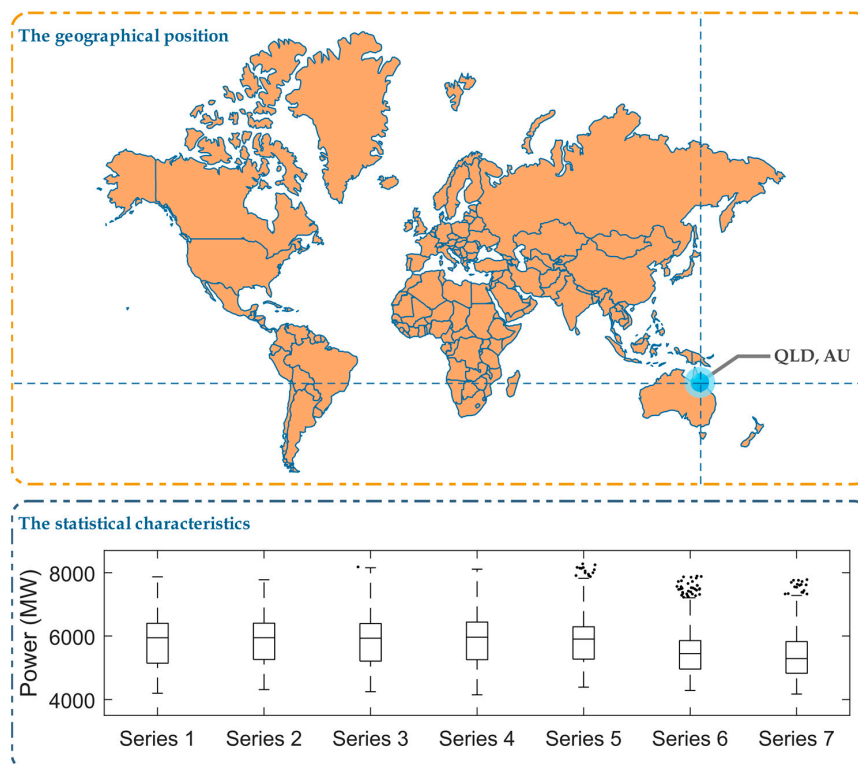
##### 5.1. Data Description

In this study, three experiments were conducted using the electricity load data from Queensland, Australia in 2013, which were sampled at 30-min intervals and can be downloaded from the Australian Energy Market Operator's website (<http://www.aemo.com.au>).

Considering the difference in daily demand pattern, the collected load data sampled at 30-min intervals were divided into seven datasets, corresponding to Monday to Sunday respectively. The forecasting strategy for this splitting method was curve estimation. In addition, we also noticed that some researchers split data by each time point, and the corresponding forecasting strategy for this splitting method is point estimation. The former data splitting method with curve estimation strategy was adopted in this study for the following three reasons. First, it considers the differences

between the behaviors of people on different days, such as when they are at work or on vacation, and treats the corresponding data respectively. The method of grouping days of the same attribute into one dataset helps to reduce the volatility of the sequence caused by the inherent differences between the characteristics of each kind of day, thus improving the forecasting accuracy. Second, the accuracy and efficiency of the model can be both considered by using the former data splitting method and forecasting strategy. In this case, the number of datasets is small, so the cost of training and forecasting is low, and the operation is convenient. Third, under the former data splitting method, there are more elements in each dataset, which means more data can be used in the learning of the model, which is more in line with the requirements of deep neural networks in terms of the size of training samples.

These data are shown in Table 1 and Figure 2. In each series, the ratio of training to testing is 3:1.



**Figure 2.** The electricity load data from Queensland (QLD).

### 5.2. The Performance Metrics

In order to comprehensively reflect the error characteristics and the forecasting performance, mean square error (MSE), normalized mean square error (NMSE), root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and Theil's inequality coefficient (TIC) were adopted, which are shown in Appendix A.

### 5.3. The Experimental Setup

Three comparative experiments were carefully set up, and the experimental process, results, and corresponding analysis are objectively presented. **Experiment I** was conducted with the major purpose of confirming the optimization ability of MOGWO and the capacity of CEEMD to preprocess data. **Experiment II** was conducted with the major purpose of verifying the relationships between the main modules in the proposed model and their influences on the forecasting performance. Finally, **Experiment III** was conducted to confirm the forecasting ability of the proposed model relative to other mainstream time series forecasting models. All experiments, except the test of MOGWO in **Experiment I**, used Series 1–7, and some key parameters were set to be the same within the proposed model, as presented in Table 2.

**Table 1.** Statistics of the electricity load data from Queensland, Australia, 2013.

	Dataset	Number	Statistical Indicator (MW)				
			Ave	Std	Median	Min	Max
<b>Series 1</b>	ALL SAMPLES	2480	5811.42475	784.52056	5952.18000	4197.77000	7868.23000
	TRAINING	1860	5803.18503	794.94666	5885.91500	4197.77000	7868.23000
	TESTING	620	5836.14390	752.45897	6067.46500	4302.44000	7138.46000
<b>Series 2</b>	ALL SAMPLES	2528	5837.81535	724.57041	5952.87500	4311.58000	7776.25000
	TRAINING	1896	5834.64836	737.46577	5933.94000	4311.58000	7776.25000
	TESTING	632	5847.31631	684.90406	5995.53000	4433.11000	7066.73000
<b>Series 3</b>	ALL SAMPLES	2480	5839.32473	776.45771	5946.47500	4246.12000	8180.74000
	TRAINING	1860	5862.24856	788.63396	5960.47000	4319.45000	8180.74000
	TESTING	620	5770.55323	735.05912	5915.85500	4246.12000	7390.58000
<b>Series 4</b>	ALL SAMPLES	2480	5854.58712	764.27322	5973.54000	4148.67000	8109.79000
	TRAINING	1860	5852.48376	765.99462	5941.06000	4315.68000	8109.79000
	TESTING	620	5860.89721	759.66630	6062.01500	4148.67000	7442.05000
<b>Series 5</b>	ALL SAMPLES	2480	5812.34784	716.67960	5911.65000	4389.57000	8278.40000
	TRAINING	1860	5809.03266	729.57825	5877.00000	4389.57000	8278.40000
	TESTING	620	5822.29340	676.98058	6012.93500	4447.88000	7180.97000
<b>Series 6</b>	ALL SAMPLES	2480	5440.03070	603.00219	5450.33500	4285.85000	7892.88000
	TRAINING	1860	5435.53583	620.85270	5433.53000	4285.85000	7892.88000
	TESTING	620	5453.51531	546.21087	5500.05000	4310.45000	6674.90000
<b>Series 7</b>	ALL SAMPLES	2480	5356.75817	656.71496	5293.97500	4172.33000	7780.52000
	TRAINING	1860	5342.39787	656.16303	5272.14500	4172.33000	7780.52000
	TESTING	620	5399.83905	657.01511	5353.88000	4250.93000	7329.04000

Due to the regularity of human behavior at different time points of a day and different days of a week, the time series of electricity load presents great seasonality and periodicity. Therefore, the time series forecasting strategy will have great significance and application prospect in this field. In this paper, a hybrid model based on the deep neural network is introduced into the time series forecasting strategy of load data. The proposed model takes the previous 16 real load data samples before each forecasting time point as the input variable, and the benchmark models in each experiment also use this principle to construct their input variables.

In addition, it is worth noting that in previous studies, many researchers tended to take temperature as an input variable for traditional models, mainly for the following two reasons. First, data collection areas such as New York and Singapore often have extreme low or high temperatures, leading to a large load on air conditioners for heating or cooling during certain periods. Second, those areas are densely populated, and when extreme weather comes, the widespread use of air conditioners causes large fluctuations in the electricity load. For the above reasons, the electricity load in those areas has a relatively large correlation with temperature, so it is considered as an important input variable by many researchers. However, Queensland, Australia, is sparsely populated, extremely low-density, and has a mild climate, according to the Ministry of Commerce of the People's Republic of China. Take Brisbane, the capital of Queensland, the third largest city in Australia, as an example. It has a total population of about 1.3 million and a population density of 12 people per hectare. The highest average annual temperature is about 24 degrees Celsius, and the lowest average annual temperature is about 15 degrees Celsius. In other words, it is hard to see a reason why people in Queensland are using appliances such as air conditioners on a large scale. Therefore, in the study of Queensland, temperature is not an appropriate input variable, and the time series forecasting strategy based on the internal correlation of the sequence itself is more effective and applicable.

**Table 2.** Some key parameters of the proposed model.

Module	Parameter	Value
CEEMD	Number of Intrinsic Mode Functions (IMFs)	11
	Ratio that divide the Std of the added noise by that of the original data	0.5
	Number of iterations	100
MOGWO	Archive size	20
	Population size	20
	Number of iterations	10
DBN	Number of iterations	100
	Number of input nodes	16
	Number of hidden layers	2
	Number of nodes in the first hidden layer	31
	Number of nodes in the second hidden layer	31

**Note:** These parameters were adopted for the proposed model in all experiments except for the test of the Multi-Objective Grey Wolf Optimizer (MOGWO) in **Experiment 1**. CEEMD: Complementary Ensemble Empirical Mode Decomposition; DBN: Deep Belief Network.

In the process of forecasting, the neural network structure is not fully extended to the next moment in terms of some internal parameters. The corresponding test dataset on Tuesday had 632 elements, so the model learnt 632 times during the forecasting of this test dataset. The test datasets corresponding to other days contained 620 elements, so the model learnt 620 times during the forecasting of them.

The data in each dataset were sampled at intervals of 30 min, and 48 data points were taken as a period. There were no missing values in the datasets. Therefore, for the corresponding dataset on Tuesday, the model forecasted a total of 13.16667 periods, while for the corresponding datasets on the other days, the model forecasted a total of 12.91667 periods. In terms of the parameters of the model, some important parameters, which are shown in Table 2, remained unchanged in each forecasting period, while the parameters obtained by neural network learning automatically changed in each forecasting process.

All experiments were conducted in MATLAB R2018a (MathWorks, Natick, MA, USA), with a computing environment running on Microsoft Windows 10 with a 64-bit, 2.60 GHz Intel Core i7 6700HQ CPU and 8.00 GB of RAM.



#### 5.4. Experiment I

The experiment had two parts. The first part was the validation of the optimization ability of MOGWO, and the second one was a test of the effectiveness of CEEMD.

##### 5.4.1. Test of MOGWO

The purpose of this part was to validate the fitting capacity of MOGWO to converge to the real Pareto optimal fronts. The Multi-Objective Dragonfly Algorithm (MODA) and the Multi-Objective Particle Swarm Optimization (MOPSO) were adopted as the controls. MODA is an intelligent swarm multi-objective optimization algorithm based on the hunting behavior of the dragonfly population proposed in recent years, and MOPSO is also a widely used heuristic multi-objective optimization algorithm based on the hunting behavior of the bird population. Their programming mechanisms are similar to that of MOGWO, as all of them are based on the biomimetic principles of animal predation. However, due to the different internal structures of the programs, the search ability of Pareto optimal solutions between these three algorithms is different. To explore this difference and the superiority of MOGWO's search capability, the ZDT functions ZDT1–3 were employed as test problems. In terms of the performance metric, the Inverted Generational Distance (IGD) [64], well-known for the evaluation of algorithms for MOPs, was selected. The test functions are shown in Appendix B, and the formula of IGD is as follows:

$$IGD = \frac{1}{N} \sqrt{\sum_{i=1}^N d_i^2(\mathbf{P}, \mathbf{P}^*)} \quad (33)$$

where  $d_i(\mathbf{P}, \mathbf{P}^*)$  denotes the distance between a point on the obtained Pareto optimal front and the nearest point on the real Pareto optimal front.

MOGWO's key parameters were set as in Table 3, and the common parameters of these three optimizers were set to be the same. In order to eliminate the influence of accidental factors on the experimental results, the experiment was repeated 50 times for each test function. The results are presented in Table 4, and the typical results of the MOGWO are drawn in Figure 3. It can be summarized as follows.

**Table 3.** Key parameters of the MOGWO.

Parameter	Value
Archive size	300
Population size	400
Number of iterations	15

**Note:** These parameters were adopted for the MOGWO in Experiment I.

**Table 4.** Results of algorithms for Multi-Objective Problems (MOPs) (using the Inverted Generational Distance, IGD).

Test Function	Algorithm	Ave	Std	Median	Best	Worst
ZDT1	MOPSO	0.00394	0.00165	0.00355	0.00223	0.01065
	MODA	0.00360	0.00103	0.00338	0.00227	0.00729
	MOGWO	<b>0.00243</b>	<b>0.00020</b>	<b>0.00241</b>	<b>0.00223</b>	<b>0.00366</b>
ZDT2	MOPSO	0.00633	0.01739	0.00295	<b>0.00215</b>	0.12234
	MODA	0.00380	0.00216	0.00325	0.00235	0.01682
	MOGWO	<b>0.00253</b>	<b>0.00025</b>	<b>0.00250</b>	0.00221	<b>0.00386</b>
ZDT3	MOPSO	0.00851	0.00465	0.00724	0.00379	0.03117
	MODA	0.00680	0.00431	0.00540	0.00286	0.02700
	MOGWO	<b>0.00358</b>	<b>0.00085</b>	<b>0.00337</b>	<b>0.00264</b>	<b>0.00264</b>

**Note:** Bolded numbers are the minimum values for each group. MOPSO: Multi-Objective Particle Swarm Optimization; MODA: Multi-Objective Dragonfly Algorithm.

In terms of the IGD, MOGWO showed the smallest Ave, Std, and Median values for the three test functions, and the smallest Best values for ZDT1 and ZDT3. It is worth noting that the MOPSO for ZDT2 showed the best IGD in one of the repeated experiments, but this is not enough to explain the significant advantages of MOPSO over MOGWO and MODA, because this may have been an accidental situation.

Intuitively, MOGWO appeared to have better characteristics than MODA and MOPSO in the vast majority of cases, showing a geometric improvement in performance relative to MOPSO in terms of the IGD distribution characteristics, and it was also greatly improved compared with MODA. Take ZDT2 as an example, the standard deviation of MOGWO's IGD was 0.00025, while the standard deviation of MOPSO's IGD reached 0.01739 and that of MODA's IGD also reached 0.00216. At the same time, the best IGD values of the three algorithms in ZDT2 had small differences, but the worst IGD values were very different: MOGWO's worst IGD was 0.00386, MODA's worst IGD was 0.01682, and MOPSO's worst IGD was 0.12234. These characteristics reflect the difference in stability of the three optimization algorithms.

**Remark 1.** *By comparing MOGWO, MODA, and MOPSO, MOGWO was found to show strong advantages over the control group algorithms no matter which test function was being used. Therefore, it is reasonable to apply MOGWO to the proposed model.*

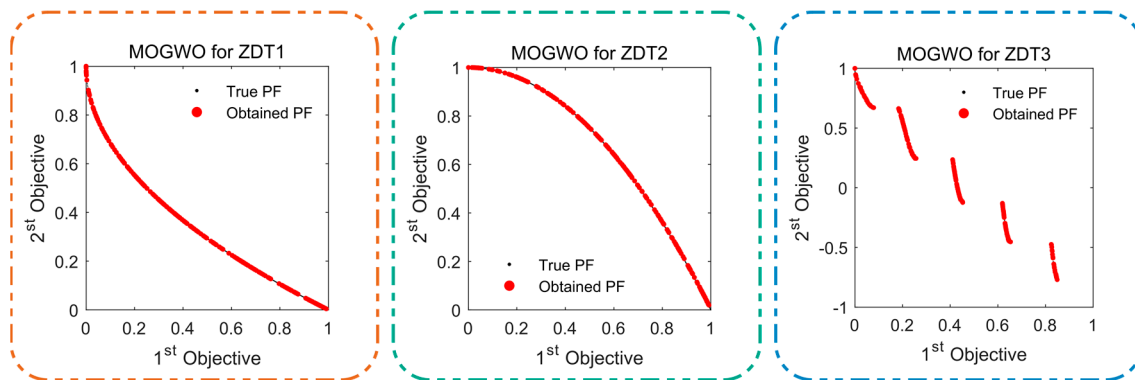


Figure 3. Obtained Pareto optimal fronts for ZDT1–3.

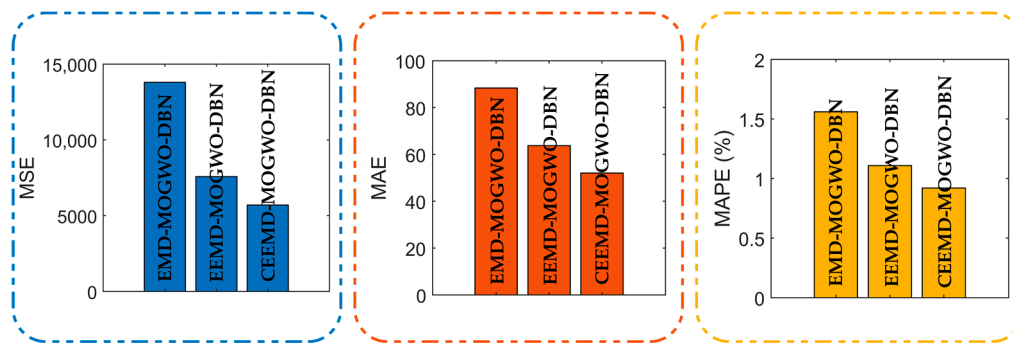
#### 5.4.2. Test of CEEMD

This part was done to prove the effectiveness and application prospect of CEEMD in time series forecasting. Since the superiority of the MOGWO had already been demonstrated, two control models were set up in this part: EMD-MOGWO-DBN and EEMD-MOGWO-DBN. Both control models were hybrid models, which were consistent with the proposed model in the overall process. They both decomposed the original data into several sub-sequences by the data preprocessing strategy and then used the DBN optimized by the MOGWO to learn and forecast each sub-sequence respectively. Finally, the forecasts were added together to output the final forecasting results. In addition, the two control models were consistent with the proposed model in terms of the construction and common parameters of the forecasting module DBN and the optimizing module MOGWO. The difference between the control models and the proposed model lay in their different data preprocessing strategies. It should be emphasized that the common parameters of these three data preprocessing strategies—EMD, EEMD, and CEEMD—were also set to be the same. For all models including control models and the proposed model CEEMD-MOGWO-DBN, Series 1–7 were employed, and the average results are presented in Table 5. In addition, Figure 4 shows the average results of the MSE, MAE, and MAPE, which are summarized below.

In the comparison of CEEMD-MOGWO-DBN and EMD-MOGWO-DBN, the former was shown to have a great advantage over the latter in terms of the forecasting accuracy. For example, on average, the MSE of CEEMD-MOGWO-DBN was only 5694.99182, while that of EMD-MOGWO-DBN was 13,796.72117. It can be inferred that CEEMD has a better preprocessing capacity than EMD.

The forecasting results of CEEMD-MOGWO-DBN were also shown to be superior to EEMD-MOGWO-DBN. It was observed that CEEMD-MOGWO-DBN had better average error metrics than EEMD-MOGWO-DBN under the condition that the running time was basically unchanged, and the parameters were the same.

**Remark 2.** In a comparison of the average performance of these models, the proposed CEEMD-MOGWO-DBN model achieved the best results among all models, regardless of the dataset. These comparisons demonstrate the superiority of CEEMD over the other two data preprocessing strategies.



**Figure 4.** The average mean square error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) results in the second part of **Experiment I**.

### 5.5. Experiment II

In this experiment, the proposed model was decomposed into one individual model (DBN) and two hybrid sub-models (CEEMD-DBN and MOGWO-DBN). The difference between these three models and the proposed model was that one or more modules were removed. The DBN model no longer had the data preprocessing and optimization modules CEEMD and MOGWO. For CEEMD-DBN, the optimization module MOGWO was eliminated, and for MOGWO-DBN, the data preprocessing module CEEMD was eliminated. It is worth noting that the remaining modules were consistent with those of the proposed model in terms of the structure and common parameters. At the same time, three comparisons were set up to explore the importance of CEEMD and MOGWO for the overall structure of the proposed model. **Comparison 1** included CEEMD-DBN, MOGWO-DBN, and DBN. Its main purpose was to explore whether the separate use of these two modules (CEEMD or MOGWO) could effectively help improve the forecasting ability of DBN. **Comparison 2** included CEEMD-DBN and MOGWO-DBN, in order to compare which module (CEEMD or MOGWO) improves the forecasting accuracy of DBN better when used alone. The purpose of **Comparison 3**, which included CEEMD-MOGWO-DBN, CEEMD-DBN, and MOGWO-DBN, was to explore whether the superposition of the two modules (CEEMD and MOGWO) could further promote the forecasting performance. The experiment was carried out based on Series 1–7, and the average results are presented in Table 5. In addition, Figure 5 shows the average results of MSE, MAE, and MAPE. It can be summarized as follows.

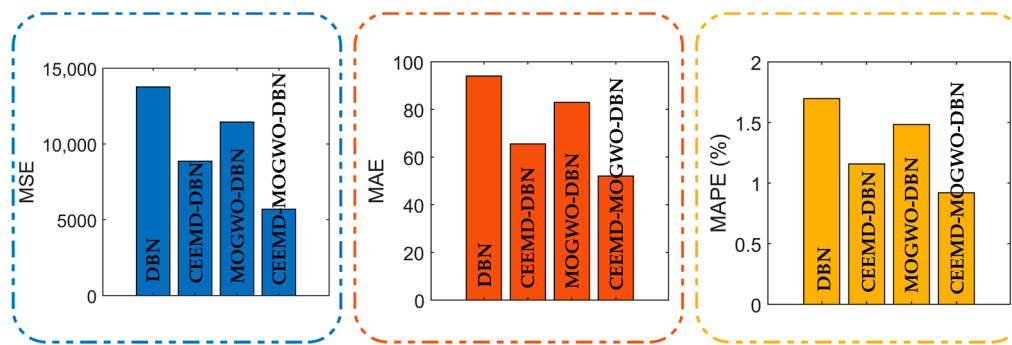
In **Comparison 1**, the performance of the two hybrid sub-models was greatly improved compared with the individual model DBN. According to the averages of the forecasting error metrics for Series 1–7, the MSE, NMSE, RMSE, MAE, MAPE, and TIC of DBN were 13,766.42486, 0.00047, 117.10453, 93.97962, 1.69669%, and 0.01015, while for CEEMD-DBN, the values of these metrics were 8865.38238, 0.00027, 91.86137, 65.54824, 1.15752%, and 0.00800, respectively, and those of MOGWO-DBN were 11,453.52281, 0.00038, 105.44077, 82.93932, 1.48279%, and 0.00916. This shows that the separate utilization of CEEMD or MOGWO is able to improve the forecasting accuracy of DBN.

In **Comparison 2**, the degree to which CEEMD contributes to the accuracy improvement of DBN was found to be deeper than that of MOGWO. Without a loss of generality, let us focus on the averages of the error metrics. The MSE, NMSE, RMSE, MAE, MAPE, and TIC of CEEMD-DBN were 2588.14043,

0.00011, 13.57940, 17.39107, 0.32527%, and 0.00116 lower than those of MOGWO-DBN in absolute values, respectively. This may be due to some limitations of DBN's ability to learn certain data features. MOGWO can only give more optimized parameters to DBN, while CEEMD enables to eliminate some data features under the limitations of DBN.

From **Comparison 3**, CEEMD-MOGWO-DBN has better accuracy compared with the two hybrid sub-models in each verification dataset. On average, the MSE, NMSE, RMSE, MAE, MAPE, and TIC of CEEMD-DBN were 8865.38238, 0.00027, 91.86137, 65.54824, 1.15752%, and 0.00800, and those of MOGWO-DBN were 11,453.52281, 0.00038, 105.44077, 82.93932, 1.48279%, and 0.00916, respectively. However, the metric values of the proposed model were 5694.99182, 0.00018, 72.47942, 52.04767, 0.91989%, and 0.00629, respectively. This seems to show that the simultaneous use of the two modules has a superposition effect on the promotion of the forecasting accuracy.

**Remark 3.** Through the comparisons above, it can be inferred that CEEMD and MOGWO are compatible with each other and have synergistic significance on the forecasting accuracy. Therefore, it is reasonable to dually utilize CEEMD and MOGWO in the proposed model.



**Figure 5.** The average results of MSE, MAE, and MAPE in **Experiment II**.

### 5.6. Experiment III

To verify the superiority of the proposed model over other time series forecasting methods, the proposed model and four representative models were included in this experiment, and Series 1–7 were used as validation datasets. The models for comparison were K-Nearest Neighbor (KNN), Support Vector Machine (SVM), MOPSO-ELM, and CEEMD-BPNN. KNN is a relatively mature statistical learning method and has been widely used in the field of multi-classification. Its main idea is to decide the category of a sample according to the category of one or several neighboring samples. SVM is an artificial intelligence method with supervised learning, which maps data features to high-dimensional space or a hyperplane to complete multi-classification tasks. In this paper, the two models were not added to other modules; they learnt and forecasted the original data directly. CEEMD-BPNN is a hybrid model composed of the data preprocessing module CEEMD and the forecasting module BPNN, while MOPSO-ELM is a hybrid model composed of the optimization module MOPSO and the forecasting module ELM. The difference between the two models is that the former first utilizes the data preprocessing strategy CEEMD to decompose the original data into sub-sequences and then uses the BPNN to learn and forecast respectively, and at last, the results are added to obtain the final output, while the latter uses the ELM optimized by MOPSO to learn and forecast the original data. In this experiment, although their structures were not identical, all comparison models used the same input variables, datasets, and common parameters as the proposed model. The average experimental results are presented in Table 5, and the results of Series 7 are drawn in Figure 6 as a typical case to reflect the forecasting ability of various models and to show more details of the forecasting results. It can be summarized as follows.

The proposed model showed an absolute advantage in terms of accuracy when compared with the KNN and SVM, representatives of the statistical and AI modeling methods. In terms of the

average values of the error metrics, the proposed model showed the leading position. Compared with CEEMD-BPNN and MOPSO-ELM, the proposed model showed a broad advancement in terms of overall performance, which was embodied by the huge reduction in average error metrics. On average, the MSE, NMSE, and MAE values of the proposed model were less than half of those of CEEMD-BPNN, and the other metrics, such as MAPE, were also less than half.

**Remark 4.** By comparing several models, this experiment showed the superiority of the proposed model over some popular models, which proves that the proposed model has great applicability and advancement in load forecasting.

**Table 5.** The average metrics of the proposed model and control models in all experiments.

Experiment	Model	MSE	NMSE	RMSE	MAE	MAPE	TIC
Experiment I	EMD-MOGWO-DBN	13,796.72117	0.00043	114.88320	88.33416	1.56016	0.00996
	EEMD-MOGWO-DBN	7576.88688	0.00022	84.81554	63.75418	1.10763	0.00741
Experiment II	DBN	13,766.42486	0.00047	117.10453	93.97962	1.69669	0.01015
	CEEMD-DBN	8865.38238	0.00027	91.86137	65.54824	1.15752	0.00800
	MOGWO-DBN	11,453.52281	0.00038	105.44077	82.93932	1.48279	0.00916
Experiment III	KNN	22,255.54597	0.00070	147.12157	107.23155	1.89466	0.01274
	SVM	33,029.01100	0.00112	177.39924	132.94256	2.41606	0.01534
	MOPSO-ELM	22,118.09268	0.00070	142.76551	101.30780	1.81040	0.01233
	CEEMD-BPNN	18,508.00399	0.00055	135.06973	104.37259	1.82192	0.01177
<b>Proposed model</b>	<b>CEEMD-MOGWO-DBN</b>	<b>5694.99182</b>	<b>0.00018</b>	<b>72.47942</b>	<b>52.04767</b>	<b>0.91989</b>	<b>0.00629</b>

**Note:** The bolded numbers are the best values. EMD: Empirical Mode Decomposition; KNN: K-Nearest Neighbor; ELM: Extreme Learning Machine; BPNN: Back Propagation Neural Network; SVM: Support Vector Machine; EEMD: Ensemble Empirical Mode Decomposition.

## 6. Discussion

In this section, six topics are discussed to further confirm the advancement of the proposed model. The topics are the significance test, correlation, the performance improvement percentage, the forecasting stability, the sensitivity analysis, and the multistep ahead forecasting.

### 6.1. Diebold–Mariano Test

The DM test was used to test whether the forecasting results of the proposed model were significantly better than those of the other models for a comparison from a statistical point of view. The relevant content and significance of the DM test were introduced in Section 4.

Table 6 shows the absolute values of DM statistics between the proposed model and the other ones. From this table, it can be observed that even the minimum value was still 3.06040, which exceeds  $Z_{0.01/2} = 2.58$ . Therefore, it is 99% certain that the null hypothesis is rejected and the alternative hypothesis is accepted. In other words, the proposed model is superior to the other ones in terms of forecasting accuracy from a statistical perspective.

**Table 6.** Diebold–Mariano (DM) statistics between the proposed model CEEMD-MOGWO-DBN and the other models.

Experiment	Model	Series 1	Series 2	Series 3	Series 4	Series 5	Series 6	Series 7
Experiment I	EMD-MOGWO-DBN	10.46140	14.02087	10.90672	14.37482	13.51899	16.75350	6.73104
	EEMD-MOGWO-DBN	3.63850	<b>3.06040</b>	4.90431	6.03657	4.66065	10.29117	7.83641
Experiment II	DBN	15.07307	13.26470	6.88396	9.40370	20.17246	16.12552	13.04906
	CEEMD-DBN	3.24853	7.90900	9.22992	3.24684	14.55746	12.15546	6.47644
	MOGWO-DBN	9.34683	11.00909	11.06282	5.64711	15.56736	15.09124	7.22580
Experiment III	KNN	15.34997	12.77729	16.14745	14.48349	18.09240	15.47069	13.33410
	SVM	17.98578	14.01903	19.49142	15.38799	24.00031	21.95435	18.92123
	MOPSO-ELM	18.06414	15.14261	7.38825	8.31480	18.62784	14.08182	10.57706
	CEEMD-BPNN	15.39158	10.52648	9.93735	14.15706	19.55250	22.90858	15.69709

**Note:** The number in bold is the minimum of all results.

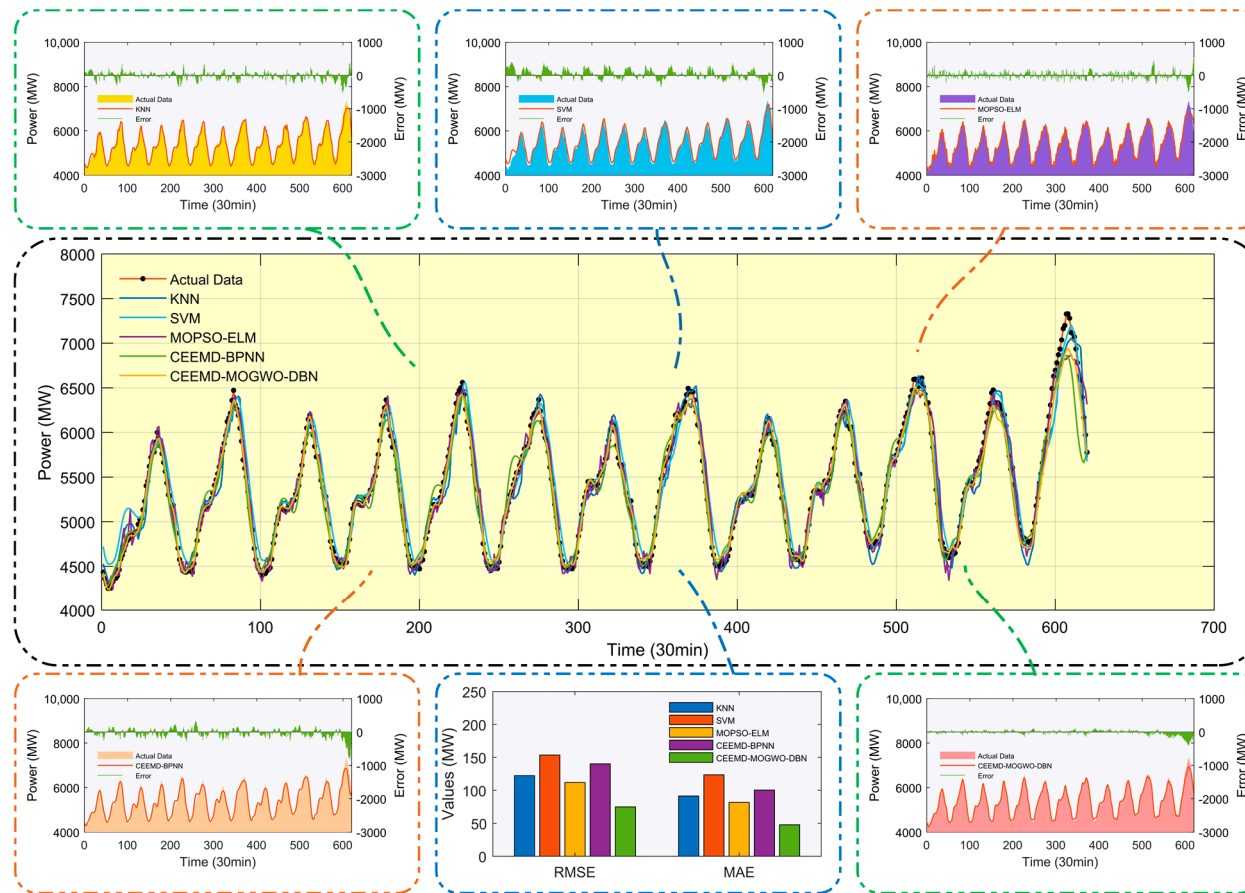


Figure 6. The forecasting results of Experiment III for Series 7.



## 6.2. Correlation

The Pearson correlation coefficient [67] was utilized to measure the degree of linear correlation between the forecasting values of a model and the real data. In the practical application of this paper, the Pearson correlation coefficient should be between 0 and 1, and the closer it is to 1, the better the performance is. The calculated results of the Pearson correlation coefficient are presented in Table 7.

**Table 7.** The results of the Pearson correlation coefficient.

Experiment	Model	Series 1	Series 2	Series 3	Series 4	Series 5	Series 6	Series 7
Experiment I	EMD-MOGWO-DBN	0.99306	0.98521	0.98368	0.97683	0.99060	0.99103	0.99117
	EEMD-MOGWO-DBN	0.99600	0.99443	0.98903	0.99343	0.99699	0.99473	0.99149
Experiment II	DBN	0.99217	0.98608	0.98908	0.99037	0.98726	0.98533	0.98895
	CEEMD-DBN	0.99588	0.99269	0.98878	0.99274	0.99743	0.99446	0.98833
	MOGWO-DBN	0.99221	0.99013	0.98414	0.99089	0.98907	0.98605	0.99145
Experiment III	KNN	0.98090	0.98213	0.97215	0.97019	0.97481	0.97709	0.98273
	SVM	0.97697	0.98582	0.96510	0.95575	0.96929	0.96580	0.97660
	MOPSO-ELM	0.97914	0.98154	0.98281	0.95133	0.98219	0.98347	0.98541
	CEEMD-BPNN	0.98532	0.98712	0.98708	0.97606	0.98690	0.99002	0.98188
Proposed model	CEEMD-MOGWO-DBN	<b>0.99657</b>	<b>0.99604</b>	<b>0.98943</b>	<b>0.99436</b>	<b>0.99794</b>	<b>0.99623</b>	<b>0.99529</b>

**Note:** The bolded numbers are the best values.

The proposed model was observed to have the largest Pearson correlation coefficient among all models in each series. This is another statistical demonstration that the proposed model performs better than the other ones in terms of the forecasting accuracy.

## 6.3. Performance Improvement Percentage

It is not sufficient to only focus on the absolute difference in forecasting error metrics between the two models when making a comparison. In many cases, it is necessary to know the relative difference. Therefore, the degree to which the proposed model improves its forecasting performance relative to the other models was explored.

The performance improvement percentage is defined as

$$P_m = \frac{m_c - m_p}{m_c} \% \quad (34)$$

where  $m_p$  refers to a kind of error metric of the proposed model, and  $m_c$  represents that of a model for comparison. Table 8 shows the performance improvement percentage of the MOGWO on the IGD compared with the other two algorithms in the former part of **Experiment I**. Table 9 shows the improvement percentage of the average performance of the proposed model on various error metrics compared with the other models in the latter part of **Experiment I** and **Experiments II–III**.

**Table 8.** Performance improvement percentage of the MOGWO.

Test Function	Algorithm	Ave	Std	Median	Best	Worst
ZDT1	MOPSO	38.48016	87.54864	31.99470	0.24381	65.63110
	MODA	32.59113	80.18044	28.53139	1.78107	49.77998
ZDT2	MOPSO	59.99774	<b>98.58045</b>	15.09655	-2.79615	96.84418
	MODA	33.35244	88.56190	23.01966	5.91433	77.04594
ZDT3	MOPSO	57.90226	81.81319	53.39034	30.39559	91.53512
	MODA	47.32379	80.35221	37.50898	7.73172	90.22715

**Note:** The number in bold is the best value of all results.

**Table 9.** The improvement percentage of the average performance of the proposed model.

Experiment	Model	MSE	NMSE	RMSE	MAE	MAPE	TIC
Experiment I	EMD-MOGWO-DBN	53.11040	53.60657	33.16819	40.69843	40.83937	32.83899
	EEMD-MOGWO-DBN	7.55020	4.76190	6.00222	15.22625	14.13953	6.03933
Experiment II	DBN	58.63129	61.18470	38.10708	44.61813	45.78336	38.01846
	CEEMD-DBN	35.76146	33.32711	21.09914	20.59640	20.52948	21.32941
	MOGWO-DBN	50.27738	52.48933	31.26054	37.24609	37.96232	31.27941
Experiment III	KNN	74.41091	74.22235	50.73502	51.46236	51.44836	50.60330
	SVM	82.75761	<b>83.90976</b>	59.14333	60.84951	61.92612	58.98592
	MOPSO-ELM	74.25189	74.14468	49.23184	48.62423	49.18862	48.95960
	CEEMD-BPNN	69.22957	67.27161	46.33926	50.13282	49.51004	46.54853

**Note:** The number in bold is the best value of all results.

The following facts can be found.

For MOPSO and MODA, MOGWO's performance was shown to be greatly improved. This was not only reflected in the improvement in the IGD by over 30% on average, but also in the improvement in IGD's standard deviation of over 80% when compared with the other two algorithms in repeated experiments, indicating that MOGWO has a stronger and more stable optimization ability.

On the condition that the running time is basically the same, the performance of the proposed model was shown to be much better than that of EEMD-MOGWO-DBN in various error metrics, on average, especially for MAE, which improved by 15.22625%. Compared with EMD-MOGWO-DBN, its average performance improved more, and the highest improvement occurred in NMSE by 53.60657%.

The combined use of CEEMD and MOGWO made the proposed model perform very well. For example, compared with DBN, CEEMD-DBN, and MOGWO-DBN, the average MSE of the proposed model improved greatly by 58.63129%, 35.76146%, and 50.27738%, respectively, through the superposition of CEEMD and MOGWO.

For the traditional time series modeling methods adopted in experiments, the proposed model improved the forecasting performance to a great extent. Compared with individual forecasting models such as KNN and SVM, the improvement of average values of some metrics even reached over 70%. For example, the average MSE of the proposed model was 74.41091% better than that of KNN and 82.75761% better than that of SVM. In addition, compared with hybrid models composed of classical neural network structures, including MOPSO-ELM and CEEMD-BPNN, the proposed model also showed an improvement of 40%–75% in terms of average error metrics.

#### 6.4. The Forecasting Stability

In previous experiments and areas of discussion, the forecasting accuracy of models was explored from various perspectives, while here, the forecasting stability is described in detail. Usually, the forecasting stability of a model is embodied by the variance or standard deviation of the forecasting errors. Table 10 presents the standard deviation estimators of the forecasting errors of all models.

The proposed model obviously showed the minimum forecasting error standard deviation on all validation datasets. This demonstrates that, in the proposed model, both high forecasting accuracy and stability are achieved.

**Table 10.** The standard deviation estimators of the forecasting error for all forecasting models in Experiments I, II, and III.

Experiment	Model	Series 1	Series 2	Series 3	Series 4	Series 5	Series 6	Series 7
Experiment I	EMD-MOGWO-DBN	88.58155	118.19889	132.46885	162.64063	92.82082	73.04270	90.12886
	EEMD-MOGWO-DBN	67.89519	73.91188	110.43459	92.40614	52.79730	56.34666	94.10260
Experiment II	DBN	94.66957	116.69391	113.71044	106.00183	107.74998	93.86037	98.91971
	CEEMD-DBN	68.27239	87.62190	114.19841	96.35408	50.90029	62.10237	102.67407
	MOGWO-DBN	93.87868	96.03359	143.22730	104.21273	100.26785	91.49436	85.83256
Experiment III	KNN	146.64712	129.78711	172.34469	184.22771	150.99963	116.38403	121.56951
	SVM	164.76477	121.49697	205.50731	242.07558	177.57617	146.08929	147.09710
	MOPSO-ELM	152.98258	130.99823	135.91668	235.82991	127.37674	99.56471	111.81620
	CEEMD-BPNN	129.63057	111.30292	127.58944	167.85435	110.38686	86.82950	134.83774
Proposed model	CEEMD-MOGWO-DBN	<b>62.79763</b>	<b>61.96000</b>	<b>109.27575</b>	<b>85.48562</b>	<b>43.53738</b>	<b>50.70754</b>	<b>72.54930</b>

Note: The bolded numbers are the best values.

### 6.5. The Sensitivity Analysis

In the proposed model, two parameters have significant effects on the performance: One is the ratio that divides the standard deviation of the added noise by that of the original data in the CEEMD strategy, and the other is the population size in the MOGWO. In this discussion, two comparisons were set up to verify whether the proposed model is robust within a certain range of these two parameters. In **Comparison A**, the ratios mentioned above in CEEMD were set as 0.3, 0.4, 0.5, 0.6, and 0.7, and the other parameters were the same as the original experiments. In **Comparison B**, the population sizes were set as 10, 15, 20, 25, and 30, and the other parameters were the same as in the original experiments. Both **Comparison A** and **Comparison B** used Series 5 as the validation data, and the results are shown in Table 11.

**Table 11.** The results of the **Comparison A** and **Comparison B**.

	Ratio	MSE	NMSE	RMSE	MAE	MAPE	TIC
<b>Comparison A</b>	0.3	5813.70992	0.00017	76.24769	56.99491	0.97718	0.00652
	0.4	2512.23620	0.00008	50.12221	40.38346	0.69894	0.00428
	0.5	<b>2115.27656</b>	<b>0.00006</b>	<b>45.99214</b>	<b>37.14969</b>	<b>0.64682</b>	<b>0.00392</b>
	0.6	2561.15508	0.00008	50.60786	39.48743	0.67939	0.00432
	0.7	3538.55532	0.00010	59.48576	47.68477	0.81947	0.00509
	Population size	MSE	NMSE	RMSE	MAE	MAPE	TIC
<b>Comparison B</b>	10	4936.40252	0.00014	70.25954	56.80433	0.97146	0.00602
	15	2373.94717	0.00007	48.72317	38.69536	0.66663	0.00416
	20	<b>2115.27656</b>	<b>0.00006</b>	<b>45.99214</b>	<b>37.14969</b>	<b>0.64682</b>	<b>0.00392</b>
	25	2227.62772	0.00007	47.19775	38.18084	0.65573	0.00403
	30	4421.73022	0.00013	66.49609	54.89949	0.93751	0.00570

Note: The bolded numbers are the best values.

From the table, it can be seen that the model's performance for **Comparison A** and **Comparison B** was similar with the change of independent variables. In other words, the MAPE of the model decreased sharply, then decreased slowly, then increased slowly and then increased sharply with the increase of the ratio in the CEEMD or the population size in the MOGWO. The variations in the other error metrics with the independent variables were basically consistent with that of MAPE, which presents like a quadratic function.

Based on these phenomena, it can be further concluded that the proposed model is stable within a certain parameter range. For example, for Series 5, the ratio should be between 0.4 and 0.6, and the population size should be between 15 and 25. Therefore, the proposed model has favorable robustness under certain conditions.

### 6.6. Multistep Ahead Forecasting

In the field of electricity load forecasting, one-step ahead forecasting may be not enough to make perfect arrangements. Therefore, a comparison of the proposed model and the other models in **Experiment III** is presented for the two-step and three-step ahead forecasting in Series 1–7. The average results are shown in Table 12.

**Table 12.** The average results of multistep ahead forecasting.

Step	Model	MSE	NMSE	RMSE	MAE	MAPE	TIC
Two-step ahead	KNN	34,215.81659	0.00106	182.90041	133.19721	2.34632	0.01583
	SVM	48,941.18630	0.00161	217.26204	166.79255	3.01063	0.01878
	MOPSO-ELM	75,034.30937	0.00314	266.45001	194.04423	3.51012	0.02304
	CEEMD-BPNN	232,584.77000	0.00724	472.55533	394.08214	6.94441	0.04137
	CEEMD-MOGWO-DBN	<b>11,084.71210</b>	<b>0.00033</b>	<b>102.05078</b>	<b>71.77219</b>	<b>1.25907</b>	<b>0.00891</b>
Three-step ahead	KNN	46,592.92144	0.00141	212.64385	153.71207	2.70164	0.01838
	SVM	68,994.14511	0.00222	258.68846	201.81107	3.62887	0.02235
	MOPSO-ELM	279,178.53100	0.00597	484.89834	317.42655	5.83106	0.04178
	CEEMD-BPNN	274,851.16010	0.00912	508.09217	413.05802	7.42217	0.04430
	CEEMD-MOGWO-DBN	<b>17,417.97342</b>	<b>0.00052</b>	<b>128.48066</b>	<b>91.23061</b>	<b>1.59609</b>	<b>0.01123</b>

**Note:** The bolded numbers are the best values.

By comparing Tables 5 and 12, it can be found that as the steps increased, almost all models showed a certain degree of increase in the forecasting error metrics. Even so, the minimum value of the average MAPE of the models for comparison in the two-step ahead forecasting reached 2.34632% and that of the models for comparison in the three-step ahead forecasting reached 2.70164%. On the contrary, during multistep ahead forecasting, the proposed model obtained the minimum average values of all the error metrics for the seven datasets. For the two-step ahead forecasting, the average MAPE of the proposed model reached a satisfying value of 1.25907%. And for the three-step ahead forecasting, the average MAPE of the proposed model was 1.59609%. In addition, the proposed model showed multiple reductions in other error metrics relative to the other models in multistep ahead forecasting.

Based on the above facts, it can be inferred that the proposed model can be effectively utilized in the multistep ahead forecasting of electricity load series. Therefore, it is reasonable to conclude that the proposed model can better learn the characteristics of data than models composed by other structures due to its special structure and principles, so it can often achieve excellent forecasting performance.

## 7. Conclusions

To meet the special requirements of load forecasting, an excellent hybrid model was proposed in this paper that integrates an advanced data preprocessing strategy, a powerful multi-objective optimization algorithm, and a cutting-edge deep neural network. Among them, the CEEMD disassembles the original data into IMF sequences, the DBN is used for data learning and forecasting, and the MOGWO optimizes the initial parameters of the DBN to improve the forecasting accuracy and stability simultaneously. In addition, reasonable experiments, multiple metrics, and areas of discussion were adopted to comprehensively verify the model's forecasting performance.

According to the experiments and discussion, the advancement of the proposed model can be summarized as follows:

- (1) The CEEMD can effectively remove the high-frequency noise in the data, thus improving the forecasting performance markedly.
- (2) Deep neural networks such as the DBN have better data learning and forecasting capabilities than models composed of other simple structures.
- (3) The MOGWO has a powerful ability to search the Pareto optimal fronts of MOPs, which simultaneously improves the forecasting accuracy and stability of the proposed model.

(4) The superposition of the three modules makes the proposed model form a complex and powerful hybrid forecasting system, which utilizes the advantages of each module at the same time and achieves great forecasting performance.

Overall, this paper contributes a novel and practical hybrid model to the field of time series forecasting of electricity load. In addition, based on the model's excellent performance in modeling nonlinear and non-stationary electricity load series, there are reasonable grounds to believe that the proposed model may be competent for use in wind power forecasting, traffic flow forecasting, solar radiation forecasting, temperature forecasting, stock price forecasting, and forecasting works in other fields.

### Abbreviation

AR	Auto Regressive
ARMA	Auto Regressive Moving Average
ARIMA	Auto Regressive Integrated Moving Average
SM	Seasonal Model
GM	Grey Model
LR	Linear Regression
AI	Artificial Intelligence
ANN	Artificial Neural Network
SOM	Self-Organizing Map
ANFIS	Adaptive Network based Fuzzy Inference System
BNN	Bayesian Neural Network
SVR	Support Vector Regression
SVM	Support Vector Machine
GOA	Grasshopper Optimization Algorithm
ELM	Extreme Learning Machine
AWNN	Advanced Wavelet Neural Network
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
EA	Evolutionary Algorithm
FA	Firefly Algorithm
CSA	Cuckoo Search Algorithm
EMD	Empirical Mode Decomposition
WT	Wavelet Transform
SSA	Singular Spectral Analysis
MOP	Multi-Objective Problem
MOPSO	Multi-Objective Particle Swarm Optimization
NSGA-II	Non-dominated Sorting Genetic Algorithm-II
MOWOA	Multi-Objective Whale Optimization Algorithm
MOEA	Multi-Objective Evolutionary Algorithm
CNN	Convolutional Neural Network
DBN	Deep Belief Network
LSTM	Long Short-Term Memory network
WPT	Wavelet Packet Transform
IEMD	Improved Empirical Mode Decomposition
EEMD	Ensemble Empirical Mode Decomposition
MOGWO	Multi-Objective Grey Wolf Optimizer
CEEMD	Complementary Ensemble Empirical Mode Decomposition
IMF	Intrinsic Mode Function

RBM	Restricted Boltzmann Machine
BP	Back Propagation
GWO	Grey Wolf Optimizer
DM test	Diebold-Mariano test
QL	Queensland
AU	Australia
MSE	Mean Square Error
NMSE	Normalized Mean Square Error
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
TIC	Theil's Inequality Coefficient
MODA	Multi-Objective Dragonfly Algorithm
IGD	Inverted Generational Distance
KNN	K-Nearest Neighbor
BPNN	Back Propagation Neural Network

**Author Contributions:** Conceptualization, K.N. and J.W.; Methodology, J.W.; Software, K.N.; Validation, J.W., G.T. and D.W.; Formal Analysis, K.N. and J.W.; Investigation, G.T. and D.W.; Resources, K.N.; Data Curation, G.T. and D.W.; Writing-Original Draft Preparation, K.N.; Writing-Review & Editing, K.N. and J.W.; Visualization, K.N. and J.W.; Supervision, J.W.; Project Administration, K.N.; Funding Acquisition, J.W.

**Funding:** This work was supported by the National Natural Science Foundation of China (grant number 71671029).

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

## Appendix A

**Table A1.** Six error metrics.

Metric	Definition	Equation
MSE	Mean Square Error	$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (A_i - F_i)^2$
NMSE	Normalized Mean Square Error	$\text{NMSE} = \frac{1}{N} \sum_{i=1}^N \frac{(A_i - F_i)^2}{A_i F_i}$
RMSE	Root Mean Square Error	$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (A_i - F_i)^2}$
MAE	Mean Absolute Error	$\text{MAE} = \frac{1}{N} \sum_{i=1}^N  A_i - F_i $
MAPE	Mean Absolute Percentage Error	$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left  \frac{A_i - F_i}{A_i} \right  \times 100\%$
TIC	Theil's Inequality Coefficient	$\text{TIC} = \sqrt{\frac{1}{N} \sum_{i=1}^N (A_i - F_i)^2} / \left( \sqrt{\frac{1}{N} \sum_{i=1}^N A_i^2} + \sqrt{\frac{1}{N} \sum_{i=1}^N F_i^2} \right)$



## Appendix B

Table A2. Test functions.

Name	Function	Search domain
ZDT1	$\text{Minimize} = \begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x})) \\ g(\mathbf{x}) = 1 + \frac{9}{29} \sum_{i=2}^{30} x_i \\ h(f_1(\mathbf{x}), g(\mathbf{x})) = 1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}} \end{cases}$	$\begin{aligned} 0 \leq x_i \leq 1, \\ 1 \leq i \leq 30 \end{aligned}$
ZDT2	$\text{Minimize} = \begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x})) \\ g(\mathbf{x}) = 1 + \frac{9}{29} \sum_{i=2}^{30} x_i \\ h(f_1(\mathbf{x}), g(\mathbf{x})) = 1 - \left(\frac{f_1(\mathbf{x})}{g(\mathbf{x})}\right)^2 \end{cases}$	$\begin{aligned} 0 \leq x_i \leq 1, \\ 1 \leq i \leq 30 \end{aligned}$
ZDT3	$\text{Minimize} = \begin{cases} f_1(\mathbf{x}) = x_1 \\ f_2(\mathbf{x}) = g(\mathbf{x})h(f_1(\mathbf{x}), g(\mathbf{x})) \\ g(\mathbf{x}) = 1 + \frac{9}{29} \sum_{i=2}^{30} x_i \\ h(f_1(\mathbf{x}), g(\mathbf{x})) = 1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}} - \left(\frac{f_1(\mathbf{x})}{g(\mathbf{x})}\right) \sin(10\pi f_1(\mathbf{x})) \end{cases}$	$\begin{aligned} 0 \leq x_i \leq 1, \\ 1 \leq i \leq 30 \end{aligned}$

## References

- Li, Z.; Hurn, A.S.; Clements, A.E. Forecasting quantiles of day-ahead electricity load. *Energy Econ.* **2017**, *67*, 60–71. [[CrossRef](#)]
- Bessec, M.; Fouquau, J. Short-run electricity load forecasting with combinations of stationary wavelet transforms. *Eur. J. Oper. Res.* **2018**, *264*, 149–164. [[CrossRef](#)]
- Fan, G.-F.; Peng, L.-L.; Hong, W.-C. Short term load forecasting based on phase space reconstruction algorithm and bi-square kernel regression model. *Appl. Energy* **2018**, *224*, 13–33. [[CrossRef](#)]
- Fischer, J.; Wilfert, H.-H. Updating of daily load prediction in power systems using AR-models. In *Stochastic Control*; Sinha, N.K., Telksnys, L.A., Eds.; IFAC Symposia Series; Pergamon: Oxford, UK, 1987; pp. 243–245. ISBN 978-0-08-033452-3.
- Chen, J.-F.; Wang, W.-M.; Huang, C.-M. Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting. *Electr. Power Syst. Res.* **1995**, *34*, 187–196. [[CrossRef](#)]
- Abdel-Aal, R.E.; Al-Garni, A.Z. Forecasting monthly electric energy consumption in eastern Saudi Arabia using univariate time-series analysis. *Energy* **1997**, *22*, 1059–1069. [[CrossRef](#)]
- Ahmed, S. Seasonal models of peak electric load demand. *Technol. Forecast. Soc. Chang.* **2005**, *72*, 609–622. [[CrossRef](#)]
- Li, G.-D.; Wang, C.-H.; Masuda, S.; Nagai, M. A research on short term load forecasting problem applying improved grey dynamic model. *Int. J. Electr. Power Energy Syst.* **2011**, *33*, 809–816. [[CrossRef](#)]
- Dudek, G. Pattern-based local linear regression models for short-term load forecasting. *Electr. Power Syst. Res.* **2016**, *130*, 139–147. [[CrossRef](#)]
- Alturki, F.A.; Abdennour, A. Ben Medium to Long-term Peak Load Forecasting for Riyadh City Using Artificial Neural Networks. *J. King Saud Univ.-Eng. Sci.* **2000**, *12*, 269–283.
- Carpinteiro, O.A.S.; Reis, A.J.R.; da Silva, A.P.A. A hierarchical neural model in short-term load forecasting. *Appl. Soft Comput.* **2004**, *4*, 405–412. [[CrossRef](#)]
- Ying, L.-C.; Pan, M.-C. Using adaptive network based fuzzy inference system to forecast regional electricity loads. *Energy Convers. Manag.* **2008**, *49*, 205–211. [[CrossRef](#)]
- Lauret, P.; Fock, E.; Randrianarivony, R.N.; Manicom-Ramsamy, J.-F. Bayesian neural network approach to short time load forecasting. *Energy Convers. Manag.* **2008**, *49*, 1156–1166. [[CrossRef](#)]
- Chen, Y.; Xu, P.; Chu, Y.; Li, W.; Wu, Y.; Ni, L.; Bao, Y.; Wang, K. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. *Appl. Energy* **2017**, *195*, 659–670. [[CrossRef](#)]

15. Wang, J.; Yang, W.; Du, P.; Niu, T. A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm. *Energy Convers. Manag.* **2018**, *163*, 134–150. [[CrossRef](#)]
16. Barman, M.; Choudhury, N.B.D.; Sutradhar, S. A regional hybrid GOA-SVM model based on similar day approach for short-term load forecasting in Assam, India. *Energy* **2018**, *145*, 710–720. [[CrossRef](#)]
17. Li, S.; Goel, L.; Wang, P. An ensemble approach for short-term load forecasting by extreme learning machine. *Appl. Energy* **2016**, *170*, 22–29. [[CrossRef](#)]
18. Rana, M.; Koprinska, I. Forecasting electricity load with advanced wavelet neural networks. *Neurocomputing* **2016**, *182*, 118–132. [[CrossRef](#)]
19. Liu, T.; Jin, Y.; Gao, Y. A New Hybrid Approach for Short-Term Electric Load Forecasting Applying Support Vector Machine with Ensemble Empirical Mode Decomposition and Whale Optimization. *Energies* **2019**, *12*, 1520. [[CrossRef](#)]
20. Hong, W.-C.; Fan, G.-F. Hybrid Empirical Mode Decomposition with Support Vector Regression Model for Short Term Load Forecasting. *Energies* **2019**, *12*, 1093. [[CrossRef](#)]
21. Li, M.-W.; Geng, J.; Hong, W.-C.; Zhang, Y. Hybridizing Chaotic and Quantum Mechanisms and Fruit Fly Optimization Algorithm with Least Squares Support Vector Regression Model in Electric Load Forecasting. *Energies* **2018**, *11*, 2226. [[CrossRef](#)]
22. Dong, Y.; Zhang, Z.; Hong, W.-C. A Hybrid Seasonal Mechanism with a Chaotic Cuckoo Search Algorithm with a Support Vector Regression Model for Electric Load Forecasting. *Energies* **2018**, *11*, 1009. [[CrossRef](#)]
23. Li, M.-W.; Geng, J.; Wang, S.; Hong, W.-C. Hybrid Chaotic Quantum Bat Algorithm with SVR in Electric Load Forecasting. *Energies* **2017**, *10*, 2180. [[CrossRef](#)]
24. Zhao, X.; Wang, C.; Su, J.; Wang, J. Research and application based on the swarm intelligence algorithm and artificial intelligence for wind farm decision system. *Renew. Energy* **2019**, *134*, 681–697. [[CrossRef](#)]
25. Huang, C.-J.; Kuo, P.-H. A Short-Term Wind Speed Forecasting Model by Using Artificial Neural Networks with Stochastic Optimization for Renewable Energy Systems. *Energies* **2018**, *11*, 2777. [[CrossRef](#)]
26. Wang, J.; Li, H.; Lu, H. Application of a novel early warning system based on fuzzy time series in urban air quality forecasting in China. *Appl. Soft Comput.* **2018**, *71*, 783–799. [[CrossRef](#)]
27. Jiang, H. Sparse estimation based on square root nonconvex optimization in high-dimensional data. *Neurocomputing* **2018**, *282*, 122–135. [[CrossRef](#)]
28. Jiang, H. Model forecasting based on two-stage feature selection procedure using orthogonal greedy algorithm. *Appl. Soft Comput.* **2018**, *63*, 110–123. [[CrossRef](#)]
29. Bates, J.M.; Granger, C.W.J. The Combination of Forecasts. *J. Oper. Res. Soc.* **1969**, *20*, 451–468. [[CrossRef](#)]
30. Yang, Z.; Wang, J. A combination forecasting approach applied in multistep wind speed forecasting based on a data processing strategy and an optimized artificial intelligence algorithm. *Appl. Energy* **2018**, *230*, 1108–1125. [[CrossRef](#)]
31. Niu, X.; Wang, J. A combined model based on data preprocessing strategy and multi-objective optimization algorithm for short-term wind speed forecasting. *Appl. Energy* **2019**, *241*, 519–539. [[CrossRef](#)]
32. Shen, S.; Li, G.; Song, H. Combination forecasts of International tourism demand. *Ann. Tour. Res.* **2011**, *38*, 72–89. [[CrossRef](#)]
33. Jiang, P.; Yang, H.; Ma, X. Coal production and consumption analysis, and forecasting of related carbon emission: Evidence from China. *Carbon Manag.* **2019**, *10*, 189–208. [[CrossRef](#)]
34. Xiao, L.; Wang, J.; Hou, R.; Wu, J. A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting. *Energy* **2015**, *82*, 524–549. [[CrossRef](#)]
35. Wang, J.; Heng, J.; Xiao, L.; Wang, C. Research and application of a combined model based on multi-objective optimization for multi-step ahead wind speed forecasting. *Energy* **2017**, *125*, 591–613. [[CrossRef](#)]
36. Wang, J.; Yang, W.; Du, P.; Li, Y. Research and application of a hybrid forecasting framework based on multi-objective optimization for electrical power system. *Energy* **2018**, *148*, 59–78. [[CrossRef](#)]
37. Pian, Z.; Li, S.; Zhang, H.; Zhang, N. The Application of the Pso Based BP Network in Short-Term Load Forecasting. *Phys. Procedia* **2012**, *24*, 626–632.
38. Ghayekhloo, M.; Menhaj, M.B.; Ghofrani, M. A hybrid short-term load forecasting with a new data preprocessing framework. *Electr. Power Syst. Res.* **2015**, *119*, 138–148. [[CrossRef](#)]
39. Wang, B.; Tai, N.; Zhai, H.; Ye, J.; Zhu, J.; Qi, L. A new ARMAX model based on evolutionary algorithm and particle swarm optimization for short-term load forecasting. *Electr. Power Syst. Res.* **2008**, *78*, 1679–1685. [[CrossRef](#)]

40. Kavousi-Fard, A.; Samet, H.; Marzbani, F. A new hybrid Modified Firefly Algorithm and Support Vector Regression model for accurate Short Term Load Forecasting. *Expert Syst. Appl.* **2014**, *41*, 6047–6056. [[CrossRef](#)]
41. Xiao, L.; Shao, W.; Yu, M.; Ma, J.; Jin, C. Research and application of a hybrid wavelet neural network model with the improved cuckoo search algorithm for electrical power system forecasting. *Appl. Energy* **2017**, *198*, 203–222. [[CrossRef](#)]
42. Zhang, X.; Wang, J.; Zhang, K. Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by Cuckoo search algorithm. *Electr. Power Syst. Res.* **2017**, *146*, 270–285. [[CrossRef](#)]
43. Heng, J.; Wang, J.; Xiao, L.; Lu, H. Research and application of a combined model based on frequent pattern growth algorithm and multi-objective optimization for solar radiation forecasting. *Appl. Energy* **2017**, *208*, 845–866. [[CrossRef](#)]
44. An, N.; Zhao, W.; Wang, J.; Shang, D.; Zhao, E. Using multi-output feedforward neural network with empirical mode decomposition based signal filtering for electricity demand forecasting. *Energy* **2013**, *49*, 279–288. [[CrossRef](#)]
45. Li, R.; Jin, Y. The early-warning system based on hybrid optimization algorithm and fuzzy synthetic evaluation model. *Inf. Sci.* **2018**, *435*, 296–319. [[CrossRef](#)]
46. Gao, X.; Li, X.; Zhao, B.; Ji, W.; Jing, X.; He, Y. Short-Term Electricity Load Forecasting Model Based on EMD-GRU with Feature Selection. *Energies* **2019**, *12*, 1140. [[CrossRef](#)]
47. Li, S.; Wang, P.; Goel, L. Short-term load forecasting by wavelet transform and evolutionary extreme learning machine. *Electr. Power Syst. Res.* **2015**, *122*, 96–103. [[CrossRef](#)]
48. Li, W.; Kong, D.; Wu, J. A Novel Hybrid Model Based on Extreme Learning Machine, k-Nearest Neighbor Regression and Wavelet Denoising Applied to Short-Term Electric Load Forecasting. *Energies* **2017**, *10*, 694.
49. Afshar, K.; Bigdeli, N. Data analysis and short term load forecasting in Iran electricity market using singular spectral analysis (SSA). *Energy* **2011**, *36*, 2620–2627. [[CrossRef](#)]
50. Borhanazad, H.; Mekhilef, S.; Ganapathy, V.G.; Modiri-Delshad, M.; Mirtaheri, A. Optimization of micro-grid system using MOPSO. *Renew. Energy* **2014**, *71*, 295–306. [[CrossRef](#)]
51. Alikar, N.; Mousavi, S.M.; Ghazilla, R.A.R.; Tavana, M.; Olugu, E.U. Application of the NSGA-II algorithm to a multi-period inventory-redundancy allocation problem in a series-parallel system. *Reliab. Eng. Syst. Saf.* **2017**, *160*, 1–10. [[CrossRef](#)]
52. Wang, J.; Du, P.; Niu, T.; Yang, W. A novel hybrid system based on a new proposed algorithm—Multi-Objective Whale Optimization Algorithm for wind speed forecasting. *Appl. Energy* **2017**, *208*, 344–360. [[CrossRef](#)]
53. Péres, M.; Ruiz, G.; Nesmachnow, S.; Olivera, A.C. Multiobjective evolutionary optimization of traffic flow and pollution in Montevideo, Uruguay. *Appl. Soft Comput.* **2018**, *70*, 472–485. [[CrossRef](#)]
54. Li, J.; Zhang, D.; Zhang, J.; Zhang, J.; Li, T.; Xia, Y.; Yan, Q.; Xun, L. Facial Expression Recognition with Faster R-CNN. *Procedia Comput. Sci.* **2017**, *107*, 135–140. [[CrossRef](#)]
55. Khatami, A.; Khosravi, A.; Nguyen, T.; Lim, C.P.; Nahavandi, S. Medical image analysis using wavelet transform and deep belief networks. *Expert Syst. Appl.* **2017**, *86*, 190–198. [[CrossRef](#)]
56. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [[CrossRef](#)]
57. He, W. Load Forecasting via Deep Neural Networks. *Procedia Comput. Sci.* **2017**, *122*, 308–314. [[CrossRef](#)]
58. Laouafi, A.; Mordjaoui, M.; Laouafi, F.; Boukelia, T.E. Daily peak electricity demand forecasting based on an adaptive hybrid two-stage methodology. *Int. J. Electr. Power Energy Syst.* **2016**, *77*, 136–144. [[CrossRef](#)]
59. Zhang, J.; Wei, Y.-M.; Li, D.; Tan, Z.; Zhou, J. Short term electricity load forecasting using a hybrid model. *Energy* **2018**, *158*, 774–781. [[CrossRef](#)]
60. Li, W.-Q.; Chang, L. A combination model with variable weight optimization for short-term electrical load forecasting. *Energy* **2018**, *164*, 575–593. [[CrossRef](#)]
61. Yeh, J.-R.; Shieh, J.-S.; Huang, N.E. Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Adv. Adapt. Data Anal.* **2010**, *02*, 135–156. [[CrossRef](#)]
62. Wu, Z.; Huang, N.E. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. Adapt. Data Anal.* **2009**, *1*, 1–41. [[CrossRef](#)]
63. Hinton, G.E. Learning multiple layers of representation. *Trends Cogn. Sci.* **2007**, *11*, 428–434. [[CrossRef](#)] [[PubMed](#)]

64. Mirjalili, S.; Saremi, S.; Mirjalili, S.M.; Coelho, L.S. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Syst. Appl.* **2016**, *47*, 106–119. [[CrossRef](#)]
65. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
66. Diebold, F.X.; Mariano, R.S. Comparing Predictive Accuracy. *J. Bus. Econ. Stat.* **1995**, *13*, 253–263.
67. Pearson, K., VII. Note on regression and inheritance in the case of two parents. *Proc. R. Soc. Lond.* **1895**, *58*, 240–242.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).