

Article

Load Forecasting Based on LVMD-DBFCM Load Curve Clustering and the CNN-IVIA-BLSTM Model

Linjing Hu ¹, Jiachen Wang ^{1,*}, Zhaoze Guo ² and Tengda Zheng ¹

¹ College of Electric Power, Inner Mongolia University of Technology, Hohhot 010051, China; hulinjing666@163.com (L.H.); www_15838898917@163.com (T.Z.)

² The Sixth Research Institute of China Aerospace Science and Industry Corporation, Hohhot 010010, China; guozhaoze111@163.com

* Correspondence: wangjiachen103@163.com

Abstract: Power load forecasting plays an important role in power systems, and the accuracy of load forecasting is of vital importance to power system planning as well as economic efficiency. Power load data are nonsmooth, nonlinear time-series and “noisy” data. Traditional load forecasting has low accuracy and curves not fitting the load variation. It is not well predicted by a single forecasting model. In this paper, we propose a novel model based on the combination of data mining and deep learning to improve the prediction accuracy. First, data preprocessing is performed. Second, identification and correction of anomalous data, normalization of continuous sequences, and one-hot encoding of discrete sequences are performed. The load data are decomposed and denoised using the double decomposition modal (LVMD) strategy, the load curves are clustered using the double weighted fuzzy C-means (DBFCM) algorithm, and the typical curves obtained are used as load patterns. In addition, data feature analysis is performed. A convolutional neural network (CNN) is used to extract data features. A bidirectional long short-term memory (BLSTM) network is used for prediction, in which the number of hidden layer neurons, the number of training epochs, the learning rate, the regularization coefficient, and other relevant parameters in the BLSTM network are optimized using the influenza virus immunity optimization algorithm (IVIA). Finally, the historical data of City H from 1 January 2016 to 31 December 2018, are used for load forecasting. The experimental results show that the novel model based on LVMD-DBFCM load curve clustering combined with CNN-IVIA-BLSTM proposed in this paper has an error of only 2% for electric load forecasting.

Keywords: power load forecasting; LVMD; DBFCM; CNN; BLSTM; IVIA



Citation: Hu, L.; Wang, J.; Guo, Z.; Zheng, T. Load Forecasting Based on LVMD-DBFCM Load Curve Clustering and the CNN-IVIA-BLSTM Model. *Appl. Sci.* **2023**, *13*, 7332. <https://doi.org/10.3390/app13127332>

Academic Editor: Seung-Hoon Yoo

Received: 27 April 2023

Revised: 12 June 2023

Accepted: 15 June 2023

Published: 20 June 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As quality of life has improved, the demand for power energy has also risen, and thus, the requirements for power generation and transmission and the use of electricity have increased [1]. Power load forecasting applies machine learning methods to mine the key factors affecting load from historical data, such as weather and time data, to build load forecasting models [2]. Load forecasting is the basis for ensuring the balance of power supply and demand. Accurate forecasting results can reduce the pressure on transmission and distribution links and facilitate the optimal scheduling of power transmission links. They can effectively reduce power generation costs and improve economic and social benefits [3,4].

Traditional forecasting methods use the overall historical load numbers as their basis [5–9]. The trend extrapolation predicts future loads based on the historical trends of the predictor variables. It predicts near future loads with high accuracy, but error gradually increases with time. The regression analysis method predicts future load changes by establishing a functional relationship between variables. It has high fitting ability but has a large error in predicting future changes. Time series extrapolation (the autoregressive

moving average model) approximates a nonlinear relationship with a linear equation [10]. The grey forecasting method performs correlation analysis by identifying the degree of dissimilarity in development trends between system factors. Traditional forecasting models have a relatively simple structure and require highly accurate historical data, resulting in forecasting results with poor accuracy.

In recent decades, deep learning has made important breakthroughs in the field of artificial intelligence. In 1957, Frank Rosenblatt invented an artificial neural network to solve complex problems by means of a multilayer perceptron (MLP). Deep neural networks (DNNs) are currently a popular topic in load prediction research, and recurrent neural networks (RNNs) have advantages in memory parameter sharing and Turing completeness [11]. Long short-term memory (LSTM) networks overcome the problem that RNNs are prone to gradient disappearance/explosion, but their ability to correct errors is weak [12]. M. Schuster proposed a BLSTM network that can make full use of the hidden layer history state and has stronger robustness [13]. Yet, BLSTM networks have many parameters, and without proper optimization, the model may be overfitted or slow to train, resulting in inefficiency. The number of hidden layer neurons, the number of training epochs, the learning rate, and the regularization coefficient in the BLSTM network are optimized using the IVIA. It can improve the model's efficiency. The challenges of the machine learning algorithms to load forecasting for power systems, with complex, nonsmooth, nonlinear time-series and "noisy" data, will decrease the accuracy of the algorithm, and a single prediction model has difficulty meeting the accuracy requirements. Integrating a hybrid preprocessing method can significantly improve the prediction accuracy [14–17].

In summary, to improve the accuracy and meet the needs of practical problems, this paper proposes a new model based on data mining and deep learning. It considers not only historical load data but also weather information, date types, real-time electricity prices, etc. Historical data load, weather information, and real-time electricity prices are normalized, and the date and holiday information are expressed through One-Hot encoding. The DBFCM algorithm is applied to load curve clustering to overcome the problems of traditional C-means clustering and fuzzy C-means clustering in which data are clustered into small classes, inaccurate classification is performed, and the "uniform effect of the cluster size" degrades performance. First, the LVMD is used to decompose and denoise the input data to improve the continuity and stability of the data. Second, the DBFCM algorithm is used to perform load curve clustering. Third, feature fusion extraction is performed using CNN. In addition, the extracted feature vectors are used as the input of IVIA-BLSTM. The IVIA is proposed to optimize the parameters in the BLSTM network. Finally, this paper proposes a new load forecasting model based on the LVMD-DBFCM algorithm and CNN-IVIA-BLSTM.

The remainder of this article is organized as follows. Section 2 describes the IVIA algorithm, Section 3 presents the methodology used in this paper, Section 4 is a case study of this paper, and Section 5 concludes the paper.

2. Influenza Virus Immunity Optimization Algorithm

The IVIA is a new metaheuristic optimization algorithm that is inspired by the way influenza viruses spread in a population and the process of population immunity. The process of an individual being infected is divided into three states: uninfected, infected, and immune, and the population has herd immunity when the number of immune individuals in the population exceeds 80% of the total population. This section examines the IVIA with 23 sets of standard test functions and compares it with seven optimization algorithms proposed in recent years. They are turbulent flow of water-based optimization (TFWO) [18], golden eagle optimization (GEO) [19], the parasitism–predation algorithm (PPA) [20], the rat swarm optimizer (RSO) [21], gray wolf optimization (GWO) [22], particle swarm optimization (PSO) [23], and whale optimization algorithm (WOA) [24]. The results of the tested functions show that the IVIA converges faster with higher accuracy, optimizes the function better, and finds the optimal value of the function in fewer iterations. Then,

the IVIA is applied to the neural network BLSTM training of the power load forecasting model to optimize the number of hidden layer neurons, the number of training epochs, the learning rate and the regularization coefficient, and other relevant parameters to improve the accuracy of the forecasting model.

2.1. Biological Characteristics

The influenza virus is very contagious, spreads quickly, and has many routes of transmission [25]. The spread of the virus in the population can be achieved through droplet transmission and physical contact, and the rate of spread is related to the social distance between people [26]. The vast majority of the population can recover on their own after infection, and those who recover produce the corresponding antibodies in their bodies and have immune functions. A small number of people with underlying diseases, that is, elderly people or those with weak resistance, are at risk. When the number of immune individuals in the population exceeds 80%, the population has herd immunity, thus preventing the next round of disease transmission. A schematic diagram of the transmission process of the influenza virus in a population is shown in Figure 1.

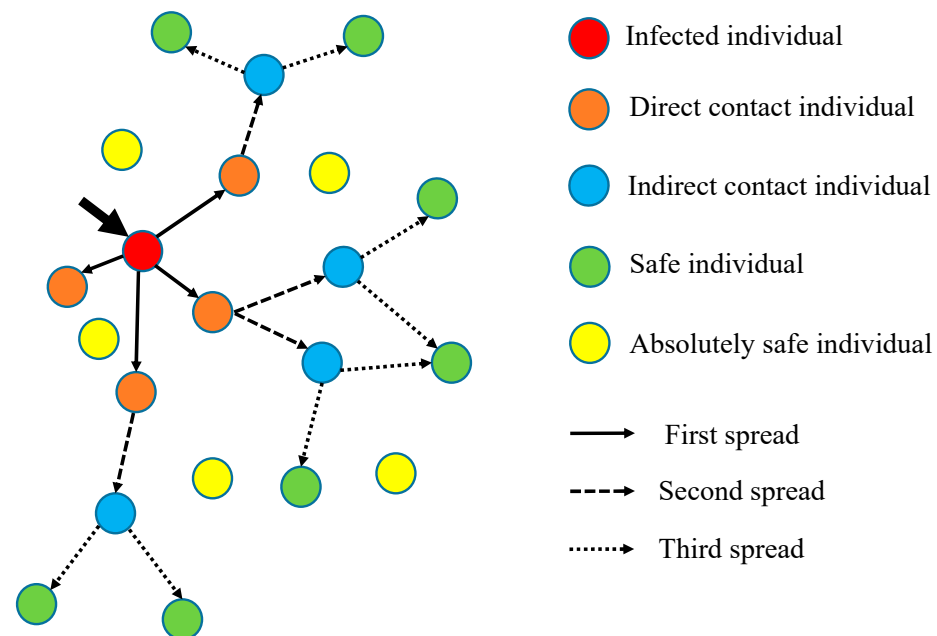


Figure 1. Diagram of the process of influenza virus population transmission.

The population is divided into infected individuals, highly susceptible individuals, susceptible individuals, safe individuals, and absolutely safe individuals.

Extremely susceptible individuals are direct contacts of infected individuals, who are socially close to infected individuals and are infected by getting influenza virus from infected individuals; susceptible individuals are indirect contacts of infected individuals who are socially distant from the infected individuals, may be infected through third parties as vectors, and have a higher risk of infection; safe individuals are not in contact with infected individuals or other individuals, and because the influenza virus is transmitted as a vector, through a change in social distance, safe individuals may become susceptible individuals or infected individuals; absolutely safe individuals have immune functions through vaccination or healing after infection and are not susceptible to infection, regardless of the contact distance of infected individuals. In this paper, the influenza virus immunization algorithm is proposed through the transmission of the influenza virus in a population and the immunization mechanism. To map the spread process of the influenza virus with the optimization algorithm, the following assumptions are made:

1. The initial number of infected individuals is small, and they are randomly distributed in the population. The category to which an individual currently belongs and the corresponding update formula are determined based on the distance between the individual and the infected individual;
2. The overall number of the population is kept constant, and absolutely safe individuals randomly move to different locations in the population. The number and location of other individuals change dynamically with the number of iterations;
3. Individuals in the population have three states: infected, uninfected, and immune. Infected individuals have immunity after infecting other individuals and will not be infected in the following process. A population is immune when 80% or more of the individuals in the population are immune;
4. Individuals are infected by contracting virus cells from an infected person, and the process of virus cell exchange is shown in Figure 2. The influenza virus cells and the diseased cells are taken as the smallest units within an individual to represent the dimension of the optimization problem.

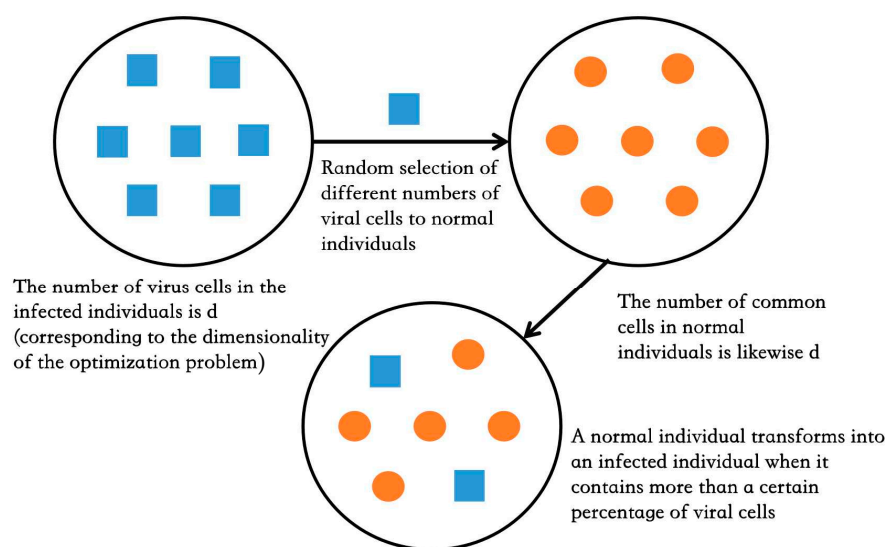


Figure 2. Diagram of the virus cell exchange process.

Immunized individuals: The individuals who are categorized as immunized are protected against the virus, and they are not affected by infected individuals.

Figure 3 shows the proportion of immune individuals to the entire population with the 23 sets of standard test functions run by IVIA. From the figure, except for functions F4, F7, F9, F16, F21, and F23, all the remaining 17 tested functions stop the iterative search process when the population immunity rate reaches 0.8. In this paper, the minimum population immunity rate for IVIA is set at 0.8, i.e., the search for optimization stops when the number of immunized individuals in the population exceeds 80% of the total. It ensures that the algorithm has high optimization results.

In this paper, we propose the IVIA based on the transmission and immunization process of the influenza virus. The influenza virus spreads rapidly in a population, and the growth curve of the cumulative number of infections is shown in Figure 4. Curve ① indicates that the number of infected individuals in the population increases rapidly within 2~3 weeks, and curves ② and ③ indicate that the number of infected individuals peaks at 7~8 weeks and then gradually starts to decline. Curve ④ indicates that the number of infected individuals gradually decreases and the number of immunized individuals increases near the end of this influenza outbreak. The IVIA has a good optimization capability at the beginning of the iteration. Figure 4 is a schematic diagram summarized according to the epidemic transmission rules; the website of the epidemic transmission rules is shown below.

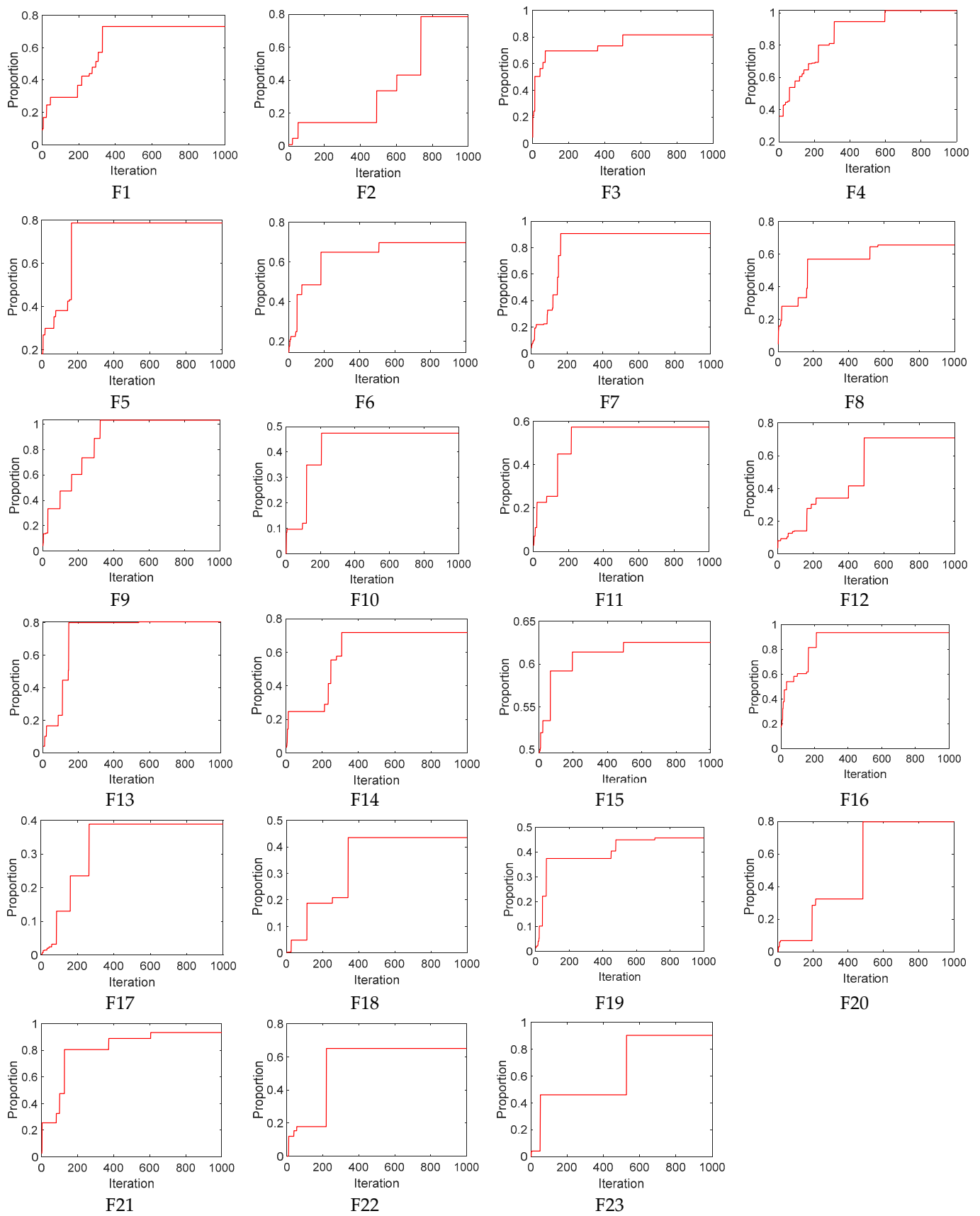


Figure 3. The proportion of immune individuals in the whole population.

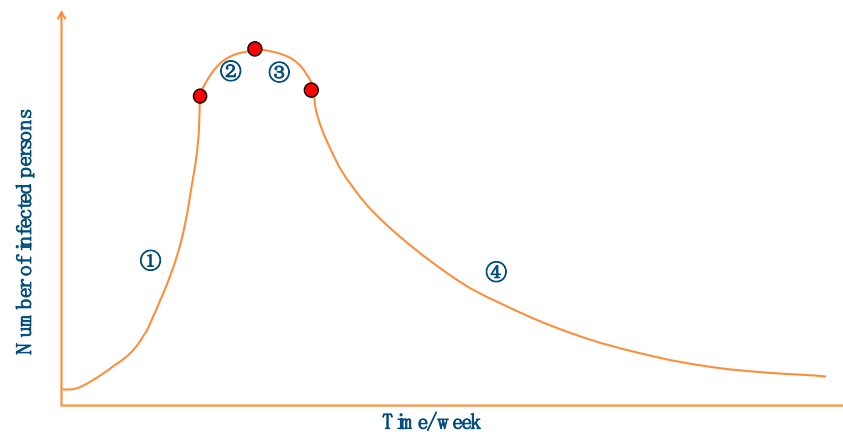


Figure 4. Growth curve of the number of infected people.

<https://www.zhzyw.com/crb/gm/08521151042985F4176E7H4KEC0B47A.html> (accessed on 12 June 2023).

2.2. Mathematical Model

The process of influenza virus transmission in a population is mapped with the influenza virus immunization algorithm. The population is represented by the matrix X , as shown in Equation (1), where n is the population size and the number of individuals, and d is the number of cells invaded by the influenza virus into the human body, corresponding to the dimensionality of the objective function. The initialization process of the algorithm randomly generates an $n \times d$ dimension matrix X , which sets the initial number of infected individuals to x .

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} \tag{1}$$

The fitness values of different individuals in the population are represented by Equation (2). Each row in the matrix is represented as the fitness value of the current individual. The individual corresponding to the optimal fitness value is the current optimal solution of the objective function.

$$F(x) = \begin{bmatrix} f(x_{11}, x_{12}, \cdots, x_{1d}) \\ f(x_{21}, x_{22}, \cdots, x_{2d}) \\ \vdots \\ f(x_{n1}, x_{n2}, \cdots, x_{nd}) \end{bmatrix} \tag{2}$$

The distance between individuals is represented by C , as shown in Equation (3). $x = \{x_1, x_2, \dots, x_d\}$ is the position of individual x , $y = \{y_1, y_2, \dots, y_d\}$ is the position of individual y , and d is the dimension of the optimization problem.

$$C = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_d - y_d)^2} \tag{3}$$

L is the maximum safety distance, as shown in Equation (4). (L_{\min}, L_{\max}) represents the range of values for individuals.

$$L = \frac{1}{2} \sqrt{L_{\min}^2 + L_{\max}^2} \tag{4}$$

The virus spreading process corresponds to the iterative optimization-seeking process in the algorithm. It is assumed that the safe distance between other individuals and the

infected person when the influenza virus spreads in the population is L . The iterative formulas for individuals in different cases are shown in Equations (5) to (9).

- Infected individuals are updated by Equation (5):

$$x_{i,j}(t + 1) = x_{i,j}(t) \left(\exp \left(\frac{N_{\max} - n}{N_{\max}} - 1 \right) \times \frac{1}{d - 1} \right)^q \tag{5}$$

When the contact distance between individuals in the population and infected individuals is $C < 0.2L$, the individuals are moved into the infected state. The position of infected individuals is updated by Equation (5). N_{\max} is the maximum number of iterations, n is the current number of iterations, d is the dimensionality of the optimization problem, and q is used to regulate the individual position update rate. The larger the value is of q , the faster the position update rate and the higher the accuracy of the optimization search.

- Extremely susceptible individuals are updated by Equation (6):

$$\begin{aligned} x_{i,j}(t + 1) &= x_{i,j}(t) + \alpha |x_{i,j}(t) - x_{i,q}(t)| \\ \alpha &= \frac{1}{d} \sum_{p=1}^d \text{rand}(-1, 1) \end{aligned} \tag{6}$$

When the contact distance between individuals in the population and infected individuals is $0.2L < C < 0.5L$, the position is updated by Equation (6). $x_{i,q}(t)$ is the random acquisition of viral cells by highly susceptible individuals; α represents update coefficient at the contact distance of $0.2L < C < 0.5L$. When the number of exchanged cells exceeds a quarter of the total number of cells, the highly susceptible individuals are transformed into infected individuals. Cell exchange occurs only once during each iteration, and the number of exchanges is random.

- Susceptible individuals are updated by Equation (7):

$$\begin{aligned} x_{i,j}(t + 1) &= x_{i,j}(t) + \beta |x_{i,j}(t) - x_{i,v}(t)| \\ \beta &= \exp \left(\frac{C - L}{\varepsilon \times N_{\max}} \right) \\ \varepsilon &= \text{rand}(0, 1) \end{aligned} \tag{7}$$

When the contact distance between individuals in the population and infected individuals is $0.5L < C < 0.8L$, the position is updated by Equation (7). $x_{i,v}(t)$ is a random acquisition of viral cells by highly susceptible individuals, β represents update coefficient at the contact distance of $0.5L < C < 0.8L$, ε is a random number in the range 0~1. When the number of exchanged cells has exceeded one-third of the total number of cells, the highly susceptible individuals are transformed into infected individuals.

- Secure individuals updated by Equation (8):

$$\begin{aligned} x_{i,j}(t + 1) &= x_{i,j}(t) + \gamma |x_{i,j}(t) - x_{i,w}(t)| \\ f(x_{i,w}) &= \text{argmin}f(x_{i,j}) \\ \gamma &= \text{rand}(-1, 1) \end{aligned} \tag{8}$$

When the contact distance between individuals in the population and infected individuals is $0.8L < C < L$, the position is updated by Equation (8). γ is a random number in the range $-1 \sim 1$. There is no cellular exchange between safe individuals and infected individuals. However, as the number of infected individuals increases, the contact distance between safe individuals and infected individuals may decrease, and safe individuals may become susceptible.

- Absolutely safe individuals updated by Equation (9):

$$\begin{aligned}x_{i,j}(t+1) &= \lambda x_{i,j}(t) \\ \lambda &= \gamma \times \left(\frac{\sqrt{2}-1}{2}\right)L\end{aligned}\quad (9)$$

Absolutely safe individuals are immune, and the contact distance between individuals and infected individuals is $C > L$; λ represents the update coefficient at the contact distance of $C > L$. They randomly change their positions during the iterative process and can be used to escape the current search region when the algorithm falls into local optima.

Stop criterion IVIA: The algorithm terminates when the number of immune individuals in the population exceeds more than 80% of the population or when the maximum number of iterations is reached. Therefore, the individual with the highest population immunity corresponds to the optimal solution of the optimization problem, and the corresponding function value is the optimal value of the optimization problem. If the problem to be solved requires higher accuracy, the population immunity rate in the Algorithm 1 can be increased. The pseudo-code of IVIA is presented below.

Algorithm 1 IVIA pseudo-code

Input:

n : The number of people

N_{\max} : The maximum number of iterations

m : Initial number of infected individuals (usually set to 1)

L : Maximum safe contact distance

R : Herd immunity ratio

C : The distance between individuals

lb, ub : Search boundary

1: Population initialization

2: Setting parameters

3: **while** ($t < N$)

4: Calculate fitness value and sort

5: **for** $i = 1:n$

6: **if** ($C < 0.2L$) then

7: Using Equation (5) to update the location of the infected individuals

8: Record the current status of the individuals

9: **else if** ($0.2L < C < 0.5L$) then

10: Using Equation (6) to update the location of highly susceptible individuals

11: Record the current status of the individuals

12: **else if** ($0.5L < C < 0.8L$) then

13: Using Equation (7) to update the location of susceptible individuals

14: Record the current status of the individuals

15: **else if** ($0.8L < C < L$) then

16: Using Formula (8) to update the position of a safe individuals

17: Record the current status of the individual

18: **else**

19: Using Equation (9) to update the position of an absolutely safe individuals

20: **end if**

21: Recalculate fitness value

22: **end for**

23: Calculate immunity rate

24: $t = t + 1$

25: **end while**

2.3. Algorithm Testing

To test the actual optimization effect of the proposed IVIA, it was evaluated in this paper on 23 sets of standard test functions. The optimization results were compared with those of turbulent flow of water-based optimization (TFWO), golden eagle optimization (GEO), the parasitism–predation algorithm (PPA), the rat swarm optimizer (RSO), gray

wolf optimization (GWO), particle swarm optimization (PSO), and the whale optimization algorithm (WOA).

Table 1 shows the expression of the test function, the dimension of the optimization problem, and the overall search range, and the last column shows the minimum value of the test function expectation, which is the theoretical optimum. The IVIA population size n is 30, the maximum number of iterations N_{max} is 1000, the maximum safe contact distance L is $0.5ub$, and the population immunity rate R is 80% during the test period. The number of dimensions of the test function and the parameters, such as search intervals lb and ub , are set according to Table 1. The overall and iteration numbers of the other optimization functions are the same as those of the IVIA.

Table 1. Standard test function.

Standard	Dimension	Search Space	Minimum
$F_1(x) = \sum_{i=1}^n x_i^2$	30	[−100, 100]	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[−10, 10]	0
$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^n x_j \right)^2$	30	[−100, 100]	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30	[−100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2) + (x_i - 1)^2]$	30	[−30, 30]	0
$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[−100, 100]	0
$F_7(x) = \sum_{i=1}^n ix_i^4 + random[0, 1]$	30	[−128, 128]	0
$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[−500, 500]	−12,569.5
$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[−5.12, 5.12]	0
$F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	[−32, 32]	0
$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600, 600]	0
$F_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[−50, 50]	0
$F_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_i) + \sum_{i=1}^n \frac{(x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]}{(x_n - 1)^2 [1 + \sin^2(2\pi x_n)]} \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[−50, 50]	0
$F_{14}(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$	2	[−65.56, 65.56]	1

Table 1. Cont.

Standard	Dimension	Search Space	Minimum
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[−5, 5]	0.0003075
$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[−5, 5]	0.398
$F_{18}(x) = \begin{bmatrix} 1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \\ 30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \end{bmatrix} \times$	2	[−2, 2]	3
$F_{19}(x) = -\sum_{i=1}^4 c_i \exp \left(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2 \right)$	3	[1, 3]	−3.86
$F_{20}(x) = -\sum_{i=1}^4 c_i \exp \left(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2 \right)$	6	[0, 1]	−3.32
$F_{21}(x) = -\sum_{i=1}^{10} \left(\sum_{j=1}^5 (x_j - C_{ji})^2 + \beta_i \right)^{-1}$ $\beta = \frac{1}{10}(1, 2, 2, 4, 4, 6, 3, 7, 5, 5)$ $C = \begin{bmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{bmatrix}$	4	[0, 10]	−10.1532
$F_{22}(x) = -\sum_{i=1}^7 \left(\sum_{j=1}^5 (x_j - C_{ji})^2 + \beta_i \right)^{-1}$	4	[0, 10]	−10.4028
$F_{23}(x) = -\sum_{i=1}^{10} \left(\sum_{j=1}^4 (x_j - C_{ji})^2 + \beta_i \right)^{-1}$	4	[0, 10]	−10.5364

The viability of the proposed IVIA is tested using 23 well-known benchmark functions with different sizes and complexity. Figure 5 shows the three-dimensional graphs of the standard test functions and their corresponding test results. The convergence of the IVIA is very good for both single-peaked and multi-peaked functions.

From the test results in Table 2, the accuracy of the IVIA for function finding is much higher than that of several other optimization algorithms, and the theoretical optimal value can be found for some functions. IVIA has fewer iterations, a shorter running time, and faster convergence compared to the other seven algorithms.

For comparative evaluation, the proposed IVIA is compared against seven well-established comparative methods using the same benchmark functions. IVIA optimizes best in F1, F2, F3, F4, F5, F6, F7, F8, converges fastest in F11, F12, F13, F14, F15, F17, and it has the same effect as other algorithms in F18, F19, F20, F21, F22, F23. The comparative results show the effectiveness of IVIA.

In summary, the IVIA is based on the characteristics of fast convergence, better robustness, and good optimization search. In this paper, the IVIA is selected to optimize the relevant parameters, such as the number of neurons in the hidden layer, the number of training times, the learning rate, and the regularization coefficients in the BLSTM network.

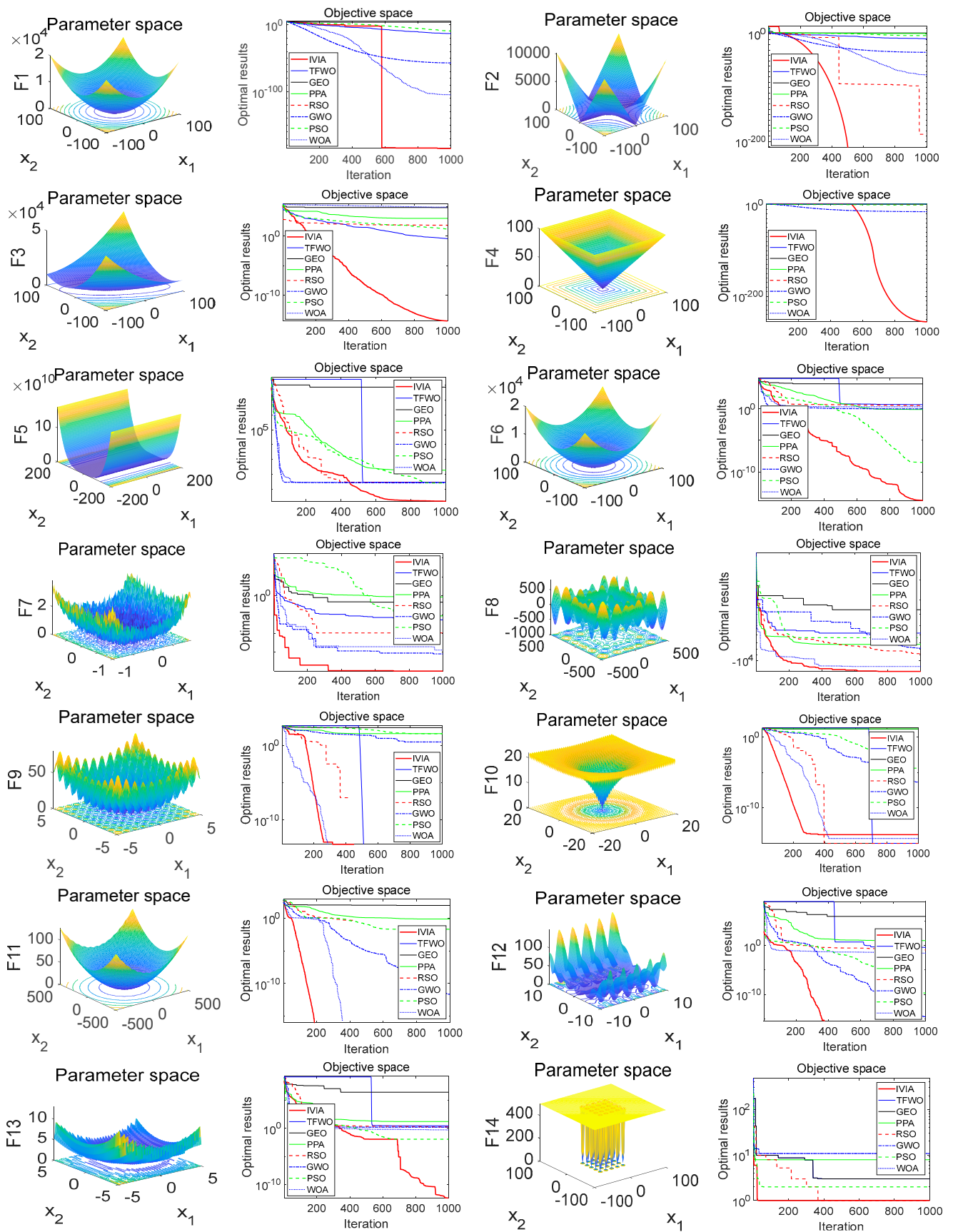


Figure 5. Cont.

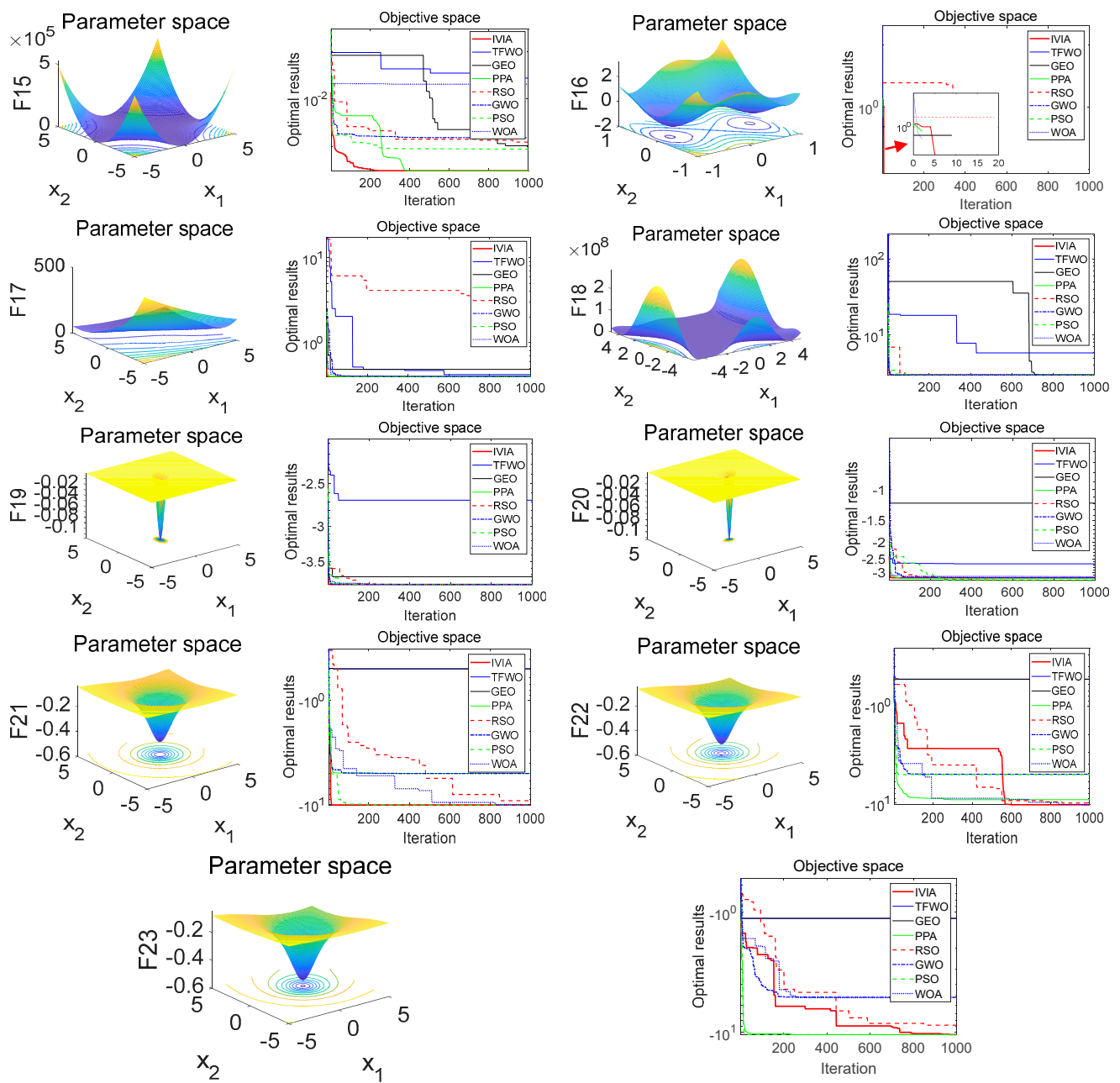


Figure 5. Standard test function optimization result.

Table 2. Results of the testing functions optimized by different algorithms.

F		IVIA	TFWO	GEO	PPA	RSO	GWO	PSO	WOA
F1	optimal	0.000×10^0	0.000×10^0	5.988×10^{-6}	0.000×10^0	0.000×10^0	0.000×10^0	8.428×10^{-5}	0.000×10^0
	average	1.820×10^{-248}	4.900×10^{-1}	2.136×10^{-1}	1.598×10^{-6}	9.458×10^{-102}	1.985×10^{-47}	6.789×10^{-12}	9.486×10^{-71}
	worst	1.245×10^{-162}	3.500×10^{-11}	5.121×10^{-1}	2.185×10^{-1}	1.169×10^{-13}	8.942×10^{-10}	8.628×10^{-18}	6.839×10^{-27}
F2	optimal	0.000×10^0	0.000×10^0	0.000×10^0	0.000×10^0	0.000×10^0	6.843×10^{-36}	6.880×10^{-10}	1.642×10^{-104}
	average	9.840×10^{-53}	1.249×10^{-8}	4.878×10^{-6}	9.785×10^{-7}	1.854×10^{-36}	6.842×10^{-13}	6.874×10^{-7}	4.859×10^{-47}
	worst	2.350×10^{-30}	1.895×10^{-6}	1.236×10^{-4}	8.545×10^{-5}	1.561×10^{-18}	6.842×10^{-6}	6.420×10^{-4}	6.465×10^{-28}
F3	optimal	0.000×10^0	2.855×10^{-1}	1.565×10^1	1.585×10^{-7}	1.265×10^{-6}	6.872×10^{-3}	4.265×10^{-2}	1.063×10^3
	average	4.185×10^{-11}	1.522×10^0	5.016×10^2	1.855×10^{-5}	1.295×10^{-6}	3.000×10^1	2.326×10^1	3.064×10^3
	worst	1.855×10^{-8}	5.489×10^0	1.000×10^5	1.855×10^{-3}	1.486×10^{-3}	1.014×10^2	6.019×10^1	1.036×10^4
F4	optimal	9.459×10^{-77}	1.125×10^0	2.153×10^0	2.126×10^0	3.165×10^{-3}	2.894×10^{-24}	8.486×10^{-2}	3.492×10^{-1}
	average	5.895×10^{-34}	2.459×10^0	4.153×10^0	3.124×10^0	2.989×10^0	1.987×10^{-14}	1.425×10^0	9.482×10^{-1}
	worst	2.349×10^{-19}	3.157×10^0	8.166×10^0	5.156×10^0	6.362×10^0	6.850×10^{-7}	6.895×10^0	3.843×10^0
F5	optimal	9.646×10^{-10}	2.360×10^{-1}	1.465×10^1	1.946×10^0	1.235×10^0	2.169×10^0	9.892×10^{-1}	1.979×10^0
	average	5.419×10^{-8}	6.523×10^0	5.988×10^2	1.249×10^1	1.565×10^1	1.249×10^1	1.468×10^1	9.162×10^0
	worst	1.242×10^{-2}	2.104×10^1	1.275×10^4	1.991×10^1	1.035×10^2	5.616×10^1	2.317×10^1	3.216×10^1
F6	optimal	2.355×10^{-18}	3.985×10^{-4}	8.852×10^{-2}	1.289×10^{-3}	1.965×10^{-6}	3.687×10^{-1}	7.985×10^{-10}	6.485×10^{-2}
	average	9.475×10^{-10}	9.125×10^{-1}	8.049×10^{-2}	1.597×10^{-1}	1.547×10^{-1}	9.006×10^{-1}	7.864×10^{-7}	9.717×10^{-1}
	worst	9.855×10^{-5}	1.836×10^0	3.598×10^0	1.099×10^0	6.218×10^{-1}	2.663×10^0	6.843×10^{-2}	2.097×10^0
F7	optimal	1.968×10^{-7}	0.000×10^0	3.942×10^{-2}	3.975×10^{-6}	4.885×10^{-3}	3.946×10^{-3}	9.847×10^{-5}	3.550×10^{-5}
	average	2.495×10^{-3}	1.698×10^{-1}	3.965×10^{-1}	3.735×10^{-1}	1.125×10^{-2}	8.972×10^{-1}	8.885×10^{-1}	4.165×10^{-1}
	worst	6.515×10^{-1}	1.317×10^0	1.569×10^0	1.032×10^0	3.339×10^{-1}	1.398×10^0	1.763×10^0	1.223×10^0
F8	optimal	-1.256×10^4	-1.024×10^4	-5.663×10^3	-1.124×10^4	-1.035×10^4	-9.176×10^3	-1.034×10^4	-1.224×10^4
	average	-1.156×10^4	-9.563×10^3	-4.983×10^3	-8.832×10^3	-8.865×10^3	-5.863×10^3	-6.856×10^3	-8.936×10^3
	worst	-1.055×10^4	-7.934×10^3	-3.527×10^3	-7.295×10^3	-5.235×10^3	-4.330×10^3	-5.845×10^3	-7.532×10^3
F9	optimal	0.000×10^0	4.946×10^{-6}	9.533×10^{-7}	0.000×10^0	1.968×10^{-22}	6.427×10^{-3}	6.848×10^{-8}	7.842×10^{-28}
	average	5.649×10^{-5}	1.568×10^{-3}	3.798×10^{-3}	1.765×10^{-3}	1.765×10^{-6}	3.499×10^{-5}	3.492×10^{-4}	9.787×10^{-9}
	worst	3.476×10^{-2}	1.065×10^0	9.833×10^{-1}	3.177×10^{-2}	1.685×10^{-1}	1.297×10^0	1.379×10^{-1}	3.790×10^{-3}
F10	optimal	9.852×10^{-21}	4.682×10^{-11}	1.649×10^{-4}	6.842×10^{-8}	9.845×10^{-6}	5.400×10^{-16}	6.463×10^{-5}	6.842×10^{-15}
	average	1.855×10^{-8}	4.190×10^{-3}	1.515×10^{-1}	6.850×10^{-3}	9.815×10^{-2}	8.943×10^{-3}	3.163×10^{-3}	8.463×10^{-8}
	worst	1.642×10^{-5}	1.684×10^{-1}	1.331×10^0	1.146×10^0	3.855×10^{-1}	6.315×10^{-1}	9.198×10^{-1}	6.318×10^{-3}
F11	optimal	5.989×10^{-9}	9.847×10^{-5}	5.646×10^{-3}	1.685×10^{-7}	9.946×10^{-5}	3.489×10^{-2}	3.965×10^{-4}	3.787×10^{-9}
	average	3.500×10^{-8}	3.490×10^{-1}	3.654×10^1	4.846×10^{-4}	1.654×10^{-3}	3.326×10^{-1}	1.038×10^0	9.779×10^{-5}
	worst	1.985×10^{-4}	2.165×10^0	1.068×10^2	3.168×10^0	5.486×10^{-1}	1.235×10^0	1.380×10^0	3.480×10^{-3}
	optimal	5.648×10^{-12}	2.165×10^{-6}	6.846×10^1	6.546×10^{-7}	6.468×10^{-7}	4.893×10^{-8}	4.896×10^{-13}	1.687×10^{-3}

Table 2. Cont.

F		IVIA	TFWO	GEO	PPA	RSO	GWO	PSO	WOA
F12	average	6.942×10^{-8}	1.656×10^{-3}	6.845×10^2	8.646×10^{-5}	6.579×10^{-4}	7.893×10^8	4.987×10^{-10}	6.447×10^{-1}
	worst	1.685×10^{-4}	9.924×10^{-2}	1.007×10^4	1.634×10^{-1}	6.846×10^{-2}	3.589×10^{-3}	4.983×10^{-7}	1.349×10^0
	optimal	6.115×10^{-8}	4.216×10^{-5}	2.647×10^1	5.242×10^{-2}	5.464×10^{-4}	4.685×10^{-4}	4.165×10^{-7}	4.198×10^{-8}
F13	average	4.359×10^{-20}	6.410×10^{-3}	1.464×10^2	5.825×10^0	1.351×10^{-2}	3.199×10^{-3}	4.147×10^{-4}	4.168×10^{-6}
	worst	4.642×10^{-8}	1.416×10^0	1.001×10^4	2.353×10^1	3.861×10^0	2.056×10^0	1.325×10^0	9.650×10^{-3}
	optimal	1.000×10^0	1.000×10^0	1.001×10^0	1.024×10^0	1.115×10^0	1.259×10^0	1.000×10^0	1.000×10^0
F14	average	1.001×10^0	1.674×10^0	1.009×10^0	2.196×10^0	1.351×10^0	1.950×10^0	1.350×10^0	1.268×10^0
	worst	1.007×10^0	2.419×10^0	2.486×10^0	1.117×10^1	2.652×10^0	1.286×10^1	2.149×10^0	1.635×10^8
	optimal	3.000×10^{-4}	5.419×10^{-4}	1.684×10^{-3}	6.554×10^{-4}	3.948×10^{-4}	9.157×10^{-4}	9.165×10^{-3}	3.917×10^{-4}
F15	average	3.425×10^{-4}	8.646×10^{-3}	1.682×10^{-3}	6.834×10^{-3}	6.422×10^{-2}	2.925×10^{-3}	6.117×10^{-3}	9.164×10^{-3}
	worst	6.453×10^{-3}	6.550×10^{-1}	1.954×10^{-1}	9.846×10^{-2}	9.648×10^{-1}	1.896×10^{-1}	1.430×10^{-2}	9.168×10^{-2}
	optimal	-1.032×10^0	-9.186×10^{-1}	-1.021×10^0	-1.001×10^0	-9.425×10^{-1}	-9.168×10^{-1}	-8.250×10^{-2}	-1.000×10^0
F16	average	-1.000×10^0	-8.634×10^{-1}	-9.648×10^{-1}	-9.545×10^{-1}	-8.217×10^{-1}	-2.198×10^{-1}	-1.625×10^{-8}	-5.491×10^{-1}
	worst	4.618×10^{-1}	-5.462×10^{-1}	-6.139×10^{-1}	-8.316×10^{-1}	-7.717×10^{-1}	1.198×10^{-1}	1.032×10^0	-1.493×10^{-1}
	optimal	3.981×10^{-3}	4.856×10^{-2}	4.404×10^{-1}	4.040×10^{-1}	3.990×10^{-2}	4.299×10^{-1}	4.399×10^{-1}	3.992×10^{-1}
F17	average	4.040×10^{-2}	4.000×10^{-2}	6.004×10^{-1}	6.040×10^{-1}	8.036×10^{-1}	6.498×10^{-1}	9.413×10^{-1}	1.065×10^0
	worst	5.033×10^{-1}	9.911×10^{-1}	1.032×10^0	2.030×10^0	1.603×10^0	2.064×10^0	2.319×10^0	1.916×10^0
	optimal	3.000×10^0	3.005×10^0	3.016×10^0	3.149×10^0	3.001×10^0	3.198×10^0	3.616×10^0	3.264×10^0
F18	average	3.015×10^0	3.147×10^0	3.656×10^0	3.648×10^0	3.004×10^0	3.616×10^0	4.086×10^0	3.949×10^0
	worst	3.146×10^0	3.142×10^0	4.411×10^0	4.245×10^0	3.133×10^0	4.691×10^0	4.620×10^0	5.017×10^0
	optimal	-3.860×10^0	-3.861×10^0	-3.195×10^0	-3.619×10^0	-3.812×10^0	-3.852×10^0	-3.801×10^0	-3.859×10^0
F19	average	-3.677×10^0	-3.852×10^0	-2.870×10^0	-3.595×10^0	-3.816×10^0	-3.742×10^0	-3.550×10^0	-3.726×10^0
	worst	-3.165×10^0	-3.832×10^0	-8.456×10^{-1}	-3.455×10^0	-3.795×10^0	-3.516×10^0	-2.949×10^0	-3.030×10^0
	optimal	-3.320×10^0	-3.146×10^0	-3.307×10^0	-3.032×10^0	-2.919×10^0	-1.982×10^0	-2.098×10^0	-3.156×10^0
F20	average	-2.996×10^0	-2.920×10^0	-3.019×10^0	-2.533×10^0	-1.398×10^0	-1.089×10^0	-9.497×10^{-1}	-1.979×10^0
	worst	-2.649×10^0	-2.520×10^0	-2.120×10^0	1.298×10^0	-5.945×10^{-1}	1.685×10^{-1}	-6.430×10^{-2}	-1.189×10^0
	optimal	-1.015×10^1	-1.007×10^1	-1.002×10^1	-1.015×10^1	-8.103×10^0	-1.002×10^1	-9.162×10^0	-1.006×10^1
F21	average	-9.122×10^0	-9.004×10^0	-9.654×10^0	-1.012×10^1	-4.137×10^0	-8.981×10^0	-7.198×10^0	-9.682×10^0
	worst	-7.596×10^0	-5.642×10^0	-8.514×10^0	-9.632×10^0	-6.517×10^{-1}	-1.192×10^0	-1.192×10^0	-8.336×10^0
	optimal	-1.040×10^1	-1.032×10^1	-1.035×10^1	-1.040×10^1	-1.039×10^1	-1.032×10^1	-9.198×10^0	-1.006×10^1
F22	average	-1.015×10^1	-8.250×10^0	-1.015×10^1	-5.365×10^0	-3.349×10^0	-9.195×10^0	-6.927×10^0	-8.198×10^0
	worst	-1.002×10^1	-4.820×10^0	-9.671×10^0	-2.389×10^{-1}	8.924×10^{-2}	-4.928×10^0	-1.162×10^0	-1.265×10^{-1}
	optimal	-1.052×10^1	-9.525×10^0	-1.051×10^1	-8.449×10^0	-1.046×10^1	-1.000×10^1	-9.198×10^0	-9.062×10^0
F23	average	-1.025×10^1	-6.489×10^0	-1.050×10^1	-4.168×10^0	-6.265×10^0	-5.984×10^0	-3.198×10^{-1}	-6.894×10^0
	worst	-9.219×10^0	-1.220×10^0	-9.647×10^0	-1.608×10^{-1}	-2.487×10^{-1}	-1.963×10^{-1}	1.625×10^{-2}	-1.820×10^0

3. Methodology

In this paper, we propose a method for power load forecasting based on LVMD-DBFCM load curve clustering, and the CNN-IVIA-BLSTM model is shown in Figure 6. First, data preprocessing is performed, and the influences of load are divided into continuous and discrete series. After normalizing the continuous sequences, they include load data, meteorological data (temperature, wind speed, and humidity), and economic factors. One-hot encoding is applied to discrete sequences, and the discrete sequences include the data types of workdays, weekends, and holidays. Load data decomposition and denoising with LVMD are performed on the preprocessed data. Then, the power load curve is clustered using the DBFCM algorithm. The processed data are extracted with a 1D CNN for feature extraction and finally, predicted with the IVIA-BLSTM model.

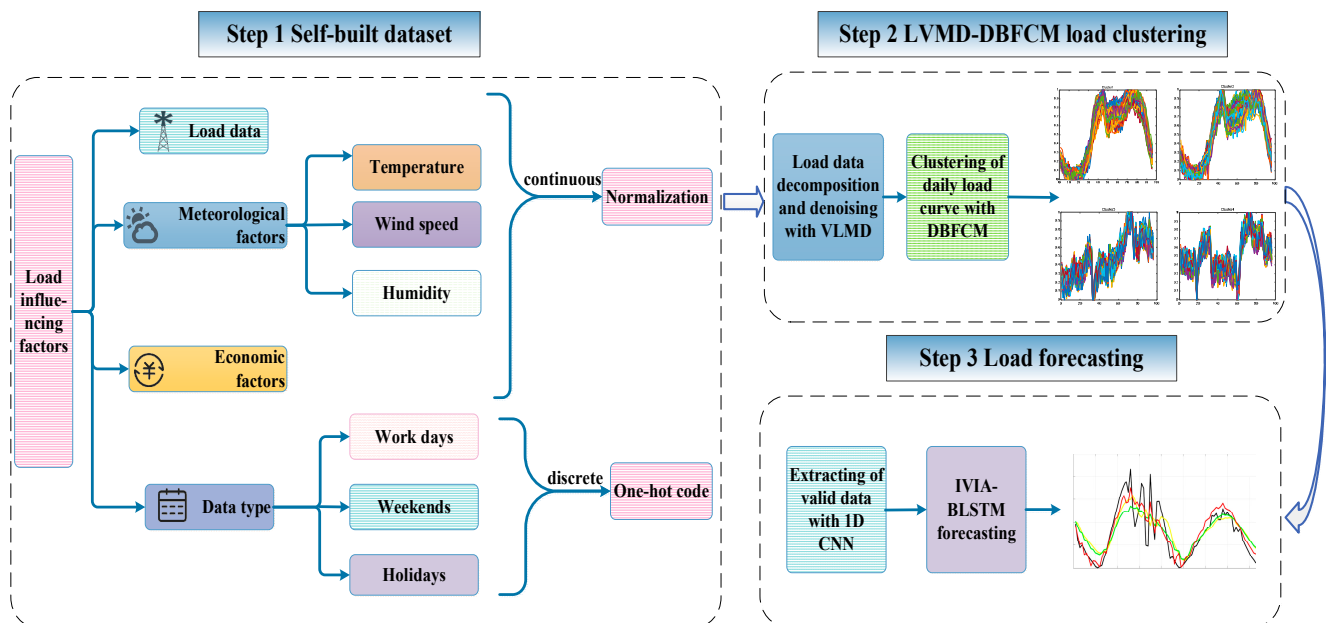


Figure 6. The overall structure design of this paper.

3.1. Self-Built Dataset

We expect to study the electrical load of a particular city or region where such datasets are rare. Data from the internet are collected and organized to produce an HS dataset. The following cities are replaced by H City because the electricity data relate to the economic performance of the country. The dataset includes 1096 days of electricity load data from 1 January 2016 to 31 December 2018, in H City. The dataset was divided; 80% is used as the training set and 20% is used as the test set.

The effects of the three factors of temperature, humidity, and wind speed on the power load are as follows. In summer, the temperature is high, and as air-conditioners usage increases, the power load increases. When the relative humidity is in the sensitive range 40~95%, the meteorological sensitive load varies significantly with the relative humidity. The load decreases with the increase of relative humidity. When the wind speed is in the sensitive range 2~6 m/s, the weather sensitive load changes significantly with the wind speed. The load increases with the increase of wind speed [27].

3.1.1. Dataset Visualization

The overall power load data from 2016–2018 are presented in Figure 7. From the box-whisker plot, the difficulty of prediction is reflected by the high number of data outliers in the morning from 6:00 to 8:00 and in the afternoon from 17:00 to 18:00 and 19:45 to 22:00. In summary, the importance of the data preprocessing and clustering and prediction models that follow in this paper is illustrated.

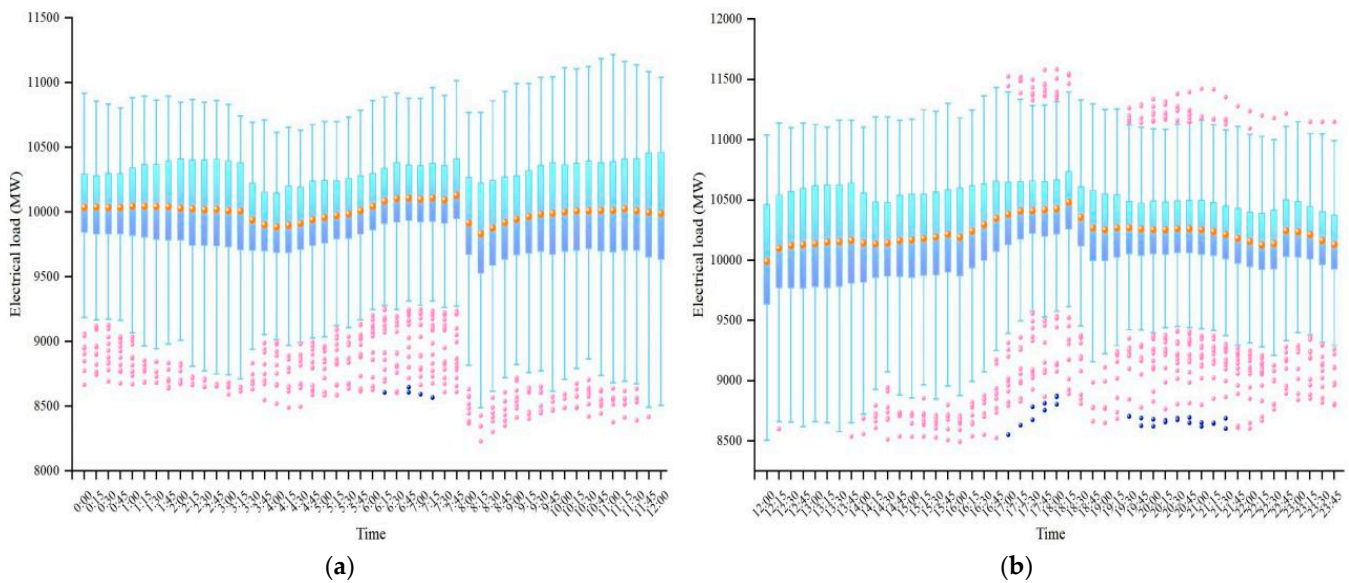


Figure 7. Distribution of whole load data in the dataset for 15 min. (a) Data from 00:00 to 12:00; (b) Data from 12:00 to 23:45. Box-whisker plot, the dots replace the abnormal value, the middle line of the box is the median of the data, which represents the average level of the sample data.

3.1.2. Data Preprocessing

- Abnormal data recognition and correction

Data preprocessing is first performed to resolve data outliers before clustering and prediction is performed [28].

Missing data: Lagrange interpolation is used to handle missing values.

Data duplication: Remove duplicate datasets by similarity.

Data mutation: 3 criteria.

- Data normalization

Normalized values for the clustering process can reduce the adverse effect on the clustering effect in the data domain due to the weight of the distance occupied between different attributes of the initial values. Normalization can eliminate the influence of the size of the load data volume on the distance in the clustering analysis, and thus, the information on the load pattern is highlighted [29].

Data normalization: Uniform metrics on data. Z Score normalization is shown in Equation (10). After normalization, the load data are normalized to the interval $[-1, 1]$.

$$Z_X = \frac{x_i - \bar{X}}{\sigma} \quad (10)$$

Data preprocessing: The equation for normalizing continuous data is given in Equation (11). Using One Hot encoding for discrete sequences, each state has unique register bits; only one bit is 1, and the rest are 0.

$$\text{MinMax}_x = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \quad (11)$$

3.2. LVMD-DBFCM Imbalanced Data Clustering

The goal of load clustering is to mine the typical daily load model [30].

Step 1. De-noising. The load data are decomposed and denoised using the LVMD (in Section 3.2.1).

Step 2. Clustering. The load data are clustered using the DBFCM (in Section 3.2.2).

Step 3. Clustering evaluation. The load data clustering effect are analyzed using the SSE evaluation index (in Section 3.2.3).

3.2.1. LVMD Double Decomposition Modal Strategy

- VMD Variational Mode Decomposition

VMD [31] is an iterative process that is used to search for the optimal solution of a variational model and to determine the mode $u_k(t)$ and its corresponding central frequency w_k and bandwidth. VMD has good denoising capabilities. VMD denoising are as follows:

1. Initialize parameters $\hat{u}_k^1, w_k^1, \hat{\lambda}^1$, and let $n = 0$.
2. $n = n + 1$, update w_k and u_k .
3. $k = k + 1$, repeat the previous step until $k = K$.
4. Update λ , and the equation is shown in Equation (12).
5. Repeat steps 2–5 until the end, when the condition of Equation (13) is satisfied.
6. The component that contains the minimum time information after decomposition is judged and considered to be random noise that is eliminated. Then, the data are reconstructed by the remaining modal component pairs, and the reconstructed data are the load data that do not contain noise.

$$\hat{\lambda}^{n+1}(\omega) = \hat{\lambda}^n(\omega) + \tau \left[\hat{f}(\omega) - \sum_k \hat{u}_k^{n+1}(\omega) \right] \tag{12}$$

$$\sum_k \left\| \hat{u}_k^{n+1} - \hat{u}_k^n \right\|_2^2 / \left\| \hat{u}_k^n \right\|_2^2 < e \tag{13}$$

- LMD Local Mean Decomposition

LMD is a new time–frequency analysis method that adaptively decomposes complex signals into a finite sum of product function (PF) components. Each of these PF components is actually a single component of the AM-FM information. LMD has good feature decomposition abilities [32,33]. The LMD principle is as follows.

1. First, find all the extreme points contained in the data series $x(t)$, assuming that the distribution of extreme points is $\{n_1, n_2, n_3, \dots\}$, and then calculate the mean m_i and envelope a_i of the adjacent extreme points according to Equations (14) and (15).

$$m_i = \frac{n_i + n_{i+1}}{2} \tag{14}$$

$$a_i = \frac{|n_i - n_{i+1}|}{2} \tag{15}$$

The local mean function curve $m_{11}(t)$ and the envelope function curve $a_{11}(t)$ are obtained by connecting m_i and a_i with line segments and smoothing, respectively, and the sliding average formula is given in Equation (16):

$$Y_s(i) = \frac{1}{2R + 1} (Y(i + R) + Y(i + R - 1) + \dots + Y(i - R)) \tag{16}$$

where $Y(i)$ is the sequence to be smoothed, $2R + 1$ is the sliding span, and R is the distance from $Y(i)$ to the starting point of the sliding process.

2. $m_{11}(t)$ is removed from the original signal $x(t)$ to obtain $h_{11}(t)$, and $s_{11}(t)$ is obtained by demodulating $h_{11}(t)$ using Equations (17) and (18):

$$h_{11}(t) = x(t) - m_{11}(t) \tag{17}$$

$$s_{11}(t) = \frac{h_{11}(t)}{a_{11}(t)} \tag{18}$$

When $a_{12}(t) = 1$, $s_{11}(t)$ decomposes into a pure FM function; if $a_{12}(t) \neq 1$, repeat the above process until $a_{1n}(t) = 1$ and $s_{1(n-1)}(t)$ is the corresponding pure FM function; see Equation (19) for details:

$$h_{1n}(t) = s_{1(n-1)}(t) - m_{1n}(t) \tag{19}$$

3. The PF component of the envelope signal is the product of the envelope signal and the pure FM signal, and the above steps are repeated after stripping the PF component from the original signal to obtain the new signal until the residual component is a monotonic function. The final result of LMD is shown in Equation (20):

$$x(t) = \sum_{p=1}^k PF_p(t) \tag{20}$$

where the last component is a monotonic function.

LMD decomposes the original signal into multiple high-frequency and low-frequency components without distorting the original signal in the decomposition process. If the sliding step of the sliding average algorithm is not chosen properly in the first step, the decomposition result will be greatly affected. In this paper, the original sliding average process is replaced by the three-time Hermite interpolation method to improve the decomposition accuracy of the LMD algorithm while reducing the overshoot and undershoot of the envelope. That is, after obtaining the extreme value points, the Hermite interpolation method is used to form the upper and lower envelopes, and then the other steps continue to apply the LMD without changes.

3.2.2. DBFCM Double Weights Fuzzy C-Means Algorithm

- FCM Fuzzy C-means Algorithm

FCM is a soft clustering algorithm and is unlike traditional hard clustering (HCM) algorithms, which allows the same object to belong to the same cluster [34]. By optimizing the objective function to obtain the affiliation of each sample point to all class centers, the affiliation range is [0, 1] to determine the class of the sample points to achieve the purpose of automatic classification of sample data [35].

The objective function and constraints *s.t.* of the FCM algorithm are given in Equation (21):

$$\begin{aligned} \min_{u_{ij}, c_i} J(u_{ij}, c_i) &= \sum_{i=1}^K \sum_{j=1}^N u_{ij}^m \|x_j - c_i\|^2 \\ \text{s.t. } \sum_{i=1}^K u_{ij} &= 1, \quad j = 1, 2, \dots, N \end{aligned} \tag{21}$$

where u_{ij} denotes that the sample belongs to the cluster affiliation value, m represents the fuzziness, K is the number of clusters, c_i is the i -th clustering center, and $\|x_j - c_i\|^2$ is the 2 parity of the Euclidean distance from each data point to the cluster center.

The clustering center and affiliation update equations are shown in Equations (22) to (23):

$$u_{ij} = \frac{1}{\sum_{l=1}^K \left(\frac{\|x_j - c_l\|}{\|x_j - c_i\|} \right)^{\frac{2}{m-1}}} \tag{22}$$

$$c_i = \frac{\sum_{j=1}^N u_{ij}^m x_j}{\sum_{j=1}^N u_{ij}^m} \tag{23}$$

- DBFCM Double Weights Fuzzy C-means Algorithm

Traditional FCM clustering is inaccurate in classification for holidays in small categories. From the date type, the power load data are unbalanced data. FCM has a “uniform effect of cluster size” issue, which affects the recognition effect and causes the algorithm to not recognize the holiday load model implied in the historical dataset [36,37]. In this paper, the DBFCM algorithm is proposed to improve the clustering performance of imbalanced data. The objective function of DBFCM utilizes the clustering volume as the weight and the weighted Euclidean distance as the metric distance.

1. The affiliation matrix defines the class volume, which is then introduced as a constraint *s.t.* into the traditional FCM algorithm objective function, as shown in Equation (24):

$$J_{DBFCM} = \sum_{i=1}^K \sum_{j=1}^N \frac{u_{ij}^m \|x_j - c_i\|^2}{v_j} \tag{24}$$

where v_j is the volume of the j -th class; see Equation (25). Constraint *s.t.* is shown in Equation (26):

$$v_j = \frac{\sum_{i=1}^N u_{ij}}{N} \tag{25}$$

$$s.t. \sum_{i=1}^K u_{ij} = 1, \quad j = 1, 2, \dots, N \tag{26}$$

The objective function of DBFCM uses the clustering volume as weights, which can balance the volume of each class in the clustering process. Thus, it can compensate for the unequal interactions between classes and improve the clustering performance of traditional algorithms for unbalanced data.

$$\frac{\partial J_{DBFCM}}{\partial c_i} = \frac{\partial \sum_{i=1}^N u_{ij}^m \|x_j - c_i\|^2 / v_j}{\partial c_i} \tag{27}$$

The derivative of Equation (27) with respect to the affiliation degree is constantly positive. Therefore, the Lagrange multiplier method is used to solve the affiliation degree and clustering center by using a greedy strategy and introducing constraint variables. The Lagrangian equation is given in Equation (28):

$$L = \sum_{j=1}^K \sum_{i=1}^N u_{ij}^m \|x_j - c_i\|^2 / v_j - \sum_{i=1}^N \lambda_i (\sum_{i=1}^K u_{ij} - 1) \tag{28}$$

The partial derivative of Equation (28) is found by setting the partial derivative = 0; see Equation (29):

$$\frac{\partial L}{\partial u_{ij}} = m u_{ij}^{m-1} \|x_j - c_i\|^2 / v_j - \lambda_j = 0 \tag{29}$$

The updated equations for the affiliation u_{ij} as well as the clustering center c_i are given in Equations (30) to (31):

$$u_{ij} = \frac{\frac{v_j}{\|x_j - c_i\|^{\frac{2}{m-1}}}}{\sum_{q=1}^K \frac{v_q}{\|x_j - c_i\|^{\frac{2}{m-1}}}} \tag{30}$$

$$c_i = \frac{\sum_{j=1}^N u_{ij}^m x_j}{\sum_{j=1}^N u_{ij}^m} \tag{31}$$

The DBFCM algorithm is more biased towards small classes and avoids the problem of the FCM uniformity effect when dealing with the same sample Euclidean clustering.

2. The load curve is characteristically weighted by introducing a weighting factor w to the traditional Euclidean distance d_{ij} . The DBFCM is used to perform cluster analysis on the decomposed and reconstructed load curves of LVMD to improve the optimal number of clusters.

The degree of influence of each dimensional feature of the analyzed sample on the classification result is assumed to be w . The feature weight coefficients of each dimension $w = \{w_1, w_2, \dots, w_n\}$ are introduced to the Euclidean distance to form a weighted Euclidean distance, which controls the weights of each dimensional feature vector. The traditional Euclidean distance d_{ij} and the Euclidean distance d_{ij}^* with the addition of weight w are expressed in Equations (32) to (33):

$$d_{ij} = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ik} - x_{jk}|^2} \tag{32}$$

$$d_{ij}^* = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \dots + w_n|x_{ik} - x_{jk}|^2} \tag{33}$$

After DBFCM adds weight w , the clustering center c_i does not change, but the objective function J_{DBFCM} , the affiliation u_{ij} , and the Euclidean distance from sample x_j to the clustering center c_i are all affected by the weighted Euclidean distance d_{ij}^* .

In summary, the objective function and affiliation of DBFCM are shown in Equations (32) to (33):

$$J_{DBFCM} = \sum_{i=1}^K \sum_{j=1}^N u_{ij}^m \|x_j - c_i\|_*^2 / v_j \tag{34}$$

$$u_{ij}^* = \frac{\frac{v_j}{\|x_j - c_i\|_*^{\frac{2}{m-1}}}}{\sum_{q=1}^K \frac{v_q}{\|x_j - c_i\|_*^{\frac{2}{m-1}}}} \tag{35}$$

3.2.3. Clustering Validity Quality Evaluation

The clustering criterion used in the clustering process is usually the sum of squared error (SSE). The analytical formula of SSE is shown in the Equation (36):

$$SSE(q) = \sum_{i=1}^q \sum_{x \in p_i} \|c_i - x\|^2 \tag{36}$$

In the formula, c_i is the i -th cluster center, and p_i represents the aggregation of data points in the i -th cluster. As the number of clusters q increases, the samples are classified more accurately, and the SSE decreases. Theoretically, the smaller the SSE, the better the clustering effect. As the value of q increases to a certain level, the rate of the decline in SSE slows down. As shown in Figure 8, when $q = 5$, it is the inflection point of the curve. The most suitable cluster number is 5.

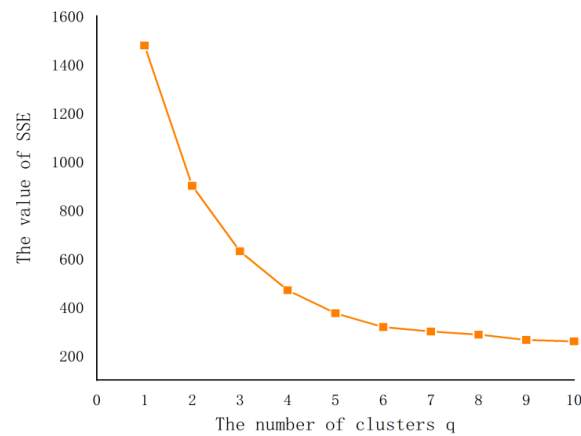


Figure 8. The gradient graph of SSE with increasing cluster number q from 1 to 10.

In this paper, two clustering validity indicators are used to assess the quality of clustering: the Calinski–Harabasz indicator (CHI) and the Davies–Bouldin indicator (DBI). The CHI is obtained by the ratio of the compactness to the separation [38]. Thus, the larger the index is, the more compact the result is. The DBI metric estimates intraclass closeness by the distance from the sample point within a class to the center of the class to which it belongs, and the distance between class centers indicates the interclass dispersion [39]. Thus, the smaller the index is, the better the effect. Equations (37) to (38) describe these metrics:

$$CHI = \frac{Tr(B_k)}{Tr(W_k)} \times \frac{n - k}{k - 1} \quad (37)$$

$$DBI = \frac{1}{k} \sum_{i=1}^k \max_{j=1, \dots, k, i \neq j} \frac{s_i + s_j}{\|c_i - c_j\|_2^2} \quad (38)$$

3.3. The Proposed CNN-IVIA-BLSTM Forecasting Model

Using the CNN-IVIA-BLSTM model to forecast the power load data:

Step 1. Feature fusion extraction. CNN is used to extract the data feature.

Step 2. Forecasting. IVIA optimizes the parameters related to the BLSTM network, then uses the optimized BLSTM to predict.

Step 3. Power load forecast evaluation. The load data forecasting effect analyzed using RMSE, MAE, and MAPE.

3.3.1. CNN: Convolutional Neural Network

A CNN can automatically extract potential features between massive loads of continuous and discontinuous data to build a compressed complete feature vector for the top fully connected layer [40]. It is a hierarchical neural feedforward network and consists of a series of network layers with different functions. It mainly comprises the input layer, implied layer, and output layer. Among these, the implied layer takes the convolutional layer as the core, and the main function of the convolution layer is to extract the feature fusion of the input data as an arithmetic of the convolution layer. In addition, the CNN includes pooling layers (to reduce the number of parameters of the neural network) and the Dropout layer (to prevent data overfitting). The 1D convolutional kernel is two-dimensional and has a length and width; however, there is a sliding window in the width or height direction, and multiplication is transformed into addition. In this paper, a 1D CNN is used, with a convolutional kernel size of 3×3 , a sliding window of 10, and a step size of 1. The overall structure of the convolutional network is shown in Figure 9. A convolutional block is a combination of M convolutional layers and b convergence layers. It can be stacked with N consecutive convolutional blocks followed by K fully connected layers.

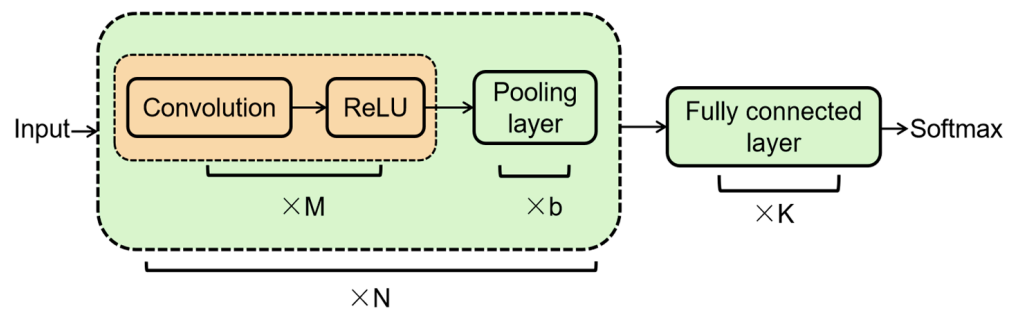


Figure 9. Overall structure of a CNN.

3.3.2. BLSTM Bidirectional Long Short-Term Memory Network

BLSTM can remember a sequence well, can solve the dependency problem of longer time spans, and has a strong advantage in improving time series correlation. Based on the LSTM (long short-term memory) network, the forward and reverse network structures form a closed loop of information, which can better verify and correct the process error information while maintaining the bidirectional data information, which has stronger robustness [41]. The LSTM network structure is shown in Figure 10, and the relevant calculation equations are shown in (39)–(44).

$$\text{Input gate : } i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{39}$$

$$\text{Forget gate : } f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{40}$$

$$\text{Output gate : } o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{41}$$

$$\text{New memory unit : } c_t' = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{42}$$

$$\text{Final memory unit : } c_t = f_t \otimes c_{t-1} + i_t \otimes c_t' \tag{43}$$

$$\text{Output : } h_t = o_t \otimes \tanh(c_t) \tag{44}$$

where $W_i, W_f,$ and W_o represent input weight vectors; $U_i, U_f,$ and U_o represent upper output weight vectors; and $b_i, b_f, b_o,$ and b_c are bias vectors. Sigmoid is generally selected as the excitation function for σ , which mainly plays a role of gating. It has an output between 0 and 1, which matches the physical definition of gating and is very close to 1 or 0 when the input is large or small, thus ensuring that the gate is open or closed. The tanh is an option to generate the new memory unit c_t' due to a faster convergence rate with an output between -1 and 1 , which coincides with the center of the feature distribution being 0 in most scenarios. The related formulas are given in Equations (45) and (46):

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{45}$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{46}$$

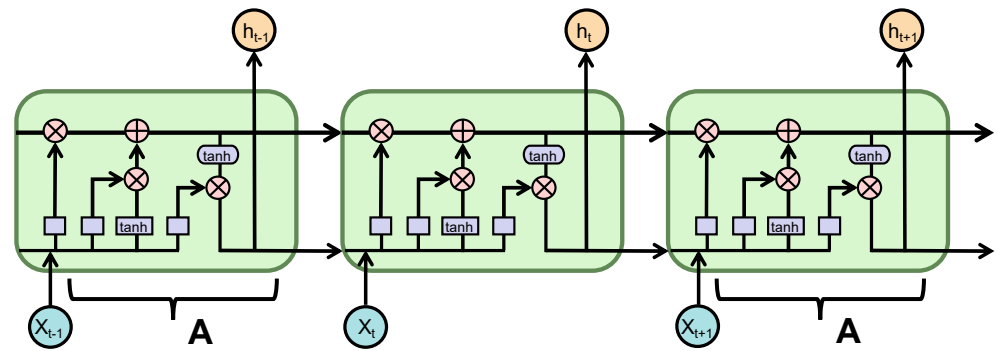


Figure 10. LSTM network structure diagram.

LSTM can only be passed in one direction, and the unit computation of the bidirectional neural network is connected with the unidirectional one. BLSTM consists of an input layer, forward LSTM layer, reverse LSTM layer, and output layer. However, the hidden layer of the bidirectional neural network must save two values, i.e., A , which is involved in the forward calculation, and A' , which is involved in the reverse calculation. The final output value depends on the sum. The BLSTM network structure is shown in Figure 11.

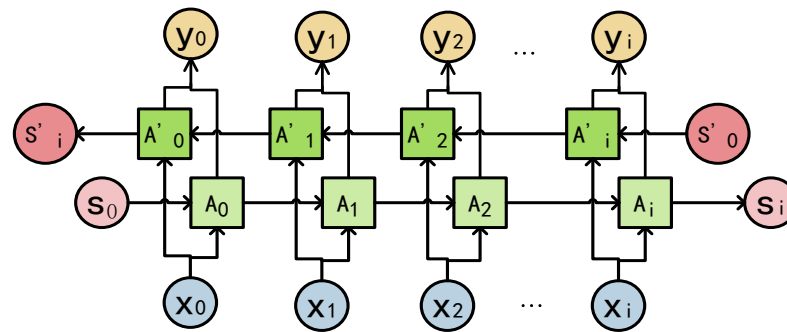


Figure 11. BLSTM network structure diagram.

However, a common problem is that BLSTM networks have many parameters, and without proper optimization, the model may be overfitted or slow to train, resulting in inefficiency. Too few nodes in the hidden layer will cause the model to not have the necessary learning ability and information processing capability, and too many nodes will increase the complexity of the network structure and make the learning process easily fall into local minima, which makes the network slow. When the learning rate is too high, the cost function is not easy to reduce to the lowest point, it is not easy to converge at the lowest point, and the convergence effect is poor. When too much training is performed, the gradient descent process may cross the nadir, which causes the training rate to be too low. In the case of reasonable tuning parameters, the more layers and neurons there are, the higher the accuracy rate; however, this can also lead to the overfitting phenomenon, and a regularization process can be used to solve the overfitting problem. In this paper, the number of implicit layer neurons, training times, learning rates, and regularization coefficients in BLSTM are optimized using the IVIA to improve the performance of the sequence modelling task.

3.3.3. Hybrid Forecasting Model

The CNN-IVIA-BLSTM model is a combined prediction model consisting of a CNN and BLSTM. BLSTM is a commonly used model architecture of deep learning models for sequence modelling tasks, and it has achieved good performance in many tasks. The CNN is first used to extract the feature vectors consisting of load influencing factors, which can be considered a “feature extractor”, then the extracted feature vectors are used for load prediction with the BLSTM model optimized by the IVIA. The hybrid prediction model

structure is shown in Figure 12, which takes full advantage of both the CNN and BLSTM network to ensure the accuracy of the power load.

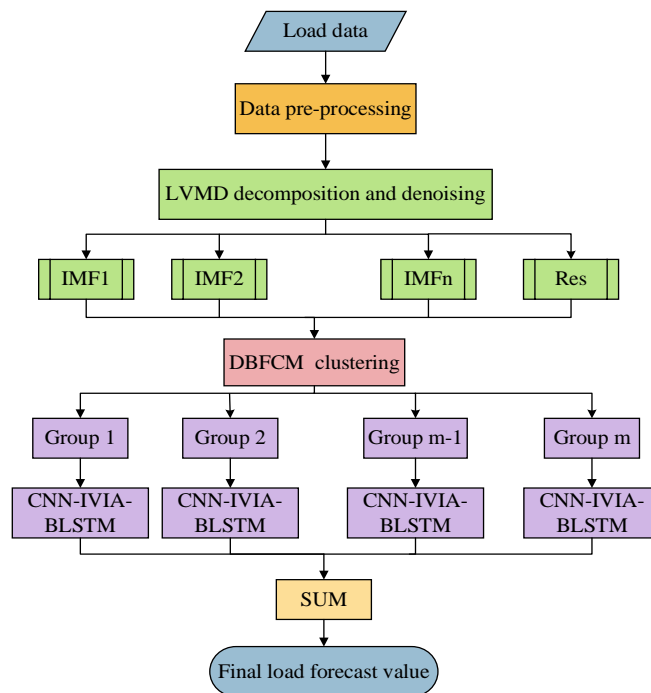


Figure 12. Hybrid forecasting model structure diagram.

The forecasting steps are as follows:

1. Selected information as model input.
2. LVMD decomposes and denoises the original sequence, and DBFCM performs clustering.
3. The IVIA population size N , the maximum number of iterations M , and the initial search range of the parameters (the number of neurons in the hidden layer H , training number E , learning rate η , and regularization factor $L2$) are set. The root mean square error (y_{RMSE}) is used as the objective function in the optimization algorithm, and finally, the model of the influenza virus immunization algorithm coupled with the bidirectional long and short-term memory network is developed.
4. The 1D CNN reads the load sequence with a sliding time window of 10 and a step size of 1 for feature extraction.
5. m prediction models are obtained by inputting the CNN-IVIA-BLSTM prediction models for each component separately.
6. Finally, the predicted values of the m prediction models are combined to obtain the predicted values of the load.

3.3.4. Power Load Forecast Evaluation Indicator

Three evaluation indexes are set as y_{MAPE} (mean absolute percentage error, MAPE), y_{RMSE} (root mean square error, RMSE), and y_{MAE} (mean absolute error, MAE). The equations are shown in (47) to (49):

$$y_{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{X_{act}(i) - X_{pred}(i)}{X_{act}(i)} \right| \tag{47}$$

$$y_{RMSE} = \sqrt{\frac{\sum_{i=1}^n (X_{act}(i) - X_{pred}(i))^2}{n}} \tag{48}$$

$$y_{MAE} = \frac{1}{n} \sum_{i=1}^n |X_{act}(i) - X_{pred}(i)| \quad (49)$$

In the above equation, n is the total number of predictions, $X_{act}(i)$ is the real value of the load at moment i , and $X_{pred}(i)$ is the predicted value of the load at moment i .

4. Case Analysis

In this paper, historical power load data from 2016 to 2018 in H city were used for prediction with a sampling interval of 15 min. LSTM, BLSTM, CNN-BLSTM, LVMD-DBFCM-CNN-BLSTM, and the proposed model in this paper are selected to compare the prediction results.

4.1. Analysis of Data Processing Results

4.1.1. Data Denoising and Decomposition

The original load data are decomposed and denoised using VMD, as shown in Figure 13, and the original load data sequence is decomposed into 8 IMF components. From the figure, we can see that the IMF7–IMF8 values are small and contain more noisy data, which are removed from the original data, then the first six IMF components are used to reconstruct the data to achieve data denoising.

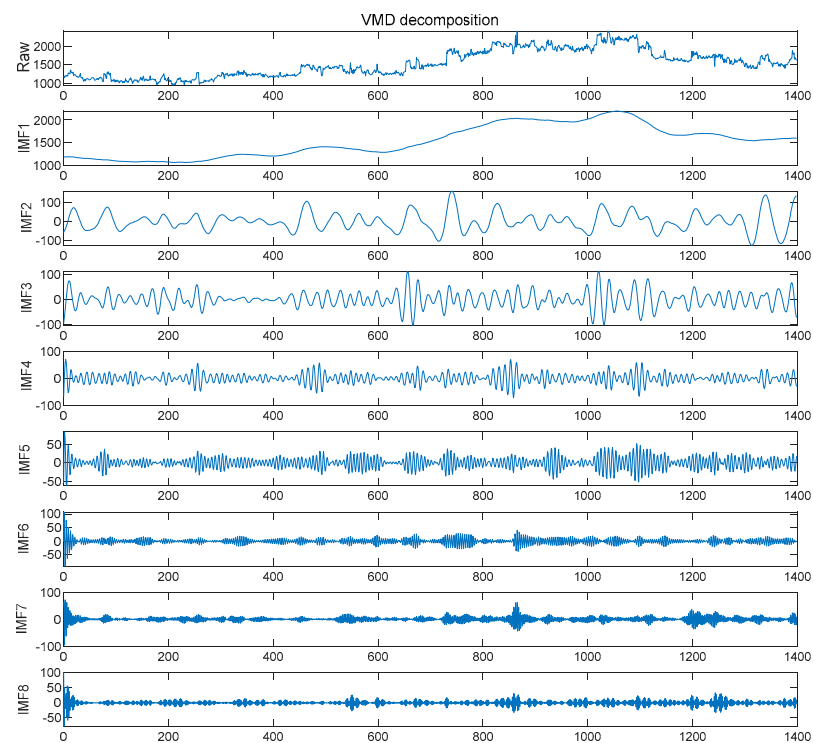


Figure 13. Result of VMD decomposition.

The second step of LVMD is to redecompose the denoised data with LMD. The denoised data are decomposed into a total of six PF components, and the decomposition results are shown in Figure 14. The values of PF1 are larger and maintain the same trend as the original data, containing the main valid information of the data. The values of PF2–PF6 are smaller, among which the periodic changes of PF2–PF4 are more obvious and PF6 is monotonically increasing, which is convenient for the prediction of each component. The randomness of the F5 sequence volatility is strong and the prediction accuracy of the F5 component is not high when making predictions.

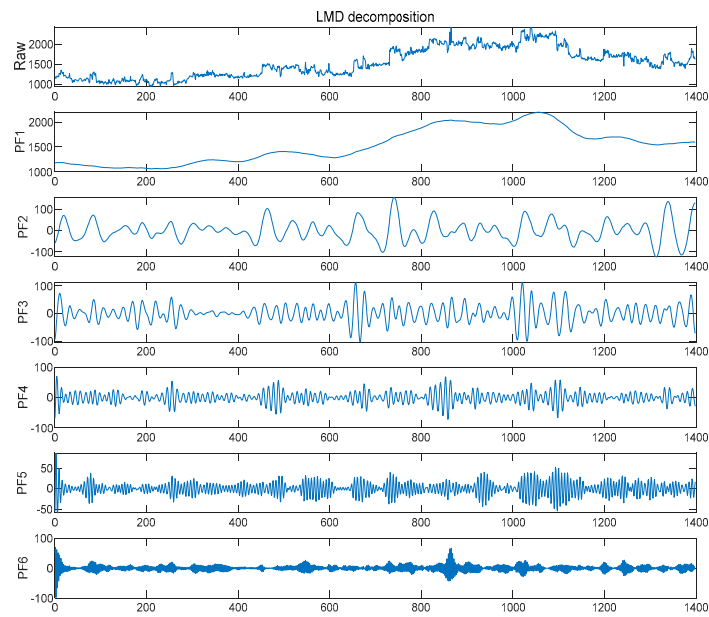


Figure 14. Result of LMD decomposition.

4.1.2. Analysis of the Clustering Results

The daily load curve after LVMD-DBFCM clustering is shown in Figure 15.

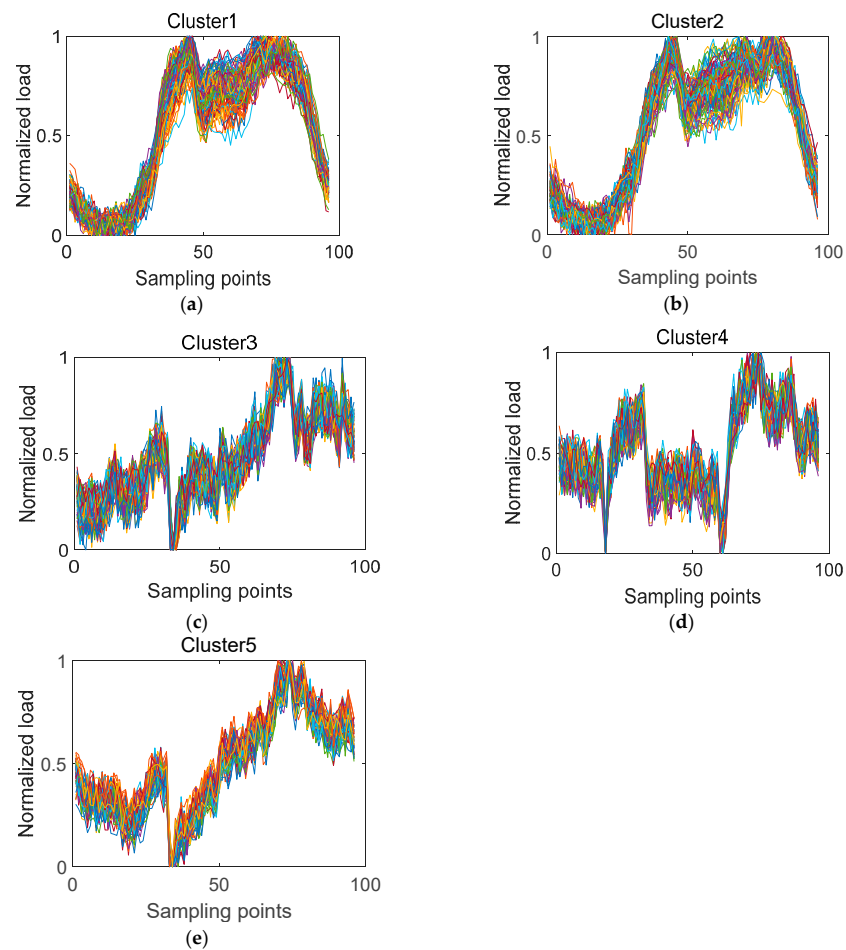


Figure 15. (a–e) Daily load curve after LVMD-DBFCM clustering when the number of clusters is determined to be 5.

The clustering algorithm proposed in this paper is compared with k-means [42], FCM [43], and DBFCM. Their comparison is displayed in Table 3 and Figure 16. From the results, it can be seen that the maximum value of the CHI metric is 1142.509 and the minimum value of the DBI metric is 1.083. The clustering validity of the method proposed in this paper is better than that of the other three methods.

Table 3. The comparative table displaying the clustering validity indicators, CHI and DBI, of the K-means, FCM, DBFCM, and LVMD-DBFCM methods.

Methods	CHI	DBI
K-means	721.201	1.549
FCM	814.014	1.401
DBFCM	1024.743	1.216
LVMD-DBFCM	1142.509	1.083

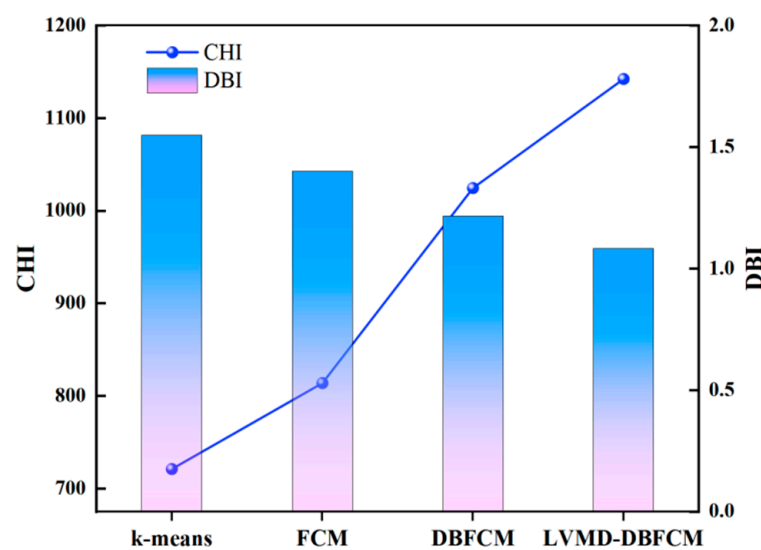


Figure 16. Comparative graph displaying the clustering validity indicators, CHI and DBI, of the K-means, FCM, DBFCM, and LVMD-DBFCM methods.

4.2. Analysis of the Prediction Results

The comparison between the prediction model proposed in this paper and the four models LSTM, BLSTM, CNN-BLSTM, and LVMD-DBFCM+CNN-BLSTM using three load evaluation indices, RMSE, MAE, and MAPE, is shown in Table 4 and Figure 17. From the results, we can see that the prediction models proposed in this paper have better prediction results than the other four models in all evaluation indices.

Table 4. The comparative table displaying the clustering validity indicators, CHI and DBI, of the K-means, FCM, DBFCM, and LVMD-DBFCM methods.

Models	RMSE	MAE	MAPE
LSTM	101.4817	80.0980	5.7087%
BLSTM	83.1498	61.4826	4.4460%
CNN-BLSTM	79.3340	58.0245	4.1782%
LVMD-DBFCM+CNN-BLSTM	57.7316	42.3669	2.9946%
Proposed	31.9942	23.3691	1.6421%

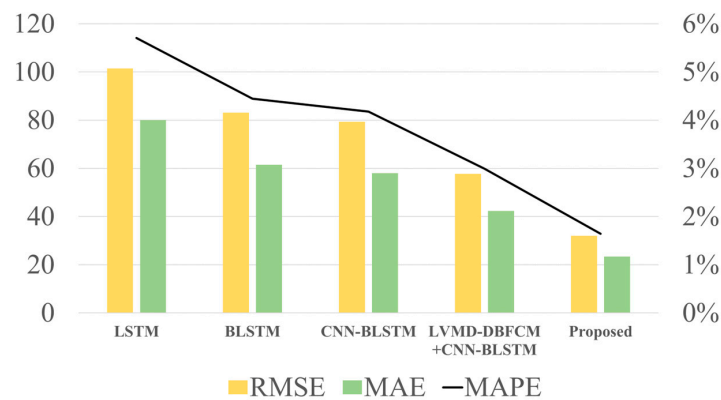


Figure 17. Comparison of the RMSE, MAE, and MAPE results for 5 models' predictions.

Figure 18 shows the prediction graphs of the five prediction models. From the figure, it can be clearly seen that the pink line, representing the model proposed in this paper, fits so closely to the target curve represented by the black line, which can better reflect the trend of the target, indicating that the model proposed in this paper has the better prediction effect than addressed models. Figure 19 shows the prediction error comparison of the 5 models, from which it can be clearly seen that the model proposed in this paper, represented by yellow, has the lowest error.

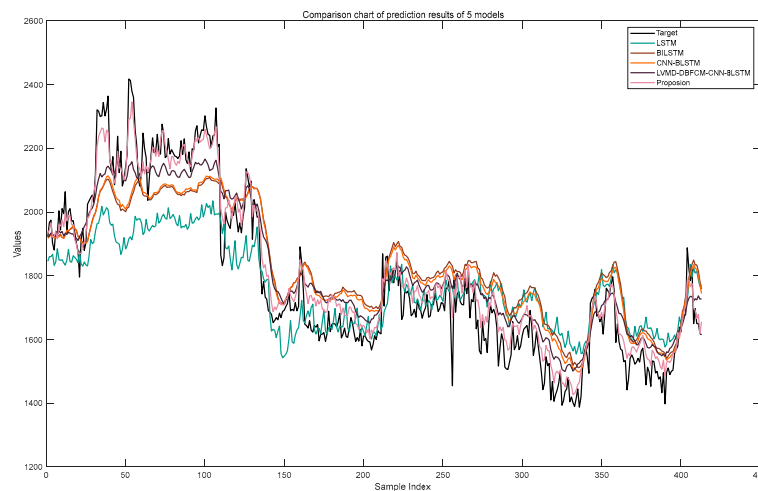


Figure 18. Comparison chart of the prediction results of 5 models.

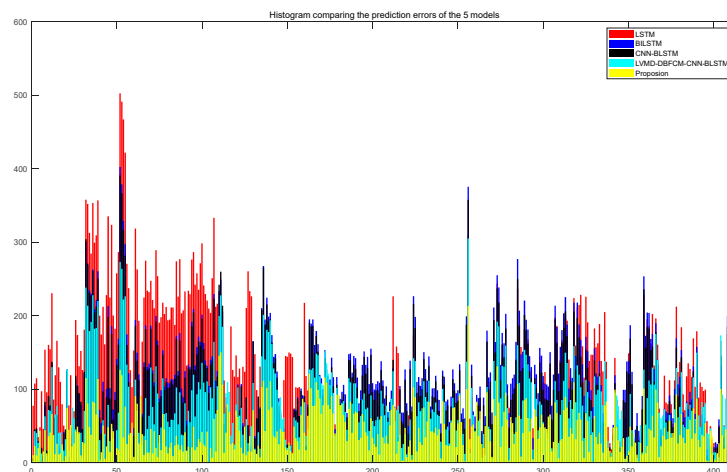


Figure 19. Comparison of prediction errors of the 5 models.

5. Conclusions

In this paper, a new model based on LVMD-DBFCM load curve clustering and a CNN-IVIA-BLSTM hybrid model for power load forecasting is proposed. This comprehensive technique takes historical load data and influencing factors (meteorology, economy, and data type) into account, where historical load data, meteorological factors, and economic factors are normalized, and the data types are uniquely heat coded.

The novel LVMD-DBFCM algorithm improves the continuity and stability of the data, and the values of the CHI and DBI quality assessment indicators are 1142.509 and 1.083, respectively, both of which reflect the good validity of the clustering method used in this paper. In the new CNN-IVIA-BLSTM model, a CNN is used for feature extraction, BLSTM is used for load forecasting, and the IVIA is used to optimize the relevant parameters in the BLSTM network. The results of the three electric load forecasting evaluation metrics of the hybrid forecasting model show that the RMSE is 31.9942, the MAE is 23.3691, and the MAPE is 1.6421%. The prediction effect of the electric load fits well with the target, and the prediction error is minimized.

Author Contributions: Conceptualization, L.H., J.W., Z.G. and T.Z.; Data curation, L.H., J.W., Z.G. and T.Z.; Formal analysis, L.H., J.W. and T.Z.; Investigation, L.H.; Methodology, L.H. and J.W.; Software, J.W. and Z.G.; Project administration, L.H.; Resources, L.H.; Supervision, L.H.; Validation, L.H.; Visualization, L.H.; Writing—original draft, L.H., J.W., Z.G. and T.Z.; Writing—review & editing, J.W. and Z.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Huang, Y. Medium-Long Term Power Load Forecasting of a Region. Master's Thesis, Xihua University, Chengdu, China, 2015.
2. Han, F.J.; Wang, X.H.; Qiao, J. Review on Artificial Intelligence based Load Forecasting Research for the New-type Power System. *Proc. CSEE* **2023**, *1*–24.
3. Zhang, Y.; Wang, A.H.; Zhang, H. Overview of smart grid development in China. *Power Syst. Prot. Control* **2021**, *49*, 180–187.
4. Gao, D.D.; Gao, S.T. Review on medium-long term power load forecasting study. *Sci. Technol. Innov. Her.* **2014**, *11*, 25.
5. Shang, C.; Gao, J.; Liu, H.; Liu, F. Short-term load forecasting based on PSO-KFCM daily load curve clustering and CNN-LSTM model. *IEEE Access* **2021**, *9*, 50344–50357. [[CrossRef](#)]
6. Huang, J.B. Based on trend extrapolation method analysis of city complexes load development characteristics of the chapter. *Rural Electrification*. **2019**, *7*, 39–42.
7. Song, F.; Liu, J.; Zhang, T. The Grey Forecasting Model for the Medium-and Long-Term Load Forecasting. *J. Phys. Conf. Ser.* **2020**, *1654*, 012104. [[CrossRef](#)]
8. Fan, G.F.; Qing, S.; Wang, H. TSupport Vector Regression Model Based on Empirical Mode Decomposition and Auto Regression for Electric Load Forecasting. *Energies* **2013**, *6*, 1887–1901. [[CrossRef](#)]
9. Qiu, X.P. Neural Network and Deep Learning. *J. Chin. Inf. Process.* **2020**, *34*, 4.
10. Jarquin, C.S.S.; Gandelli, A.; Grimaccia, F.; Mussetta, M. Short-Term Probabilistic Load Forecasting in University Buildings by Means of Artificial Neural Networks. *Forecasting* **2023**, *5*, 390–404. [[CrossRef](#)]
11. Liu, J.H. Power load forecasting in Shanghai based on CNN-LSTM combined model. *Proc. SPIE* **2022**, *12254*, 125–132.
12. Li, J.W.; Luong, T.; Dan, J. When are tree structures necessary for deep learning of representations. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015.
13. Gu, H.C.; Mou, P.; Li, J.W. Modeling and application of ethylene cracking furnace based on cross-iterative BLSTM network. *CIESC J.* **2019**, *70*, 548–555.
14. Xu, Y.; Xiang, Y.F.; Ma, T.X. Short-term Power Load Forecasting Method Based on EMD-CNN-LSTM Hybrid Model. *J. North China Electr. Power Univ.* **2022**, *49*, 81–89.
15. Gao, W.C. Urban Gas Load Forecasting Based on EWT-CNN-LSTM Model. Master's Thesis, Shanghai Normal University, Shanghai, China, 2021.

16. Sun, G.L.; Li, B.J.; Xu, D.M.; Li, Y.P. Monthly Runoff Prediction Model Based on VMD-SSA-LSTM. *Water Resour. Power* **2022**, *40*, 18–21.
17. Yang, Y. Research and Application of Medium and Long Term Load Forecasting Technology. Master's Thesis, Shenyang Institute of Computing Technology, Chinese Academy of Science, Shenyang, China, 2021.
18. Mojtaba, G.; Faraji, D.I.; Ebrahim, A.; Abolfazl, R. A novel and effective optimization algorithm for global optimization and its engineering applications: Turbulent Flow of Water-based Optimization (TFWO). *Eng. Appl. Artif. Intell.* **2020**, *92*. [[CrossRef](#)]
19. Abdolkarim, M.B.; Mahmoud, D.N.; Adel, A. Golden eagle optimizer: A nature-inspired metaheuristic algorithm. *Comput. Ind. Eng.* **2021**, *152*, 107050.
20. Mohamed, A.A.; Hassan, S.A.; Hemeida, A.M.; Alkhalaf, S. Parasitism-Predation algorithm (PPA): A novel approach for feature selection. *Ain Shams Eng. J.* **2019**, *11*, 293–308. [[CrossRef](#)]
21. Dhiman, G.; Garg, M.; Nagar, A.; Kumar, V. A novel algorithm for global optimization: Rat Swarm Optimizer. *J. Ambient Intell. Humaniz. Comput.* **2020**, *12*, 8457–8482. [[CrossRef](#)]
22. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
23. Ramya, S.; Rajesh, N.B.; Viswanathan, B. Particle Swarm Optimization (PSO) based optimum Distributed Generation (DG) location and sizing for Voltage Stability and Loadability Enhancement in Radial Distribution System. *Int. Rev. Autom. Control IREACO* **2014**, *7*, 288–293.
24. Jhila, N.; Modarres, K.F.; Akiko, Y. A whale optimization algorithm (WOA) approach for clustering. *Cogent Math. Stat.* **2018**, *5*, 1483565.
25. Karol, S.; Ewelina, H.; Katarzyna, L.; Katarzyna, K. Spread of Influenza Viruses in Poland and Neighboring Countries in Seasonal Terms. *Pathogens* **2021**, *10*, 316.
26. Oxford, J.S. Special Article: What Is the True Nature of Epidemic Influenza Virus and How Do New Epidemic Viruses Spread. *Epidemiol. Infect.* **1987**, *99*, 1–3. [[CrossRef](#)] [[PubMed](#)]
27. Zhu, Z.W. Study on the Impact of Weather Factors on Characteristics of Electric Load. Ph.D. Thesis, Zhejiang University, Hangzhou, China, 2008.
28. Xu, F.; Weng, G.Q. Research on Load Forecasting Based on CNN-LSTM Hybrid Deep Learning Model. In Proceedings of the 2022 IEEE 5th International Conference on Electronics Technology (ICET), Chengdu, China, 13–16 May 2022; pp. 1332–1336.
29. Hu, Y.C. Research on Power Load Pattern Recognition Method Based on Improved k-means Clustering Algorithm. Master's Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2018.
30. Zhang, H.; Zhang, Y.; Xu, Z. Thermal Load Forecasting of an Ultra-short-term Integrated Energy System Based on VMD-CNN-LSTM. In Proceedings of the 2022 International Conference on Big Data, Information and Computer Network (BDICN), Sanya, China, 20–22 January 2022; pp. 264–269.
31. Dragomiretskiy, K.; Zosso, D. Variational mode decomposition. *IEEE Trans. Signal Process.* **2014**, *62*, 531–544. [[CrossRef](#)]
32. Yang, S.S.; Zhou, H.; Zhao, H.Y. Fault diagnosis method for the bearing of reciprocating compressor based on LMD multiscale entropy and SVM. *J. Mech. Transm.* **2015**, *39*, 119–123.
33. Wu, D.S.; Yang, Q.; Zhang, J.Y. Ensemble fault diagnosis method based on VMD-LMD-CNN. *Bearing* **2020**, *10*, 57–63.
34. Li, P.S.; Li, X.R.; Chen, H.H. The characteristics classification and synthesis of power load based on fuzzy clustering. *Proc. CSEE* **2005**, *25*, 73–78.
35. Zhou, K.L. Theoretical and Applied Research on Fuzzy C-means Clustering and Its Cluster Validation. Ph.D. Thesis, Hefei University of Technology, Hefei, China, 2014.
36. Dong, T. Research on Electrical Load Pattern Recognition and Load Forecasting Based on Deep Learning. Master's Thesis, Jilin University, Changchun, China, 2022.
37. Ma, Z.B.; Xu, S.A.; Zhu, S.B. Power Load Classification Based on Feature Weighted Fuzzy Clustering. *ELECTRIC POWER* **2022**, *55*, 25–32.
38. Tomasini, C.; Emmendorfer, L.; Borges, E.N. A methodology for selecting the most suitable cluster validation internal indices. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (SAC '16), Pisa, Italy, 4–8 April 2016; pp. 901–903.
39. Hao, X.H.; Li, Y.L.; Gu, Q. Power load data clustering algorithm based on DTW histogram. *Transducer Microsyst. Technol.* **2020**, *39*, 140–142.
40. Liu, Y.H.; Zhao, Q. Ultra-short-term power load forecasting based on cluster empirical mode decomposition of CNN-LSTM. *Power Syst. Technol.* **2021**, *45*, 4444–4451.
41. Zhu, H.S. Design and Implementation of CNN-BLSTM Speech Separation Algorithm Fused With Self-attention Mechanism. Master's Thesis, Hebei University of Science and Technology, Shijiazhuang, China, 2021.
42. Wang, J.D.; Gu, Z.C.; Ge, L.J. Load Clustering Characteristic Analysis of the Distribution Network Based on the Combined Improved Firefly Algorithm and K-means Algorithm. *J. Tianjin Univ. Sci. Technol.* **2023**, *56*, 137–147.
43. Pan, X.G. Research of Fuzzy Clustering Algorithm on Complicated Data and Feature Weight Learning Techniques. Ph.D. Thesis, Jiangnan University, Wuxi, China, 2022.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.