Ministry of Science and Higher Education of the Russian Federation
NATIONAL RESEARCH
TOMSK STATE UNIVERSITY (NR TSU)
Research and Education Center "Higher IT School" (HITs)

APPROVED BY
Head of the Main Educational Program
Professor, Doctor of Physical
and Mathematical Sciences
__________ O.A. Zmeev
«10» июня 2022

BACHELOR'S THESIS

WIND SPEED PREDICTION PERFORMANCE BASED ON MODEL DECOMPOSITION METHODS AND OPTIMIZATION ALGORITHMS

Main Educational Program
02.03.02 – Fundamental Computer Science and Information Technology
Specialization "Software Engineering"
Hu Zhichao

Bachelor's Thesis Supervisor
Associate Professor, Cand.of Science
(Physics & Mathematics)
______ Zh. N. Zenkova
«08» 06 2022

Consultant
Assistant Lecturer of Software
Engineering Department of TSU
______ D. O. Zmeev
«08» 06 2022

Written by
Student of Group
No. 971812
Zhichao Hu Zh. Hu
«06» 06 2022

Tomsk – 2022

The Ministry of Science and Higher Education of the Russian Federation
NATIONAL RESEARCH TOMSK STATE UNIVERSITY (NR TSU)
REC “Higher IT School”

APPROVED
MAP Head
Dr. Sc. (Phys.-Math.), professor
____________ O.A. Zmeev
21 февраля 20 22

THE TASK

Regarding the bachelor’s final qualification work implementation by a student
Hu Zhichao
(student’s full name)

In the field of study Fundamental Computer Science and Information Technology, specialization “Software Engineering”

1. The bachelor’s final qualification work theme (in the English and in the Russian language)
Wind speed prediction performance based on model decomposition methods and optimization algorithms

Эффективность прогнозирования скорости ветра на основе методов декомпозиции модели и алгоритмов оптимизации

2. The deadlines for the completion of task:
а) to academic office – 10.06 2022
б) to State Examination Board – 14.06 2022

3. Description:
The purpose is to explore the wind speed prediction models with the best prediction accuracy, needs for better management of wind farm.
goals and objectives of the FQW, expected results

Expected results: The error analysis of the selected model can be carried out according to the evaluation index, and the model with the smallest error and the optimal model can be obtained.

The organization that is being researched:
Tomsk State University, Digital Service Department

Final qualification work supervisor
Associate Professor, Cand. of Science (Physics & Mathematics)
(position, place of work)
____________ / Zhanna N. Zenkova
(signature) (initials, surname)

Final qualification work consultant
Assistant Lecturer of Software Engineering Department of TSU
(position, place of work)
____________ / Denis O. Zmeev
(signature) (initials, surname)

The task is accepted for execution
18.02 2022
(date)
Hu zhichao / Zhichao Hu
(signature) (initials, surname)

ABSTRACT

With the problems such as climate change caused by fossil energy, various countries have started to pay attention to renewable energy, and wind energy has become a key research object due to the problem of non-pollution. Since the accuracy of wind energy prediction mainly depends on the accuracy of wind speed prediction. Therefore, exploring and seeking methods to improve the accuracy of wind speed prediction has become the most important issue at present. In this paper, propose to use Back-Propagation Neural Networks(BPNN) and Support Vector Regression (SVR) and Kernel Based Extreme Learning Machine (KELM) as three different sets of base models and optimize them by using different machine learning algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Sparrow Search Algorithm (SSA), Sine Cosine Algorithm (SCA), and then combine them with different decomposition methods such as Empirical Mode Decomposition (EMD), Ensemble Empirical Mode Decomposition (EEMD), Variational Mode Decomposition (VMD), Wavelet Decomposition (Wavelet) to form the corresponding combined prediction models to explore which combined prediction model has higher accuracy. Based on the data from the National Center for Meteorological Sciences, the experiments show that the prediction accuracy of the combined prediction model formed by the VMD method is significantly higher than that of other combined prediction models in some models of BPNN and most of SVR, while the prediction accuracy of the combined prediction model formed by the Wavelet combination is higher in the models of KELM and part of models of SVR.

Keywords – Wind speed prediction, Model Decomposition Method, Back-Propagation Neural Networks, Support Vector Regression, Kernel Based Extreme Learning Machine, Genetic Algorithm, Particle Swarm Optimization, Sparrow Search Algorithm, Sine Cosine Algorithm.

Bachelor’s thesis 70 pages, 22 figures, 12 code samples, 5 tables, 63 sources, 1 appendix.

CONTENTS

GLOSSARY

EMD[19]– Empirical Mode Decomposition

EEMD[20] – Ensemble Empirical Mode Decomposition

VMD[21] – Variational Mode Decomposition

Wavelet[22] – Wavelet Decomposition

IMF – Intrinsic Mode Functions

BPNN[14] – Back Propagation Neural Network

SVR[15] – Support Vector Regression

SVM – Support Vector Machine

KELM[16] – Kernel Based Extreme Learning Machine

GA[12] – Genetic Algorithm

PSO[13] – Particle Swarm Optimization

SSA[17] – Sparrow Search Algorithm

SCA[18]– Sine Cosine Algorithm

MAPE – Mean Absolute Percentage Error

MSE – Mean Squared Error

MAE – Mean Absolute Error

RMSE – Root Mean Squared Error

Matlab[1] – It’s a programming and numeric computing platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models

INTRODUCTION

Since the 21st century, as the new round of technological revolution and industrial change continues to develop in depth, a new generation of technologies and applications represented by artificial intelligence has penetrated into every aspect of society, and people are enjoying the convenience and prosperity brought by these new technologies. However, climate change, caused by the use of fossil fuels, is threatening everything. According to the UN survey elevation, the temperature on earth has increased by 1.1°C in the last decade (2011-2020) compared to the late 19th century, and extreme droughts, water shortages, major fires, and sea level rise caused by climate change are threatening human life[2]. As a result, renewable energy has become a key research object for countries around the world, and renewable energy targets and support policies have spread almost all over the world, and according to the report, at least 164 countries have set renewable energy targets and 145 countries have introduced support policies[3]. And wind energy, as a non-polluting renewable energy source, has become a key research target for countries around the world, and according to IRENA, solar and wind energy account for 91% of all new installed renewable energy capacity[4]. However, the stochastic and fluctuating nature of wind energy complicates the entry of a high percentage of wind energy into the grid system. Between the accuracy of wind energy prediction results depends mainly on the accuracy of wind speed prediction. Wind speed prediction can effectively reduce the risk associated with wind-related uncertainty[5]. How to use modern machine learning techniques to improve the accuracy of wind speed prediction results and how to use machine learning algorithms to explore the optimal prediction model have become the fields that this paper wants to explore deeply.

According to the time scale of prediction, wind speed predictions can be classified into ultra-short-term predictions (mainly in minutes), short-term predictions (mainly in days), medium-term predictions (mainly in months, weeks) and long-term predictions (mainly in years)[6]. For wind speed prediction methods are mainly divided into physical and statistical methods. Physical methods use real-time meteorological conditions for prediction, however, due to the complexity of numerical meteorological models, they are not suitable for short-term and ultra-short-term wind speed prediction[7].Statistical methods are used to establish a functional relationship between historical data and wind power output by means of mathematical statistics, such as time series method[8], neural network method[9], Kalman filter method[10], support vector machine method[11], etc. However, many scholars have combined with other optimization algorithms and other techniques in order to build a better combined prediction model and thus to improve the prediction accuracy. For example, some scholars have used machine learning optimization algorithms such as genetic optimization algorithm[12] and Particle Swarm Optimization algorithm[13] to optimize the parameters of a single prediction model in order to improve the wind speed prediction accuracy.

Based on this, this paper proposes Back-Propagation Neural Networks (BPNN)[14] and Support Vector Regression(SVR)[15] and Kernel Based Extreme

Learning Machine(KELM)[16] as the three base models, and the base models are optimized by machine learning algorithms such as Genetic Algorithm (GA)[12], Particle Swarm Optimization (PSO)[13], Sparrow Search Algorithm (SSA)[17], Sine Cosine Algorithm (SCA)[18], etc. to optimize the base model, and then with different decomposition methods such as Empirical Mode Decomposition (EMD)[19], Ensemble Empirical Mode Decomposition (EEMD)[20], Variational Mode Decomposition (VMD)[21], Wavelet Decomposition (Wavelet)[22], etc., to form a combined prediction model to predict the wind speed, and to explore which combined prediction model has a higher The accuracy of the combined prediction model is higher.

The paper is organized as follows: the first part is a background introduction, the second part focuses on the theoretical knowledge of decomposition and prediction methods and the corresponding optimization algorithms, and the third part analyzes different combinatorial prediction models through experimental studies. The fourth part is introduction to the main program code. The fifth part presents the corresponding research conclusions based on the experimental data analysis. Part A of the Appendix A introduces the specific code realization of the optimization algorithm in Matlab.

1 Introduction to the methodologies

1.1 Introduction to the base model algorithms

1.1.1 Back-Propagation Neural Networks

Back-Propagation Neural Networks (BPNN) is a multilayer neural network trained according to the error back-propagation algorithm. The BPNN model has a powerful computational power and a very complex mapping capability. Based on this, the BPNN model can be trained adaptively for a large number of unstructured, non-exact laws[23].

The network structure of BPNN consists of: input layer, hidden layer and output layer. The structure is illustrated in Figure 1[24] below. The input layer is the first layer of the neural network, which takes the input information and passes the signal to the next layer. The input layer does not perform any processing on the data signal. Hidden layer: The other layers between the input layer and the output layer are called hidden layers, which usually do not receive signals directly from the outside world and do not send signals to the outside world. It is a general term for the different functional layers. Output layer: as the last layer of the neural network, it is mainly used to receive the input from the hidden layer and to output the resultant values predicted by the model, The code construction of the keras-based BPNN algorithm is shown in code sample 1[25], For example, the neural network shown in code sample 1 is built quickly by keras and contains an input layer with 7129 nodes, and a hidden layer with 128 nodes, and an output layer, which is a common binary classification model.

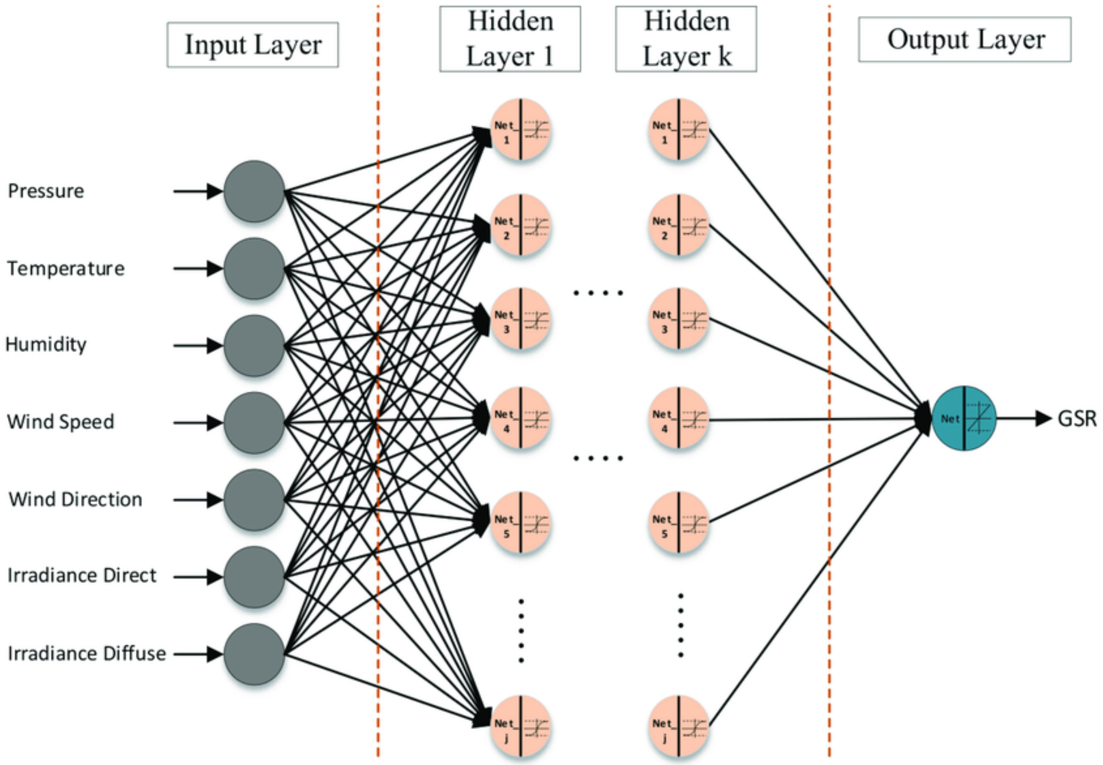


Figure 1 – Backward propagation neural network structure

```python
import keras
model = keras.Sequential()
model.add(keras.layers.Dense(7129, input_dim=7129, kernel_initializer='normal', activation='tanh'))
model.add(keras.layers.Dense(128, kernel_initializer='normal', activation='tanh'))
model.add(keras.layers.Dense(2, kernel_initializer='normal', activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
history = model.fit(x_train, y_train, epochs=6, batch_size=200, verbose=1)
```

Code sample 1 – Neural networks built on keras code

1.1.2 Support Vector Regression

Support Vector Regression are a new type of classifier based on the theoretical foundation of Support Vector Machine (SVM). The purpose of SVM[26] is to find a hyperplane so that the sample points are as far away from the hyperplane as possible. The SVR, on the other hand, wants the sample points to be as close to the hyperplane as possible. The model is optimized by minimizing the width between intervals[27]. The structure is schematically shown in the following Figure 2 and Figure 3[28].

So, it can be considered statistically that the traditional regression method is only considered correct when the regression function f(x) is exactly equal to y, when and only when this condition occurs, all kinds of complex losses need to be calculated, which makes our computation much more. On the contrary, support vector regression considers that if the regression function f(x) deviates from y within an acceptable range, this acceptable range is called the threshold (N). As long as $|f(x) - y| > N$. Then the prediction can be considered correct. The equation for the specific SVR regression model can be expressed as formula (1)[29]:

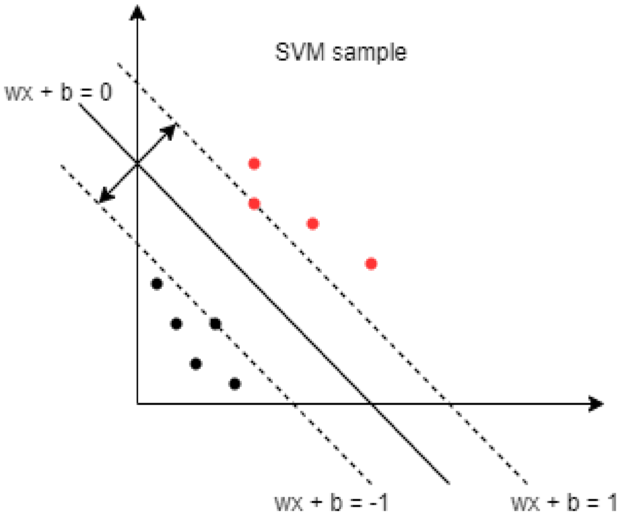


Figure 2 – Schematic diagram of SVM

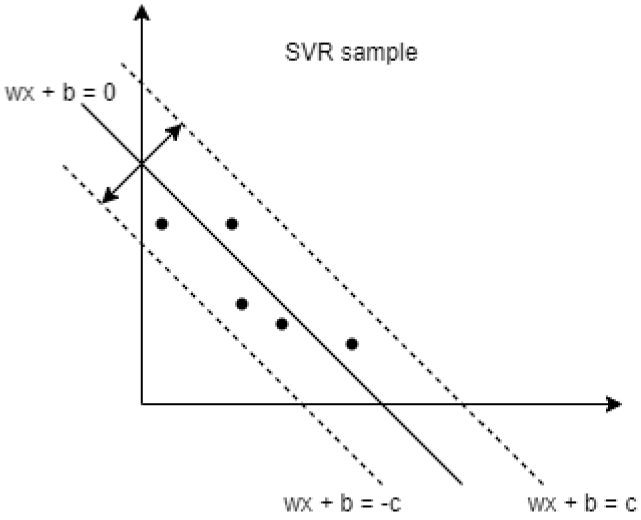


Figure 3 – Schematic diagram of SVR

$$f(x) = w^T x + b \tag{1}$$

In the formula (1), w and b are the model parameters to be determined.

1.1.3 Kernel Based Extreme Learning Machine

Kernel Based Extreme Learning Machine (KELM) is an improved learning algorithm based on the theoretical basis of Extreme Learning Machine (ELM)[30] and combined with kernel functions. By referring to the kernel function in ELM, the KELM algorithm increases the stability and robustness of the classification model on the basis of the original ELM algorithm[31], which makes the KELM algorithm have better performance to deal with more classification problems[32].

ELM is feed-forward neural network, whose objective function *f(x)* can be expressed as formula 2[33]:

$$f(x) = a(x) \times b = A \times b = L \tag{2}$$

In the formular 2, *x* is the input vector, *a(x)*, *A* is the output of the hidden layer node, *b* is the output weight, *L* is the desired output.

By introducing the regularization factor *C* and the unit matrix *I* on the basis of formula 2, the expression of the output weights is shown in formula 3[33]:

$$b = A^T(AA^T + \frac{I}{C})^{-1}L \tag{3}$$

Then, the kernel function is introduced, and the matrix equation of the kernel function can be expressed as formula 4[33]:

$$\Omega ELM = AA^T = a(x_i)a\left(x_j\right) = K(x_i,\, x_j) \tag{4}$$

In formula 4, x_i, x_j is the test input vector.

Finally, the expression can be expressed as formula 5[33]:

$$f(x) = [K(x,x_1);\ldots.;K(x,x_n)](\frac{I}{C} + \Omega ELM)^{-1}L \qquad (5)$$

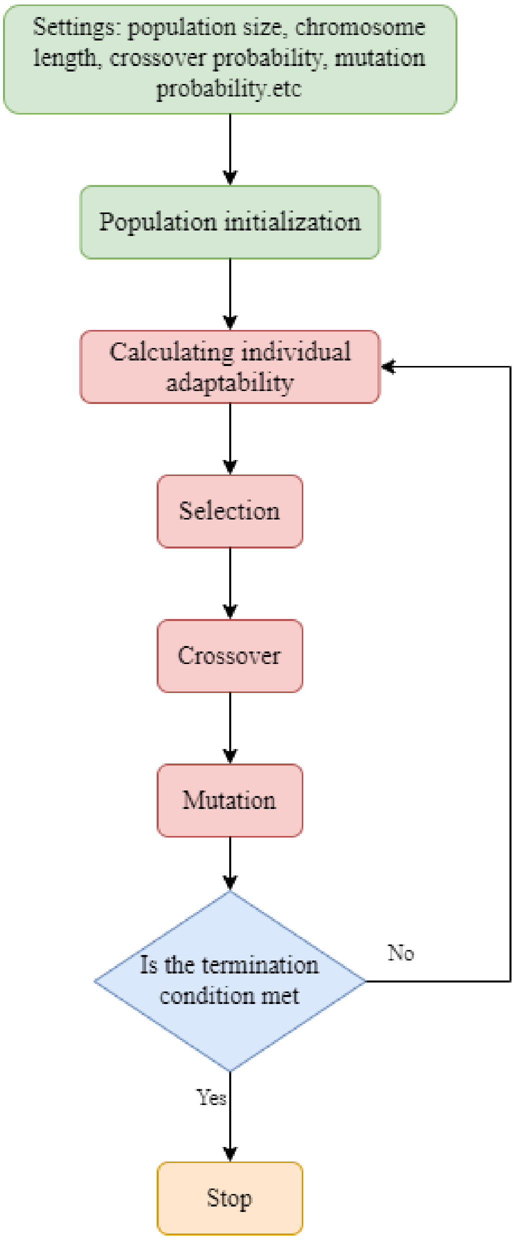
In formula 5, $(x_1, x_2, x_3, \ldots, x_n)$ is the given training sample， n indicates the sample size, *K()* is denoted as the kernel function.

1.2 Optimization algorithms

1.2.1 Genetic Algorithm (GA)

The main idea of Genetic Algorithm is derived from Darwin's theory of biological evolution of natural selection and the computational model of biological evolution of genetic mechanism, and it is a machine learning algorithm that searches for the optimal solution through the natural selection process of natural evolution of superiority[35].

The Genetic Algorithm is an efficient search for each running parameter code by randomization technique using all the individuals in the population as the object[36].The algorithm operation process is shown in Figure 4 below, which selects the optimal parameter code through three evolutionary operations: selection, crossover and mutation. Because of its own efficiency and class parallel processing and global optimality has a very obvious advantage in dealing with various nonlinear problems and solving multi-objective constrained problems[37].



Settings: population size, chromosome length, crossover probability, mutation probability.etc
Population initialization
Calculating individual adaptability
Selection
Crossover
Mutation
Is the termination condition met
No
Yes
Stop

Figure 4 – Execution process of Genetic Algorithm

The three important operations of Genetic Algorithm are selection, crossover and mutation. For the code implementation, please refer to the Genetic Algorithm in Appendix A[38].

1.2.2 Particle Swarm Optimization (PSO)

The main idea of the Particle Swarm Optimization algorithm originates from the process of foraging for food by flocks of birds and the process of collaborating with each other to complete the search for food. It is simpler than the Genetic Algorithm because there is no "crossover" or "mutation" operation[39]. The Particle Swarm Optimization algorithm is based on an iterative approach to find the global optimal solution by iteratively updating the optimal solution over and over again[40].

It has become a popular research area in optimization algorithms because of its simple principle, few adjustable parameters and easier implementation[41]. The algorithm operation process is shown in Figure 5 below:

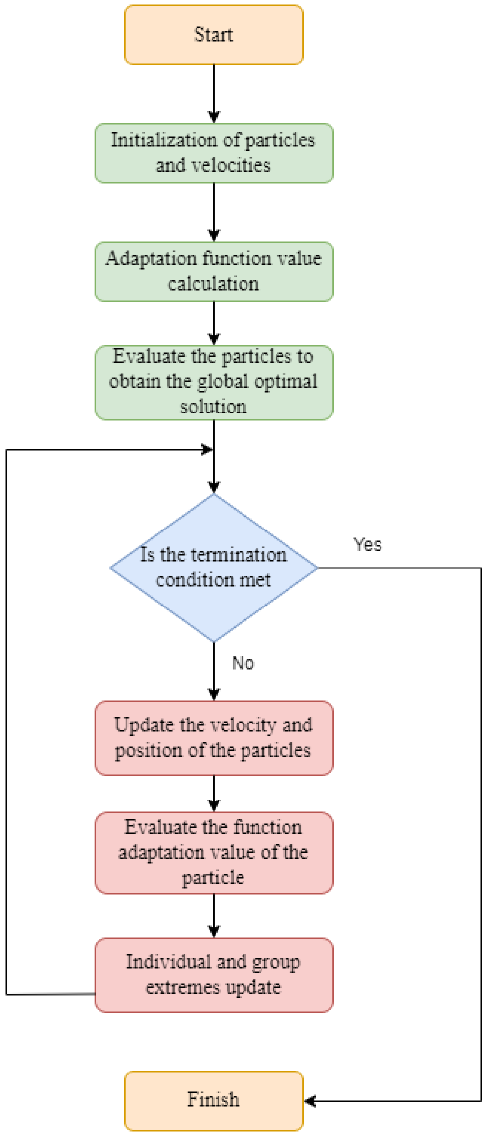

Start
Initialization of particles and velocities
Adaptation function value calculation
Evaluate the particles to obtain the global optimal solution
Is the termination condition met
Yes
No
Update the velocity and position of the particles
Evaluate the function adaptation value of the particle
Individual and group extremes update
Finish

Figure 5 – Execution process of Particle Swarm Optimization

The PSO algorithm finds the optimal solution by iteration. During each iteration, the particle updates its velocity and position by individual and population extremes. The optimal solution found by the particle itself is called the individual extremum, which can be represented by pbest. The other optimal solution found in the population is called the population extreme, which can be represented by gbest. The formula for the particle to update its own velocity and position can be represented by the following formula 6 and formula 7[42]:

Velocity:

$$v_{i+1} = w \times v_i + c_1 \times rand_1 \times (pbest_i - x_i) + c_2 \times rand_2 \times (gbest_i - x_i) \quad (6)$$

Position:

$$x_i = x_i + v_{i+1} \quad (7)$$

In the above two formulas, w is the inertia factor, generally taken as 1. c_1 and c_2 are the learning factors, generally taken as 2. $rand_1$ and $rand_2$ are random numbers between (0, 1). v_i and x_i denote the velocity and position of the particle in the i-th dimension, respectively. $pbest_i$, $gbest_i$ denote the value of the i-th dimension of the best position of a particle, the value of the i-th dimension of the best position of the whole population, respectively.

For code implementation of Particle Swarm Optimization algorithm, please refer to Particle Swarm Optimization in Appendix A.

1.2.3 Sparrow Search Algorithm (SSA)

The sparrow search optimization algorithm was proposed in 2020, and its main idea is derived from sparrow foraging behavior and anti-predation behavior, which is a new type of swarm intelligence optimization algorithm[44]. The sparrow search optimization algorithm can be considered as an improvement of the artificial swarm algorithm, and it has almost the same structure as the artificial swarm algorithm, only with some differences in the search. Its algorithm flow is mainly shown in Figure 6 below[45].

In the Sparrow Search Algorithm, in which the discoverers, especially those possessing better adaptation worthy discoverers, are given priority in the search process to obtain food. At the same time, the discoverers can also provide foraging directions for the joiners in all populations, so that the discoverers can forage in a larger area. Similarly, the joiners will always monitor the discoverers during the foraging process. If the entire sparrow population is threatened by a predator, it will immediately engage in anti-predatory behavior[46].

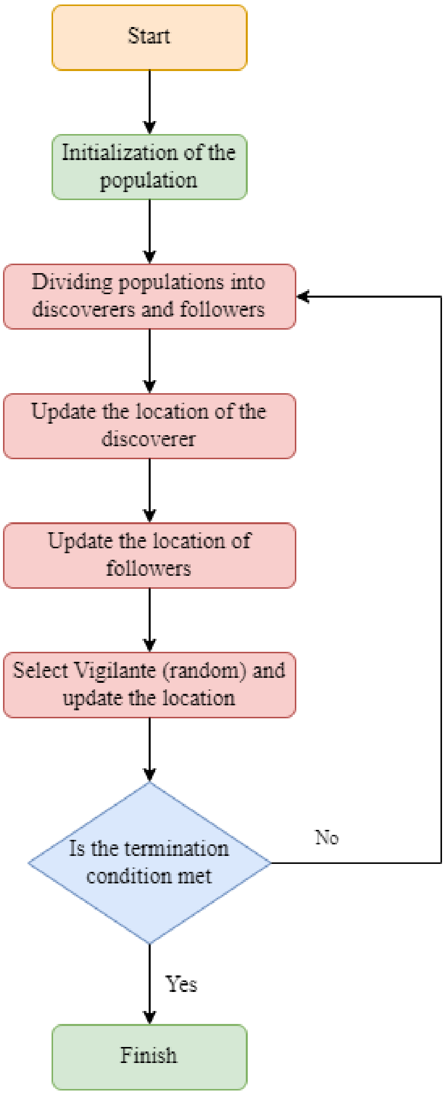

Start
Initialization of the population
Dividing populations into discoverers and followers
Update the location of the discoverer
Update the location of followers
Select Vigilante (random) and update the location
Is the termination condition met
No
Yes
Finish

Figure 6 – Flow chart of the execution process of the Sparrow Search Algorithm

For code implementation of Sparrow Search Algorithm, please refer to Sparrow Search Algorithm in Appendix A.

1.2.4 Sine Cosine Algorithm (SCA)

The Sine Cosine Algorithm is a new stochastic optimization algorithm that is an intelligent optimization algorithm proposed by Seyedali Mirjalili, Australia[47]. The algorithm works by creating multiple initial stochastic candidate solutions and then fluctuates towards the optimal solution using mathematical models of sine and cosine. The stochastic solutions are evaluated iteratively by the objective function so that different regions in the space can be used to achieve the global optimum, very efficiently avoiding the local optimum problem[48].

The sine cosine optimization algorithm is widely used in various fields of optimization problems because of its flexibility, simplicity and ease of implementation. The operation process of the sine cosine optimization algorithm can be divided into the following two phases: the exploration phase and the development phase. In the exploration phase, the initial stochastic solution is combined with the optimization algorithm to quickly search for feasible spatial regions among all the stochastic solutions. During the development phase, the random solution changes, but the rate of change is always lower than that of the exploration phase. Its formulation can be expressed by the following equation 8[49].

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times sin(r_2) \times |r_3 P_i^t - X_i^t| & r_4 < 0.5 \\ X_i^t + r_1 \times cos(r_2) \times |r_3 P_i^t - X_i^t| & r_4 > 0.5 \end{cases} \quad (8)$$

In this equation, X_i^t is the i-th dimension of the current individual and the position of the t-th generation. r_2 denotes a random number from 0 to 2π. r_3 denotes a random number between 0 to 2. r_4 denotes a random number from 0 to 1. P_i^t denotes the position of the *i-th* dimension of the optimal individual position variable at the *t-th* iteration.

The specific execution flow chart of the SCA is shown in Figure 7 below:

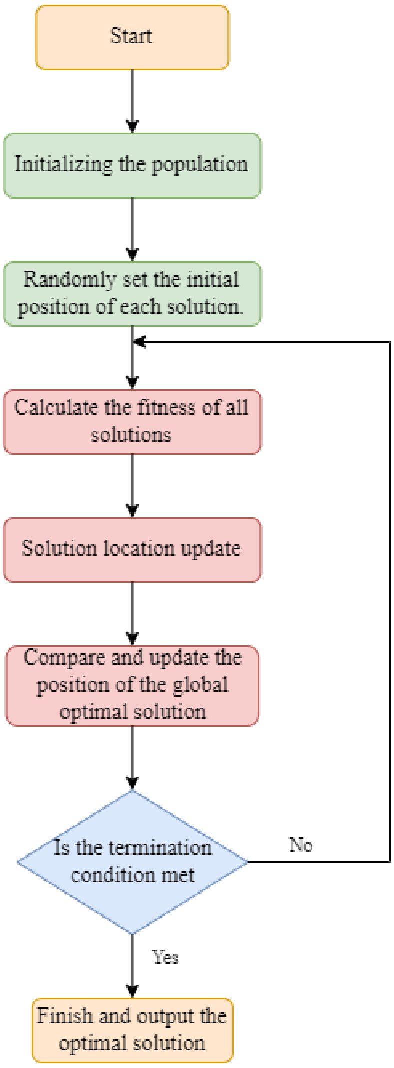

Start
Initializing the population
Randomly set the initial position of each solution.
Calculate the fitness of all solutions
Solution location update
Compare and update the position of the global optimal solution
Is the termination condition met
No
Yes
Finish and output the optimal solution

Figure 7 – Flow chart of the execution process of the Sine Cosine Algorithm

The code to initialize the random result set of the Sine Cosine Algorithm and the main loop is shown in Sine Cosine Algorithm in Appendix A.

1.3 Decomposition methods

1.3.1 Empirical Mode Decomposition (EMD)

The EMD algorithm was proposed by Norden e. Huang, an American scientist, and it can decompose a signal by virtue of the time-scale characteristics of the data itself without setting any basis function in advance[51]. EMD algorithm has a very obvious advantage in dealing with various nonlinear and non-stationary signals because of its data-driven adaptive nature[52].

EMD obtains the first-order residual quantity $r_1(t)$ by differentiating the original signal $X(t)$ and $c_1(t)$, and then replaces $r_1(t)$ with the original signal $X(t)$ for the corresponding processing, and the n-th-order modal function $c_n(t)$ and the final residual quantity $r_n(t)$ can be obtained after repeating n times. The final decomposition expression is[5]:

$$X(t) = \sum_1^n c_n(t) + r_n(t) \tag{9}$$

During the EMD method, it is very easy to generate modal mixing because of the presence of signals with similar scales in the IMF component. Also because the signal has no sample points to be considered after the beginning and the end, it can also lead to the polar packets diverging at the end points, thus leading to errors, which is often referred to as the end effect. Therefore, in order to solve these problems, the EEMD method is proposed[53].

1.3.2 Ensemble Empirical Mode Decomposition (EEMD)

The EEMD algorithm is an improved algorithm proposed by Norden e. Huang and other scientists to suppress the modal aliasing phenomenon caused by the EMD algorithm. By superimposing a Gaussian white noise auxiliary signal processing (NADA), the modal aliasing phenomenon is effectively suppressed[55].

The specific steps of the decomposition of the EEMD algorithm are shown below[5]:

1) The white noise $M_i(t)$ with normal distribution is added to the original signal $X(t)$ to obtain the new signal $X'(t)$[5]:

$$X'(t) = X(t) + M_i(t) \tag{10}$$

2) Then the obtained new signal X'(t) is decomposed by EMD, so that the IMF component and the other remaining components can be obtained[5]:

$$X'(t) = \sum_{j=1}^{n} c_j(t) + r_n(t) \tag{11}$$

3) Then repeat the execution of step 1) and step 2) N times, and then the above corresponding IMF will be averaged[5], and finally can obtain the final IMF $(C_n(t)$, after EEMD as follows:

$$C_n(t) = \frac{1}{n}\sum_{i=1}^{N} c_i(t) \tag{12}$$

1.3.3 Variational Mode Decomposition (VMD)

The VMD algorithm is an adaptive signal processing method whose core idea is to construct and solve variational problems. Unlike the methods proposed by scientists such as Norden e. Huang, the VMD algorithm aims to decompose the original signal into subsignals of different frequencies[56]. This is because the VMD algorithm considers the original signal as formed by the superposition of subsignals with different frequency dominance. Therefore, IMF is defined in the VMD algorithm as a bandwidth-constrained frequency modulation function, and the original signal is decomposed into a number of specified IMF components by constructing and solving a variational problem[57]. It is also because of this feature that the VMD algorithm can effectively avoid the phenomenon of modal aliasing and has the advantage of better decomposition accuracy for processing complex data, and it is for this reason that the VMD algorithm is often used to deal with various non-smooth and non-linear signal problems[58].

1.3.4 Wavelet Decomposition (Wavelet)

The Wavelet Decomposition algorithm can be seen as an optimization of the Fourier transform, By breaking up a continuous periodic signal into a linear combination of trigonometric signals of different frequencies, when the period of this periodic signal is infinite, this is the Fourier transform[59]. The basic equation of Fourier transform is sin function and cos function, and the basic equation of wavelet transform is wavelet function. The change made by wavelet transform lies in turning the infinitely long trigonometric function basis into a finite length wavelet basis that will decay, and different wavelets have very big differences in waveform, and similar wavelets form a wavelet family[61]. The functional representation of the wavelet transform is shown below. Due to the sparse coding property of Wavelet Decomposition, wavelet packet decomposition is commonly used to deal with various signal filtering problems and to deal with various signal noise reduction problems and various data compression problems[62].

$$\alpha = W^T f \tag{13}$$

where α denotes the wavelet coefficients obtained from the transform and W is the orthogonal matrix. f is the input signal.

2 Experimental research

2.1 Data preparation

The experimental data for this experiment were obtained from the National Center for Meteorological Sciences (NCMS), and the station located in Qingdao, Shandong Province, with a latitude and longitude position of (36.07°N, 120.33°E) and an elevation of 76 m. The specific experimental idea of this experiment is shown in Figure 8 below:

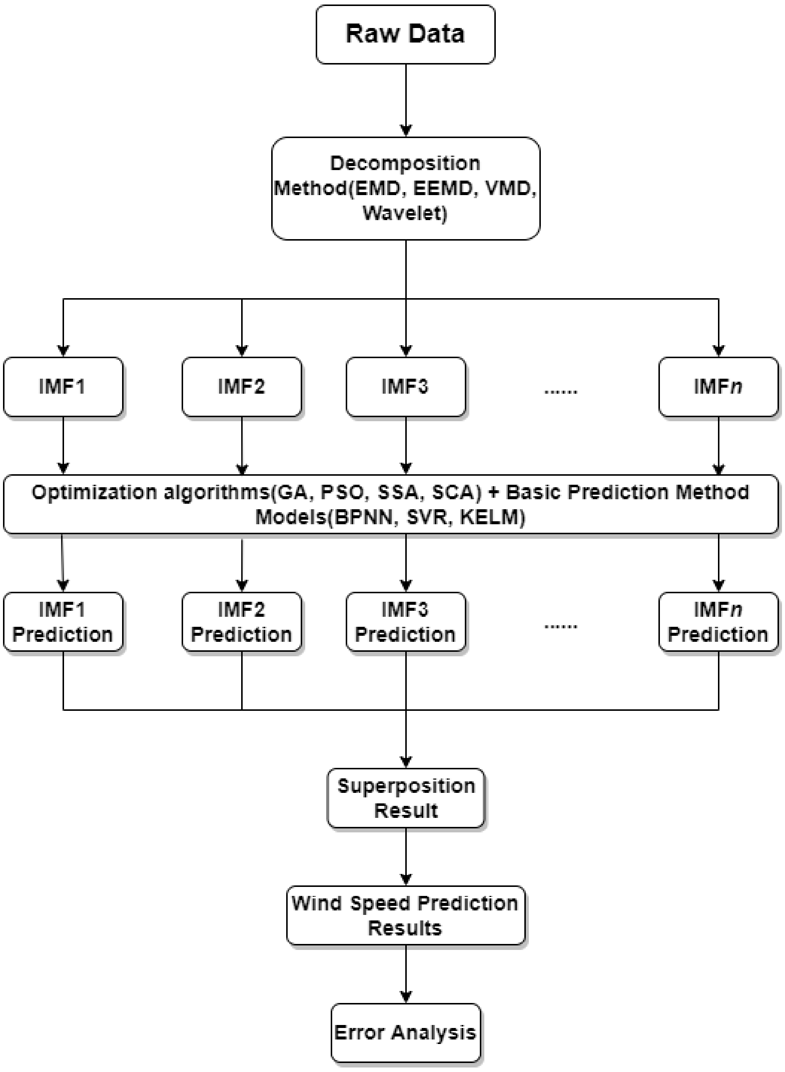

Raw Data
Decomposition Method(EMD, EEMD, VMD, Wavelet)
IMF1
IMF2
IMF3
......
IMFn
Optimization algorithms(GA, PSO, SSA, SCA) + Basic Prediction Method Models(BPNN, SVR, KELM)
IMF1 Prediction
IMF2 Prediction
IMF3 Prediction
......
IMFn Prediction
Superposition Result
Wind Speed Prediction Results
Error Analysis

Figure 8 – Flow chart of experimental idea design

2.2 Data decomposition

The sample data for this experiment has a total of 300 and is divided into two different data sets, the training set and the test set, for training and prediction. The first 80% of the sample data is used as the training set for the whole model, and the remaining 20% of the sample data is used as the test set for the whole model.

Firstly, by using different decomposition methods (EMD, EEMD, VMD, Wavelet) to decompose the original wind speed data to obtain IMF*1*, IMF*2*, IMF*3*......IMF*n*. The specific decomposition results are shown in the following Figure 9,10:

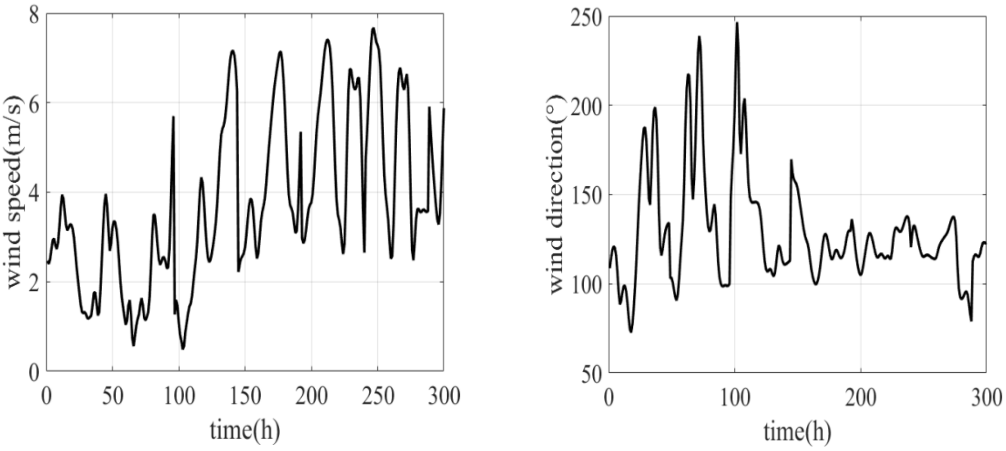


Figure 9, 10 – Raw data signal decomposition

The raw data decomposed by the EMD method is shown in the following Figure 11:

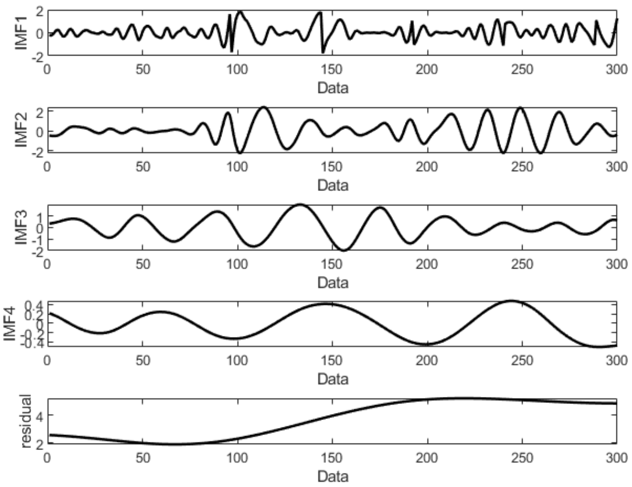


Figure 11 – Raw data signal decomposed by EMD

The raw data by the EEMD method through decomposition is shown in the following Figure 12, 13:

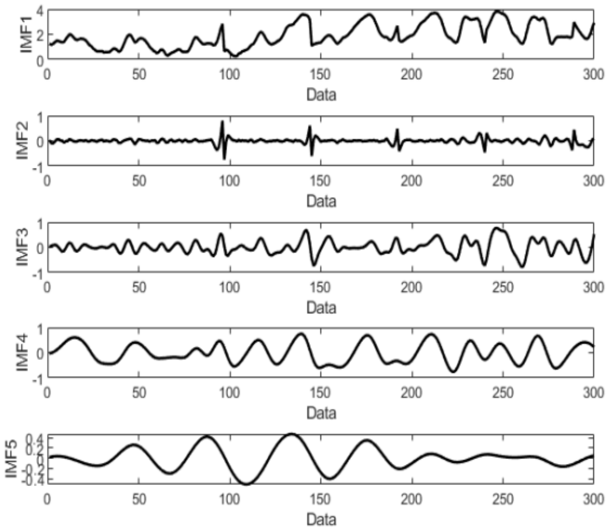


Figure 12 – Raw data signal decomposed by EEMD

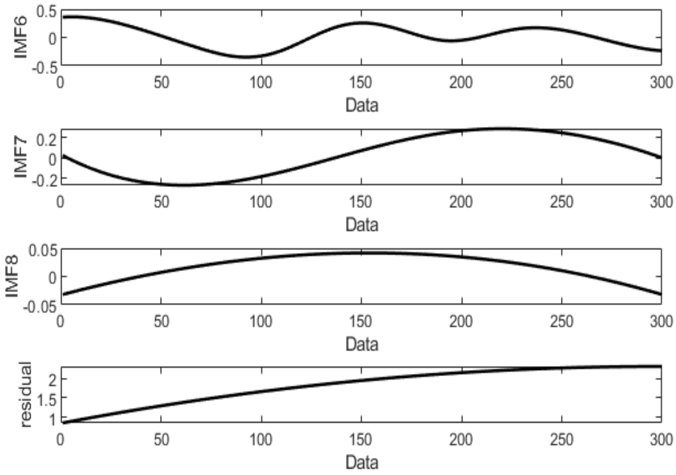


Figure 13 – Raw data signal decomposed by EEMD

The raw data decomposed by the VMD method is shown in the following Figure 14, 15:

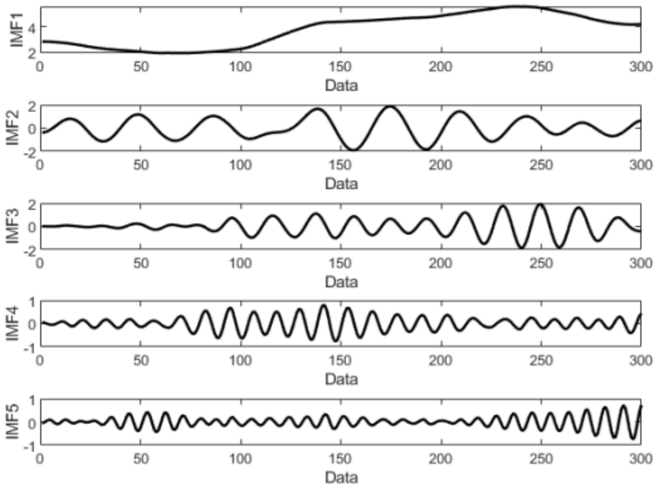


Figure 14 – Raw data signal decomposed by VMD

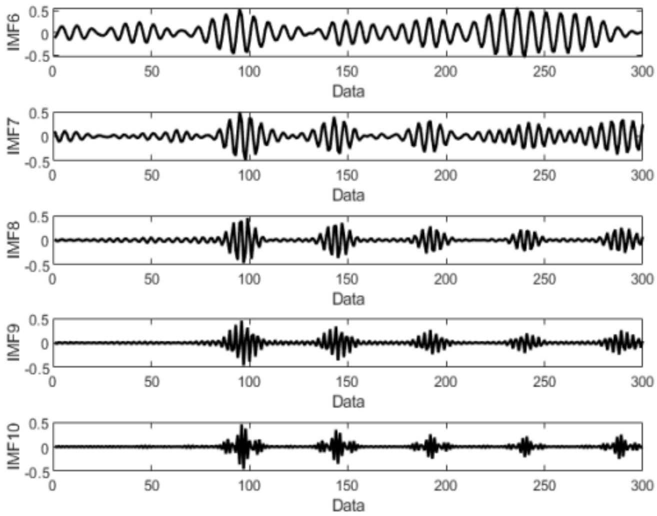


Figure 15 – Raw data signal decomposed by VMD

The raw data decomposed by the Wavelet Decomposition method is shown in the following Figure 16:

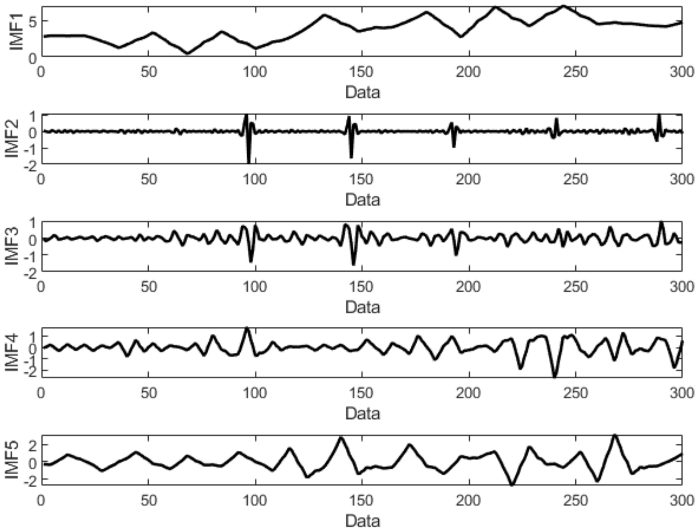


Figure 16 – Raw data signal decomposed by Wavelet

2.3 Predictive models

In this paper, BPNN, SVR and KELM are used as three groups of basic prediction models. In order to improve the accuracy of the model, four different optimization methods (GA, PSO, SSA and SCA) are used to optimize the three groups of basic models, and then they are combined with the decomposition method to obtain the following different groups of combined prediction models.

The prediction results of the combined prediction model formed by combining the EMD method with the prediction model are shown in the following Figure 17:

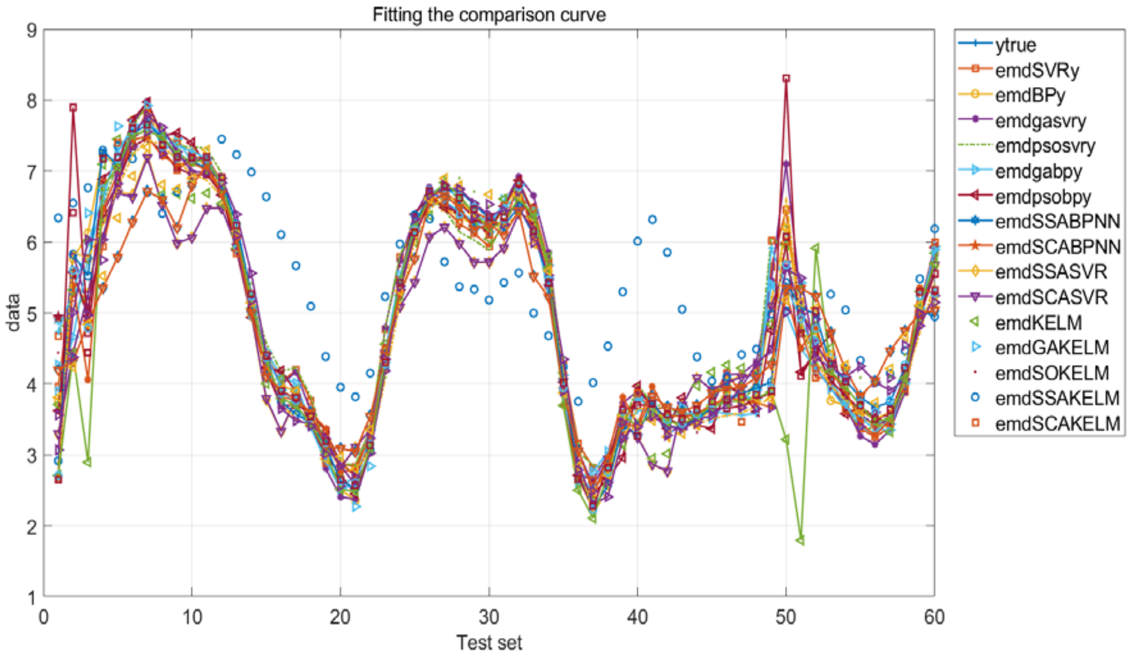


Figure 17 – Prediction diagram of EMD combined prediction model

The prediction results of the combined prediction model formed by combining the EEMD method with the prediction model are shown in the following Figure 18:

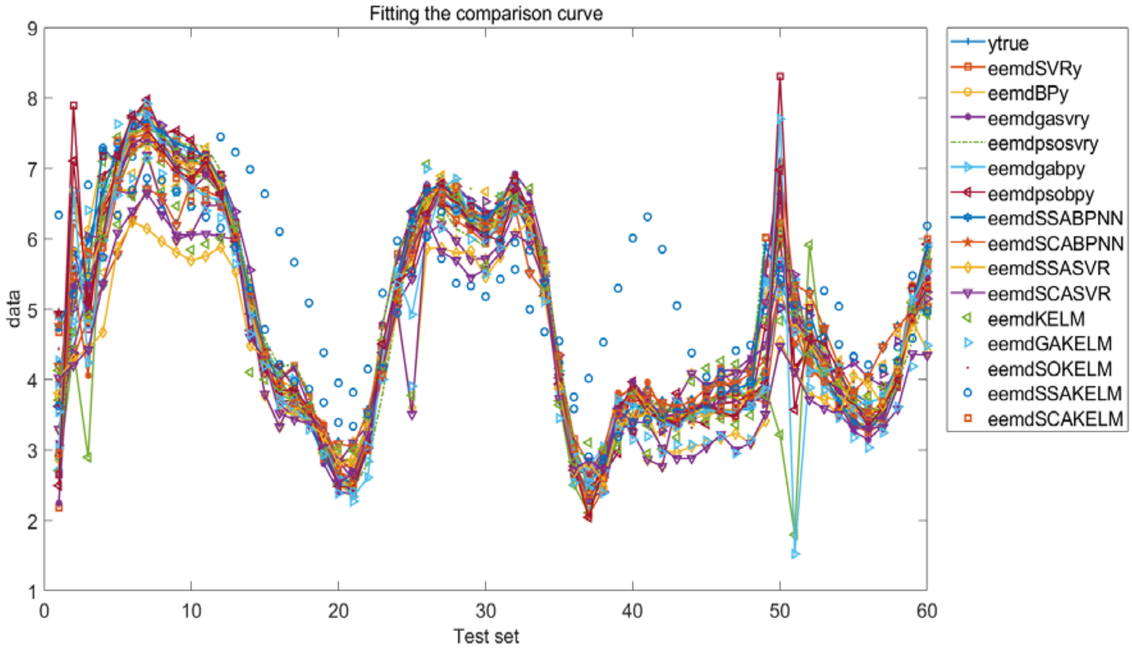


Figure 18 – Prediction diagram of EEMD combined prediction model

The prediction results of the combined prediction model formed by combining the VMD method with the prediction model are shown in the following Figure 19:

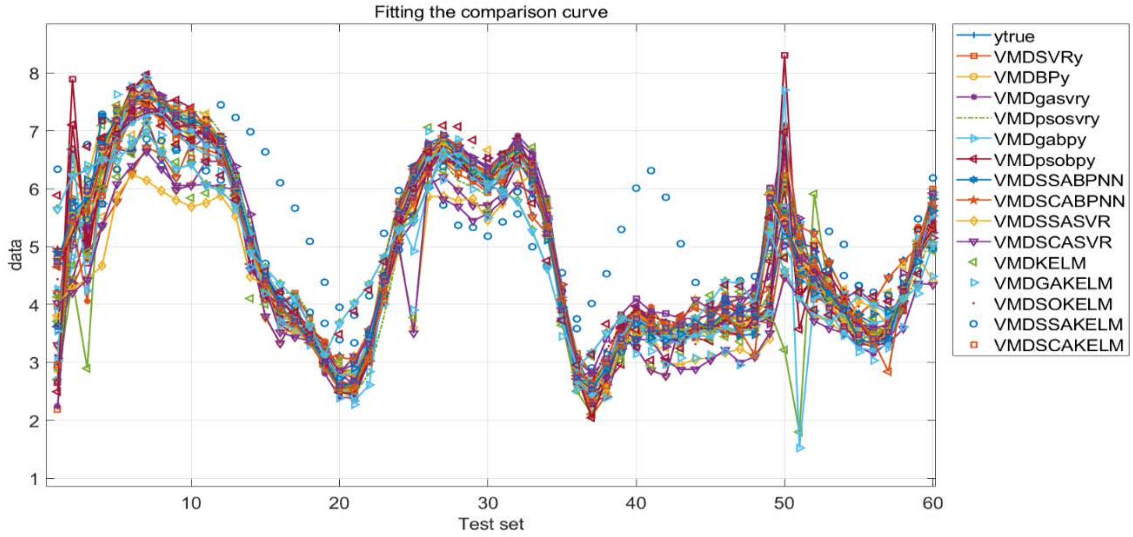


Figure 19 – Prediction diagram of VMD combined prediction model

The final prediction results of the prediction model formed by combining the Wavelet Decomposition method with the prediction model are shown in the following Figure 20:

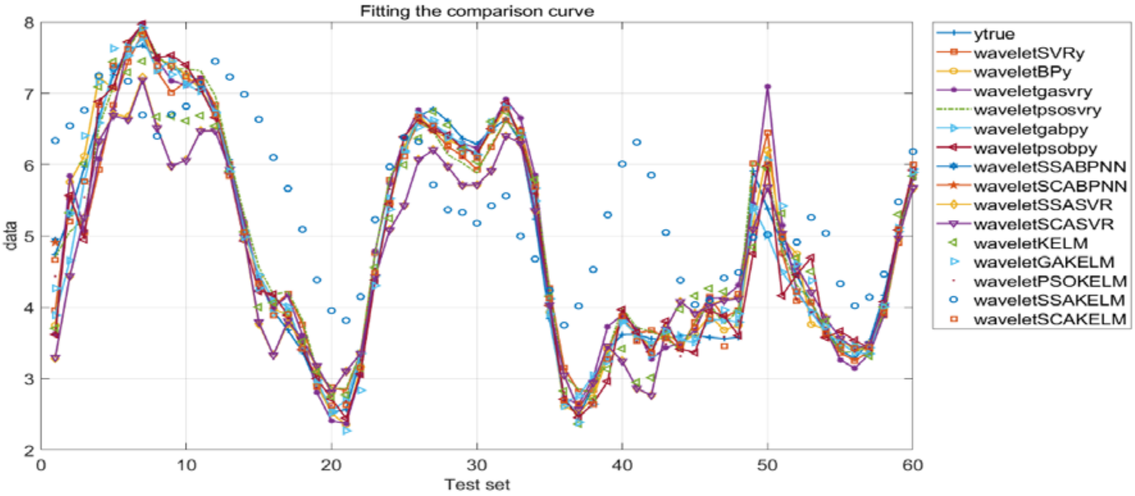


Figure 20 – Prediction diagram of Wavelet Decomposition combined prediction model

2.4 Evaluation indicators

After predicting the model, it is necessary to analyze the error of the model to measure the prediction accuracy of the model. In this paper, the prediction accuracy of the combined prediction model is analyzed by using four common evaluation indexes: mean absolute percentage error, mean square error, mean absolute error, and root mean square error. Through them, quantitative error analysis is carried out. The evaluation indexes range from [0, +∞), and when the error between the predicted value and the true value is infinitely closer to 0, it means that the prediction accuracy of the model is closer to the perfect combination prediction model. Among them, the definitions and formulas of the four evaluation indicators are shown below:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{15}$$

$$MAE = \sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{16}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{17}$$

In the formula: *n* represents the number of test samples, y_i and $\widehat{y_i}$ represent the true and predicted values at moment *i-th*.

2.5 Error analysis

After the combination prediction model predicts the original data, it is necessary to analyze and compare the prediction results of the combination prediction model according to the evaluation index to obtain the optimal combination prediction model.

The error table of the combined prediction model formed by combining the EMD method with the basic prediction model is shown in the following Table 1:

Table 1 Error analysis of EMD combined prediction method

Model\Index	MAPE(%)	MSE(m/s)	MAE(m/s)	RMSE(m/s)
emdBPNN	5.44	0.1784	0.2469	0.4223
emdGABPNN	6.11	0.2244	0.2863	0.4738
emdPSOBPNN	9.20	0.4299	0.4364	0.6557
emdSSABPNN	3.20	0.032	0.1519	0.1519
emdSCABPNN	4.29	0.0429	0.2032	0.2032
emdSVR	6.83	0.4635	0.3345	0.6808
emdGASVR	5.44	0.2561	0.2631	0.5061
emdPSOSVR	6.42	0.2408	0.3000	0.4907
emdSSASVR	8.87	0.0887	0.5958	0.5958
emdSCASVR	8.87	0.0887	0.5961	0.5961
emdKELM	7.02	0.0702	0.5061	0.5061
emdGAKELM	5.36	0.0536	0.5004	0.5004
emdPSOKELM	7.06	0.0706	0.4474	0.4474
emdSSAKELM	5.07	0.0507	0.4377	0.4377
emdSCAKELM	6.87	0.0687	0.4680	0.4680

The error table of the combined prediction model formed by combining the EEMD method with the basic prediction model is shown in the following Table 2:

Table 2 Error analysis of EEMD combined prediction method

Model\Index	**MAPE(%)**	**MSE(m/s)**	**MAE(m/s)**	**RMSE(m/s)**
eemdBPNN	5.75	0.2335	0.2752	0.4832
eemdGABPNN	6.27	0.2964	0.2938	0.5445
eemdPSOBPNN	7.00	0.3480	0.3419	0.5899
eemdSSABPNN	2.83	0.0283	0.1343	0.1343
eemdSCABPNN	0.54	0.0054	0.0255	0.0255
eemdSVR	6.65	0.2903	0.3245	0.5388
eemdGASVR	6.25	0.2527	0.3104	0.5027
eemdPSOSVR	6.14	0.2386	0.2926	0.4456
eemdSSASVR	9.34	0.0934	0.7235	0.7235
eemdSCASVR	12.48	0.1248	0.8973	0.8973
eemdKELM	8.19	0.0819	0.626	0.626
eemdGAKELM	6.06	0.0606	0.4897	0.4897
eemdPSOKELM	5.54	0.0554	0.4013	0.4013

Continuation of Table 2

Model\Index	**MAPE(%)**	**MSE(m/s)**	**MAE(m/s)**	**RMSE(m/s)**
eemdSSAKELM	11.08	0.1108	0.6324	0.6324
eemdSCAKELM	5.83	0.0583	0.4065	0.4065

The error table of the combined prediction model formed by combining the VMD method with the basic prediction model is shown in the following Table 3:

Table 3 Error analysis of VMD combined prediction method

Model\Index	**MAPE(%)**	**MSE(m/s)**	**MAE(m/s)**	**RMSE(m/s)**
VMDBPNN	1.89	0.0127	0.0820	0.1126
VMDGABPNN	1.99	0.0110	0.0852	0.1048
VMDPSOBPNN	2.17	0.0158	0.0946	0.1256
VMDSSABPNN	2.54	0.0254	0.1204	0.1204
VMDSCABPNN	2.33	0.0233	0.1106	0.1106
VMDSVR	5.08	0.0597	0.2058	0.2443
VMDGASVR	4.95	0.0759	0.2291	0.2755
VMDPSOSVR	3.04	0.1986	0.1419	0.1737
VMDSSASVR	7.25	0.1225	0.642	0.642

Continuation of Table 3

Model\Index	MAPE(%)	MSE(m/s)	MAE(m/s)	RMSE(m/s)
VMDSCASVR	7.27	0.1227	0.6417	0.6417
VMDKELM	10.62	0.0702	0.5061	0.5061
VMDGAKELM	5.36	0.0536	0.5004	0.5004
VMDPSOKELM	7.06	0.0706	0.4474	0.4474
VMDSSAKELM	5.07	0.0507	0.4377	0.4377
VMDSCAKELM	6.87	0.0687	04.68	0.468

The error table for the combined prediction model formed by combining the Wavelet Decomposition method with the basic prediction model is shown in the following Table 4:

Table 4 Error analysis of Wavelet combined prediction method

Model\Index	MAPE(%)	MSE(m/s)	MAE(m/s)	RMSE(m/s)
waveletBPNN	3.57	0.0357	0.2555	0.2555
waveletGABPNN	3.47	0.0347	0.2148	0.2148
waveletPSOBPNN	5.08	0.0508	0.3403	0.3403
waveletSSABPNN	4.17	0.0417	0.1978	0.1978

Continuation of Table 4

Model\Index	MAPE(%)	MSE(m/s)	MAE(m/s)	RMSE(m/s)
waveletSCABPNN	3.61	0.0361	0.1711	0.1711
waveletSVR	5.57	0.057	0.3358	0.3358
waveletGASVR	5.85	0.0585	0.3979	0.3979
waveletPSOSVR	4.80	0.048	0.2651	0.2651
waveletSSASVR	8.67	0.0867	0.5454	0.5454
waveletSCASVR	8.79	0.0873	0.5463	0.5463
waveletKELM	6.65	0.0665	0.3883	0.3883
waveletGAKELM	4.16	0.0416	0.246	0.246
waveletPSOKELM	3.42	0.0342	0.1961	0.1961
waveletSSAKELM	5.08	0.0508	0.1993	0.1993
waveletSCAKELM	3.3	0.032	0.0906	0.0906

In order to make a more intuitive comparison and analysis of the prediction error results of these 60 different sets of combined prediction models, the Table 5 obtained by collation is shown below:

Table 5 Error analysis of all combined prediction method

EMD/EEMD/VMD/Wavelet	MAPE(%)	MSE(m/s)	MAE(m/s)	RMSE(m/s)
BPNN	5.44/5.75/1.89/3.57	0.1784/0.2335/0.0127/0.0357	0.2469/0.2752/0.0820/0.2555	0.4223/0.4832/0.1126/0.2555
GABPNN	6.11/6.27/1.99/3.47	0.2244/0.2964/0.0110/0.0347	0.2863/0.2938/0.0852/0.2148	0.4738/0.5445/0.1048/0.2148
PSOBPNN	9.20/7.00/2.17/5.08	0.4299/0.3480/0.0158/0.0508	0.4364/0.3419/0.0946/0.3403	0.6557/0.5899/0.1256/0.3403
SSABPNN	3.20/2.83/2.54/4.17	0.032/0.0283/0.0254/0.0417	0.1519/0.1343/0.1204/0.1978	0.1519/0.1343/0.1204/0.1978
SCABPNN	4.29/0.54/2.33/3.61	0.0429/0.0054/0.0233/0.0361	0.2032/0.0255/0.1106/0.1711	0.2032/0.0255/0.1106/0.1711
SVR	6.83/6.65/5.08/5.57	0.4635/0.2903/0.0597/0.057	0.3345/0.3245/0.2058/0.3358	0.6808/0.5388/0.2443/0.3358
GASVR	5.44/6.25/4.95/5.85	0.2561/0.2527/0.0759/0.0585	0.2631/0.3104/0.2291/0.3979	0.5061/0.5027/0.2755/0.3979
PSOSVR	6.42/6.14/3.04/4.80	0.2408/0.2386/0.1986/0.048	0.3000/0.2926/0.1419/0.2651	0.4907/0.4456/0.1737/0.2651
SSASVR	8.87/9.34/7.25/8.67	0.0887/0.0934/0.1225/0.0867	0.5958/0.7235/0.642/0.5454	0.5958/0.7235/0.642/0.5454
SCASVR	8.87/12.48/7.27/8.79	0.0887/0.1248/0.1227/0.0873	0.5961/0.8973/0.6417/0.5463	0.5961/0.8973/0.6417/0.5463
KELM	7.02/8.19/10.62/6.65	0.0702/0.0819/0.0702/0.0665	0.5061/0.626/0.5061/0.3883	0.5061/0.626/0.5061/0.3883
GAKELM	5.36/6.06/5.36/4.16	0.0536/0.0606/0.0536/0.0416	0.5004/0.4897/0.5004/0.246	0.5004/0.4897/0.5004/0.246
PSOKELM	7.06/5.54/7.06/3.42	0.0706/0.0554/0.0706/0.0342	0.4474/0.4013/0.4474/0.1961	0.4474/0.4013/0.4474/0.1961
SSAKELM	5.07/11.08/5.07/5.08	0.0507/0.1108/0.0507/0.0508	0.4377/0.6324/0.4377/0.1993	0.4377/0.6324/0.4377/0.1993
SCAKELM	6.87/5.83/6.87/3.3	0.0687/0.0583/0.0687/0.032	0.468/0.4065/0.468/0.0906	0.468/0.4065/0.468/0.0906

Through the analysis of the above experimental data, the following conclusions can be obtained:

1) Among the combined prediction models formed by different decomposition methods, through the overall comparison, it can be found that the error of the combined prediction model formed by the combination of VMD methods is much smaller in BPNN and SVR models, indicating that in BPNN and most SVR (SVR, GASVR, PSOSVR) models, the combined prediction model formed by VMD has better prediction effect. The research also shows that the

combined prediction model formed by VMD overcomes the modal confusion caused by EMD method and solves the residual noise caused by EEMD.

2) In contrast, the combined prediction model formed by Wavelet presents a smaller experimental error in comparison with the KELM prediction model and part of SVR (SCASVR, SSASVR), indicating that the combined prediction model formed by Wavelet has a better prediction effect in the model of KELM. It also shows that Wavelet is also very good at overcoming the problems arising from the EMD method and the EEMD method.

3) Through the overall vertical comparison of the models, it can be found that although there are some deviations in different models, in general, the prediction model using SCA optimization algorithm shows more accurate prediction ability in BPNN prediction model. In the SVR prediction model, the prediction model using PSO optimization algorithm generally has more accurate prediction ability. Similarly, in the KELM model, the prediction model optimized by SCA optimization algorithm shows a more accurate prediction.

4) Through experimental analysis, it can be concluded that it is very undesirable to simply infer that a method must be better in theory, so as to ignore the actual situation.

INTRODUCTION OF CODE PROGRAM

In this section, the main program of the program as well as the drawing program will be introduced. The main programming tool used is Matlab.

In the main program, different decomposition models and different prediction models are selected for prediction, and then they are combined into a combined prediction model. Then the prediction accuracy is compared according to the evaluation index. The main program mainly includes three parts: parameter initialization, data decomposition and model selection.

The program code for the initialization of the parameters is shown below:

```
%% Parameter initialization
interval = 1;
start_train = 1;
lag = 4;
data = wave;
end_train = floor(0.8*length(data));
[row col]   = size(data);
Y = [];
mape = [];
dstat = [];
rmse = [];
```

Code Sample 8 – Parameter initialization in the main program

After initializing the parameters, a choice of different decomposition methods needs to be made. The procedure for data decomposition is shown below. The decomposition method model that can be chosen is indicated after %.

```
   %% Data decomposition
decomp_type = 'VMD'; %emd,eemd,VMD，wavelet
data_sample = data;
[Comp] = tm_decomposition_method(decomp_type,data_sample);
```

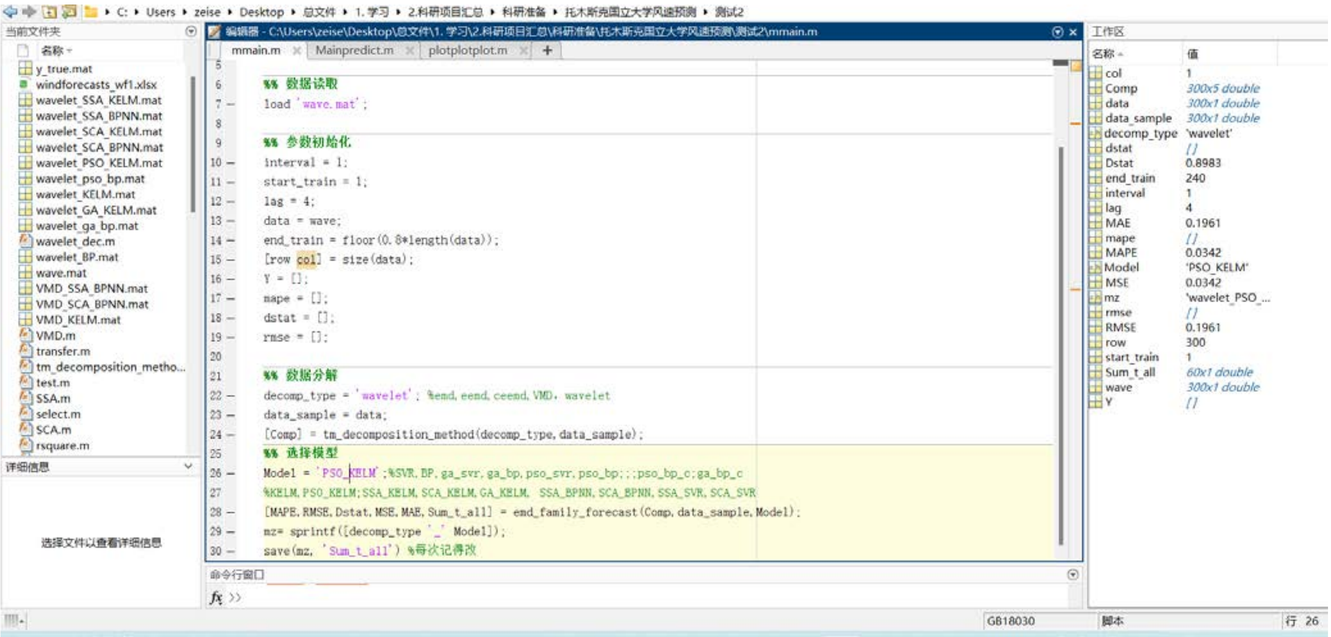
Code Sample 9 – Data decomposition in the main program.

Then select different prediction models and combine them with decomposition methods to form a combined prediction model to predict and analyze data.The code is shown below. where the model's after % indicate the predictor models that can be selected.

```
%% Selecting a predictive model
Model = 'SVR';%SVR,BP,ga_svr,ga_bp,pso_svr,pso_bp;;;
%KELM,PSO_KELM;SSA_KELM,SCA_KELM,GA_KELM, SSA_BPNN,SCA_BPNN,SSA_SVR,SCA_SVR
[MAPE,RMSE,Dstat,MSE,MAE,Sum_t_all]
   = emd_family_prediction(Comp,data_sample,Model);
mz= sprintf([decomp_type '_' Model]);
save(mz, 'Sum_t_all')
```

Code Sample 10 – Prediction model selection in the main program.

Then, by running the main program, the program will analyze the combined prediction model around the 4 set evaluation indicators. Then it will show the analysis result of the program on the right side. It will also load the data from the run into the taskbar on the left. This is shown in the Figure 21 below:



```
%% 数据读取
load 'wave.mat';

%% 参数初始化
interval = 1;
start_train = 1;
lag = 4;
data = wave;
end_train = floor(0.8*length(data));
[row col] = size(data);
Y = [];
mape = [];
dstat = [];
rmse = [];

%% 数据分解
decomp_type = 'wavelet'; %emd, eemd, ceemd, VMD, wavelet
data_sample = data;
[Comp] = tm_decomposition_method(decomp_type, data_sample);
%% 选择模型
Model = 'PSO_KELM';%SVR, BP, ga_svr, ga_bp, pso_svr, pso_bp;;;pso_bp_c;ga_bp_c
%KELM, PSO_KELM;SSA_KELM, SCA_KELM, GA_KELM,  SSA_BPNN, SCA_BPNN, SSA_SVR, SCA_SVR
[MAPE, RMSE, Dstat, MSE, MAE, Sum_t_all] = emd_family_forecast(Comp, data_sample, Model);
mz= sprintf([decomp_type '_' Model]);
save(mz, 'Sum_t_all') %每次记得改
```

名称	值
col	1
Comp	300x5 double
data	300x1 double
data_sample	300x1 double
decomp_type	'wavelet'
dstat	[]
Dstat	0.8983
end_train	240
interval	1
lag	4
MAE	0.1961
mape	[]
MAPE	0.0342
Model	'PSO_KELM'
MSE	0.0342
mz	'wavelet_PSO_...
rmse	[]
RMSE	0.1961
row	300
start_train	1
Sum_t_all	60x1 double
wave	300x1 double
Y	[]

Figure 21 – The result of the main program

After all the combined prediction models are run, the analysis diagram of the combined prediction model can be drawn according to the running results. The drawing code is divided into two main parts. The first part loads the data for the run, and the second part is the plotting function. Taking VMD combination prediction method as an example.

The code to load the data is shown below:

```
clc
clear all
%%
load y_true.mat
y_true = s;
load VMD_SVR.mat
VMD_SVR = Sum_t_all;
load VMD_BP.mat
VMD_BP = Sum_t_all;
...
```

Code Sample 11 – Loading data

The analysis diagram code for drawing the combined prediction model is shown below:

```
   %%
plot(y_true,'-+','LineWidth',1.5);
hold on
plot(VMD_SVR,'-s','LineWidth',1.5);
hold on
plot(VMD_BP,'-o','LineWidth',1.5);
hold on
...
...
grid
xlabel('Test set')
ylabel('data')
title('Fitting the comparison curve')
legend('ytrue','VMDSVRy','VMDBPy','VMDgasvry','VMDpsosvry','VMDgabpy','VMDpsobpy','VMDSSABPNN','VMDSCABPNN','VMDSSASVR','VMDSCASVR','VMDKELM','VMDGAKELM','VMDSOKELM','VMDSSAKELM','VMDSCAKELM','Location','SouthEast');
```

Code Sample 12 – Plotting of combined predictive models

Then, by running the plotting function, the following analysis diagram of the combined prediction model (Figure 22) is obtained.

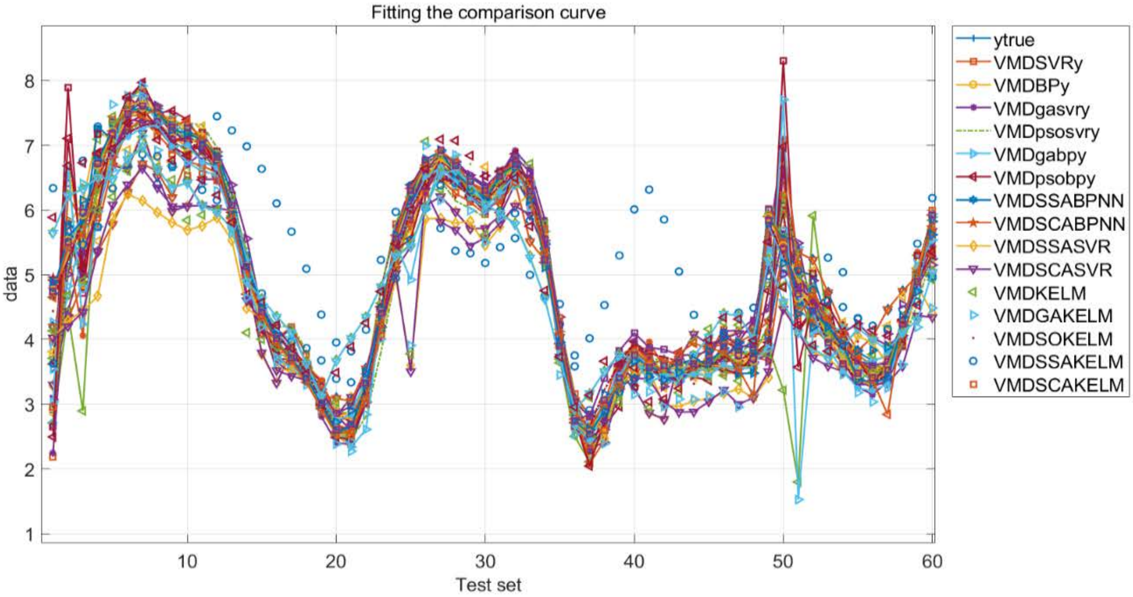


Figure 22 – Prediction diagram sample of VMD combined prediction model

CONCLUSION

In this paper, BPNN, SVR and KELM are used as three groups of basic prediction models. GA, PSO, SSA, SCA and other optimization algorithms are used to optimize the basic prediction model, and then four different decomposition methods are used to form different combined prediction models, and each combined prediction model is compared and analyzed.

The analysis of experimental data shows that

- The combined prediction model consisting of VMD method has higher prediction accuracy in the BPNN and most of SVR models compared with the combined prediction model formed by other decomposition methods. The accuracy of the combined prediction model formed by the Wavelet Decomposition method in the KELM model and part of SVR model is higher. It also shows that both the VMD method and the Wavelet Decomposition method can overcome the modal confounding problem generated by the EMD and the noise residual problem generated by the EEMD.
- Through the overall vertical comparison of the models, it can be found that although there are some deviations in different models, in general, the prediction model using SCA optimization algorithm shows more accurate prediction ability in BPNN prediction model. In the SVR prediction model, the prediction model using PSO optimization algorithm generally has more accurate prediction ability.

Similarly, in the KELM model, the prediction model optimized by SCA optimization algorithm shows a more accurate prediction.

- Through experimental analysis, it can be concluded that it is very undesirable to infer that a certain method is better only through theoretical analysis, while ignoring the actual situation. In the selection of prediction methods, more appropriate combinations should be selected according to different actual conditions to improve the prediction accuracy.

- Prospects and applications: This combined prediction model can be used not only for wind speed prediction, but also for other predictions, such as financial risk prediction models, etc.

LITERATURE

1. Matlab // Matlab website. – [N.p.], – URL: https://www.mathworks.com/help/pdf_doc/matlab/rn.pdf (access date: 10.09.2021).

2. Renewable Energy Policy Network for the 21st Century (REN21),Renewables 2015 Global Status Report // Annual Reporting on Renewables: Ten Years of Excellence (Paris, REN21 Secretariat, 2015), p. 18. Available. [N.p.], 2015. URL: www.ren21.net/gsr. (access date: 10.10.2021)

3. International Energy Agency (IEA) // Medium Term Energy Efficiency,Market Report (Paris, Organisation for Economic Co-operation and Development (OECD) and IEA, (2015), p. 86. Available. [Paris], 2015. URL: http://www.iea.org/publications/freepublications/publication/MediumTermEnergyefficiencyMarketReport2015.pdf. (access date: 10.10.2021)

4. IRENA. – [N.p.], URL: https://www.irena.org/ (access date: 11.10.2021)

5. Z. Hu, R. Zhang, Z. Zenkova and Y. Wang, "Wind Speed Prediction Performance Based on Modal Decomposition Method," // 2020 2nd International Conference on Information Technology and Computer Application (ITCA), 2020, pp. 736-741, doi:10.1109/ITCA52113.2020.00158.

6. Wang J,Li WD. Ultra-short-term wind speed prediction based on CEEMD and GWO[J] // Power System Protection and Control,2018,46(09):69-74.
7. LING Jin, Mao Meiqin, Li Fugen, Zhang Enming. Study on the correlation method of short-term wind speed prediction and its application[J] // Journal of Hefei University of Technology (Natural Science Edition),2017,40(11):1502-1506.
8. LYDIA M, SURESH K S, IMMANUEL S A, et al. Linear and non-linear autoregressive models for short-term wind speed predictioning[J] // Energy Conversion and Management, 2016, 112: 115-124.
9. LI Zhi, YE Lin, ZHAO Yongning. Short-term wind power prediction based on extreme learning machine with error correction[J] // Protection and Control of Modern Power Systems, 2016, 1(1): 9-16. DOI: 10.1186/s41601-016-0016-y.
10. XIU Chunbo, REN Xiao, LI Yanqing, et al. Short-term prediction method of wind speed series based on Kalman filtering fusion[J] //
11. Transactions of China Electrotechnical Society, 2014, 29(2): 253-259.
12. MOHANDES M A, HALAWANI T O, REHMAN S, et al. Support vector machine for wind speed prediction[J] // Renewable Energy, 2004, 29(6): 939-947.

13. AN Xiaojuan, GONG Renxi, ZHANG Qianfeng. Application of optimization SVM based on improved Genetic Algorithm in short-time wind speed prediction[J] // Power System Protection and Control, 2016, 44(9):38-42.

14. WU Zhongqiang, JIA Wenjing, WU Changhan, et al. Short-term wind speed predictioning based on PSO-BSNN[J] // Power System Protection and Control, 2015, 43(15):36-41.

15. Karsoliya S. Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture[J] // International Journal of Engineering Trends and Technology, 2012, 3(6): 714-717.

16. Balabin R M, Lomakina E I. Support vector machine regression (SVR/LS-SVM)—an alternative to neural networks (ANN) for analytical chemistry Comparison of nonlinear methods on near infrared (NIR) spectroscopy data[J] // Analyst, 2011, 136(8): 1703-1712.

17. Lu H, Du B, Liu J, et al. A kernel extreme learning machine algorithm based on improved particle swam optimization[J] // Memetic Computing, 2017, 9(2): 121-128.

18. Xue J, Shen B. A novel swarm intelligence optimization approach: Sparrow Search Algorithm[J] // Systems Science & Control Engineering, 2020, 8(1): 22-34.

19. Mirjalili S. SCA: a Sine Cosine Algorithm for solving optimization problems[J] // Knowledge-based systems, 2016, 96: 120-133.

20. Rilling G, Flandrin P, Goncalves P. On empirical mode decomposition and its algorithms[C] // IEEE-EURASIP workshop on nonlinear signal and image processing. Grado: IEEER, 2003, 3(3): 8-11.

21. Wu Z, Huang N E. Ensemble empirical mode decomposition: a noise-assisted data analysis method[J] // Advances in adaptive data analysis, 2009, 1(01): 1-41.

22. ur Rehman N, Aftab H. Multivariate variational mode decomposition[J] // IEEE Transactions on signal processing, 2019, 67(23): 6039-6052.

23. Abramovich F, Silverman B W. Wavelet decomposition approaches to statistical inverse problems[J] // Biometrika, 1998, 85(1): 115-129.

24. Qu Z, Mao W, Zhang K, et al. Multi-step wind speed predictioning based on a hybrid decomposition technique and an improved back-propagation neural network[J] // Renewable energy, 2019, 133: 919-929.

25. Aljanad, Ahmed & Tan, Nadia & Agelidis, Vassilios & Shareef, Hussain. (2021). Neural Network Approach for Global Solar Irradiance Prediction at Extremely Short-Time-Intervals Using Particle Swarm Optimization Algorithm. // Energies. 14. 1-20. 10.3390/en14041213.

26. Jianshu // Combing of BP neural networks. [N.p.], 2019. URL: https://www.jianshu.com/p/9037890c9b65 (access date: 12.12.2021)

27. Wang H, Hu D. Comparison of SVM and LS-SVM for regression[C] // 2005 International Conference on Neural Networks and Brain. IEEE, 2005, 1: 279-283.

28. Abo-Khalil A G, Lee D C. MPPT control of wind generation systems based on estimated wind speed using SVR[J] // IEEE transactions on Industrial Electronics, 2008, 55(3): 1489-1490.

29. Ren Y, Suganthan P N, Srikanth N. A novel empirical mode decomposition with support vector regression for wind speed predictioning[J] // IEEE transactions on neural networks and learning systems, 2014, 27(8): 1793-1798.

30. ZHANG Lin,WANG Ting-Hua,ZHOU Hui-Ying. Research progress of SVR parameter optimization based on swarm intelligence algorithm[J] // Computer Engineering and Applications,2021,57(16):50-64.

31. Ding S, Zhao H, Zhang Y, et al. Extreme learning machine: algorithm, theory and applications[J] // Artificial Intelligence Review, 2015, 44(1): 103-115.

32. Liu H, Mi X, Li Y. An experimental investigation of three new hybrid wind speed predictioning models using multi-decomposing strategy and ELM algorithm[J] // Renewable energy, 2018, 123: 694-705.

33. Luo F, Guo W, Yu Y, et al. A multi-label classification algorithm based on kernel extreme learning machine[J] // Neurocomputing, 2017, 260: 313-320.

34. CSDN // Kernel Extreme Learning Machine (KELM) based classification. [N.p.], 2021. URL: https://blog.csdn.net/u011835903/article/details/116840012 (access date: 02. 01. 2022)

35. Whitley D. A Genetic Algorithm tutorial[J] // Statistics and computing, 1994, 4(2): 65-85.

36. Kumar M, Husain M, Upreti N, et al. Genetic Algorithm: Review and application[J] // Available at SSRN 3529843, 2010.

37. Katoch S, Chauhan S S, Kumar V. A review on Genetic Algorithm: past, present, and future[J] // Multimedia Tools and Applications, 2021, 80(5): 8091-8126.

38. Vose M D. The simple Genetic Algorithm: foundations and theory[M] // MIT press, 1999.

39. CSDN // Big talk about genetic algorithms. [N.p.], 2017. URL: https://blog.csdn.net/acelit/article/details/78187715 (access date: 12.11.2021)

40. Kennedy J, Eberhart R. Particle Swarm Optimization[C] // Proceedings of ICNN'95-international conference on neural networks. IEEE, 1995, 4: 1942-1948.

41. Shi Y. Particle Swarm Optimization: developments, applications and resources[C] // Proceedings of the 2001 congress on evolutionary computation (IEEE Cat. No. 01TH8546). IEEE, 2001, 1: 81-86.

42. Shi Y. Particle Swarm Optimization[J] // IEEE connections, 2004, 2(1): 8-13.

43. CSDN // Optimization Algorithm. [N.p.], 2018. URL: https://blog.csdn.net/wang454592297/article/details/80367158 (access date: 03. 05. 2021)

44. CSDN // Introduction to Particle Swarm Optimization (PSO) algorithm and MATLAB implementation. [N.p.], 2018. URL: https://blog.csdn.net/lyxleft/article/details/82978362 (access date: 11.11.2021)

45. Ouyang C, Zhu D, Wang F. A learning Sparrow Search Algorithm[J] // Computational Intelligence and Neuroscience, 2021, 2021.

46. CSDN // Optimization Algorithm Notes - Sparrow Search Algorithm. [N.p.], 2020. URL: https://www.jianshu.com/p/70ed22bc609d (access date: 12.12.2021)

47. Zhihu // Sparrow Search Algorithm. [N.p.], 2020. URL: https://zhuanlan.zhihu.com/p/134217408 (access date: 12.12.2021)

48. Code sample[6] - [N.p.], 2020. URL: https://python.iitter.com/other/135147.html (access date: 10.11.2021)

49. Abualigah L, Diabat A. Advances in Sine Cosine Algorithm: a comprehensive survey[J] // Artificial Intelligence Review, 2021, 54(4): 2567-2608.

50. CSDN // Intelligent optimization algorithm: sine cosine optimization algorithm. [N.p.], 2020. URL: https://blog.csdn.net/u011835903/article/details/107762654 (access date: 01.03.2022)

51. CSDN // [Optimization Algorithm] Sine Cosine Algorithm. [N.p.], 2021. URL: https://qq912100926.blog.csdn.net/article/details/120209221?spm=1001.2101.3001.6650.2&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-2.pc_relevant_default&utm_relevant_index=5 (access date: 02.03.2022)

52. ZHOU Xuejun,CHEN Xiaoqiang,XIE Lei,JIANG Chenglong. A hybrid model for short-term wind speed prediction based on EMD[J] // Journal of

Liaoning University of Petroleum and Chemical Technology,2021,41(06):79-86.

53. Zhang Y, Zhang C, Sun J, et al. Improved wind speed prediction using empirical mode decomposition[J] // Advances in Electrical and Computer Engineering, 2018, 18(2): 3-10.

54. Github // Introduction to empirical modal decomposition. [N.p.], 2018. URL: https://muyi110.github.io/2019/%E6%B5%85%E8%B0%88%E7%BB%8F%E9%AA%8C%E6%A8%A1%E6%80%81%E5%88%86%E8%A7%A3-EMD/ (access date: 05.10.2021)

55. Torres M E, Colominas M A, Schlotthauer G, et al. A complete ensemble empirical mode decomposition with adaptive noise[C] // 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2011: 4144-4147.

56. Zhang, J., Yan, R., Gao, R. X., & Feng, Z. (2010). Performance enhancement of ensemble empirical mode decomposition. // Mechanical Systems and Signal Processing, 24(7), 2104-2123.

57. Dragomiretskiy K, Zosso D. Variational Mode Decomposition[J] // IEEE Transactionson Signal Processing,2014,62(3):531-544.

58. Hu H, Wang L, Tao R. Wind speed predictioning based on variational mode decomposition and improved echo state network[J] // Renewable Energy, 2021, 164: 729-751.

59. Zhang, Y., Liu, K., Qin, L., & An, X. (2016). Deterministic and probabilistic interval prediction for short-term wind power generation based on variational mode decomposition and machine learning methods. // Energy Conversion and Management, 112, 208-219.

60. Walczak, B., and D. L. Massart. “Noise suppression and signal compression using the wavelet packet transform.”// Chemometrics and Intelligent Laboratory Systems 36.2 (1997): 81-94.

61. Soltani, Skander. "On the use of the wavelet decomposition for time series prediction." // Neurocomputing 48.1-4 (2002): 267-277.

62. Abramovich, F., & Silverman, B. W. (1998). Wavelet decomposition approaches to statistical inverse problems. // Biometrika, 85(1), 115-129.

63. Zhang D. Wavelet transform[M] // Fundamentals of Image Data Mining. Springer, Cham, 2019: 35-44.

64. Chun-Lin L. A tutorial of the wavelet transform[J] // NTUEE, Taiwan, 2010, 21: 22.

APPENDIX A

Part A Optimize algorithms for specific code implementation

1). Genetic Algorithm

Code Sample 2[38]: Genetic Algorithm of selection: Selection of evolved individuals using the Roulette Wheel Selection method.

```
for i=1:population_size
    r = rand * fitness_sum(population_size);
    first = 1;
    last = population_size;
    mid = round((last+first)/2);
    idx = -1;

    % Individual selection by the middle row method
    while (first <= last) && (idx == -1)
        if r > fitness_sum(mid)
            first = mid;
        elseif r < fitness_sum(mid)
            last = mid;
        else
            idx = mid;
            break;
        end
        mid = round((last+first)/2);
        if (last - first) == 1
            idx = last;
            break;
        end
    end

   % Generating a new generation of individuals
   for j=1:chromosome_size
        population_new(i,j) = population(idx,j);
   end
end
for i=1:population_size
   for j=1:chromosome_size
       population(i,j) = population_new(i,j);
    end
end
```

Code sample 2[38] – matlab code implementation of selection in Genetic Algorithm

Code Sample 3[38]: Genetic Algorithm of crossover.

```
for i=1:2:population_size
    % Crossover manipulation of chromosome strings of two individuals
    if(rand < cross_rate)
        cross_position = round(rand * chromosome_size);
        if (cross_position == 0 || cross_position == 1)
            continue;
        end
        % Swap binary strings for cross_position and beyond
        for j=cross_position:chromosome_size
            temp = population(i,j);
            population(i,j) = population(i+1,j);
            population(i+1,j) = temp;
        end
    end
end
```

Code sample 3[38] – Matlab code implementation of crossover in Genetic Algorithm

Code Sample 4[38]: Genetic Algorithm of mutation.

```
for i=1:population_size
    if rand < mutate_rate
        mutate_position = round(rand*chromosome_size);
        if mutate_position == 0
            continue;
        end
        population(i,mutate_position) = 1 - population(i, mutate_position);
    end
end
```

Code sample 4[38] – Matlab code implementation of Mutation in Genetic Algorithm

2). Particle Swarm Optimization

Code Sample of Particle Swarm Optimization algorithm [43]:

```
  %% I. Emptying the environment
clc
clear all

%% II.  Plot the objective function curve
x = 1:0.01:2;
y = sin(10*pi*x) ./ x;
figure
plot(x, y)
hold on

%% III. Parameter initialization
c1 = 1.49445;
c2 = 1.49445;
maxgen = 50;   % Number of evolutions
sizepop = 10;   % Population size
Vmax = 0.5;   %Range of speed。
Vmin = -0.5;
popmax = 2;   %Individual range of variation
popmin = 1;


%% IV. Generate initial particles and velocities
for i = 1:sizepop
    pop(i,:) = (rands(1) + 1) / 2 + 1;
    V(i,:) = 0.5 * rands(1);
    fitness(i) = fun(pop(i,:));
end


%% V. Individual and group extremes
[bestfitness bestindex]= max(fitness);
zbest = pop(bestindex,:);
gbest = pop;
fitnessgbest = fitness;
fitnesszbest = bestfitness;


%% VI. Iterative Optimization Search
```

```
for i = 1:maxgen
    for j = 1:sizepop
        V(j,:) = V(j,:) + c1*rand*(gbest(j,:) - pop(j,:)) + c2*rand*(zbest - pop(j,:));
        V(j,find(V(j,:)>Vmax)) = Vmax;
        V(j,find(V(j,:)<Vmin)) = Vmin;

        pop(j,:) = pop(j,:) + V(j,:);
        pop(j,find(pop(j,:)>popmax)) = popmax;
        pop(j,find(pop(j,:)<popmin)) = popmin;

        fitness(j) = fun(pop(j,:));
    end

    for j = 1:sizepop
        if fitness(j) > fitnessgbest(j)
            gbest(j,:) = pop(j,:);
            fitnessgbest(j) = fitness(j);
        end

        if fitness(j) > fitnesszbest
            zbest = pop(j,:);
            fitnesszbest = fitness(j);
        end
    end
    yy(i) = fitnesszbest;
end
```

Code sample 5[43] – Matlab code implementation of PSO algorithm

3). Sparrow Search Algorithm

Code sample of Sparrow Search Algorithm[47]:

```
clear all;
close all;
clc;

%% Parameter Setting
N=30;
dim=2;
N_discoverer=0.7*N;
N_Followers=0.1*N;
N_Vigilant=0.2*N;
```

```
Max_iter=100;
ST=0.6;

%% Test Functions
f=@(x) sum(x.^2);
ub=10;
lb=-10;

%% Initialization
x=lb+rand(N,dim).*(ub-lb);
for i=1:N
    fitness(i)=f(x(i,:));
end
[A,index]=sort(fitness);
x_best=x(index(1),:);
x_worst=x(index(end),:);
best_fitness=A(1);
worst_fitness=A(end);
x_best_currently=x(index(1),:);
x_worst_currently=x(index(end),:);
best_fitness_currently=A(1);
worst_fitness_currently=A(end);
x_discoverer=x(index(1:N_discoverer),:);
x_Followers=x(index(N_discoverer+1:N_discoverer+N_Followers),:);
x_Vigilant=x(index(N_discoverer+N_Followers+1:N),:);
B=[-1,1];
F=best_fitness;
iter=1;

%% Start of iterative updates
while iter<Max_iter
    for i=1:dim
        C(i)=B(round(rand)+1);
    end
    A=C'*inv((C*C'));
    R2=rand;
    for i=1:N_discoverer
        for j=1:dim
            if R2<ST
                x_discoverer(i,j)=x_discoverer(i,j)*exp(-i/rand*Max_iter);
            else
                 x_discoverer(i,j)=x_discoverer(i,j)+randn;
            end
        end
        ub_flag=x_discoverer(i,:)>ub;
        lb_flag=x_discoverer(i,:)<lb;
        x_discoverer(i,:)=(x_discoverer(i,:).*(~(ub_flag+lb_flag)))+ub.*ub_flag+lb.*lb_flag;
    end
    for i=1:N_Followers
        for j=1:dim
            if i>N/2
                x_Followers(i,j)=rand*exp((x_worst_currently(j)-x_Followers(i,j))/i^2);
            else
                x_Followers(i,j)=x_discoverer(1,j)+abs(x_Followers(i,j)-x_discoverer(1,j))*A(j);
            end
        end
        ub_flag=x_Followers(i,:)>ub;
        lb_flag=x_Followers(i,:)<lb;
        x_Followers(i,:)=(x_Followers(i,:).*(~(ub_flag+lb_flag)))+ub.*ub_flag+lb.*lb_flag;
    end
    for i=1:N_Vigilant
        for j=1:dim
            if f(x_Vigilant(i,:))~=best_fitness_currently
                x_Vigilant(i,j)=x_best_currently(j)+randn*abs(x_Vigilant(i,j)-x_best_currently(j));
            else
                x_Vigilant(i,j)=x_Vigilant(i,j)+B(round(rand)+1)*(abs(x_Vigilant(i,j)-x_worst_currently(j)))/abs(f(x_Vigilant(i,:))-worst_fitness_currently)+1;
            end
        end
        ub_flag=x_Vigilant(i,:)>ub;
        lb_flag=x_Vigilant(i,:)<lb;
        x_Vigilant(i,:)=(x_Vigilant(i,:).*(~(ub_flag+lb_flag)))+ub.*ub_flag+lb.*lb_flag;
    end
    x=[x_discoverer;x_Followers;x_Vigilant];
    for i=1:N
        fitness(i)=f(x(i,:));
    end
    [E,index]=sort(fitness);
    if f(x(index(1),:))<best_fitness
        best_fitness=f(x(index(1),:));
        x_best=x(index(1),:);
```

```
 end
 if f(x(index(end),:))>worst_fitness
     worst_fitness= f(x(index(end),:));
     x_worst=x(index(end),:);
 end
 x_best_currently=x(index(1),:);
 x_worst_currently=x(index(end),:);
 best_fitness_currently=E(1);
 worst_fitness_currently=E(end);
    x_discoverer=x(index(1:N_discoverer),:);
    x_Followers=x(index(N_discoverer+1:N_discoverer+N_Followers),:);
    x_Vigilant=x(index(N_discoverer+N_Followers+1:N),:);
    F=[F,best_fitness];
    iter=iter+1;
end
```

Code sample 6[47] – Matlab code implementation of SSA algorithm

4). Sine Cosine Algorithm

Code sample of Sine Cosine Algorithm[50]:

```
   clear all
clc

SearchAgents_no=30; % Number of search agents

Function_name='F1'; % Name of the test function that can be from F1 to F23 (Table 1,2,3 in the paper)

Max_iteration=1000; % Maximum numbef of iterations

% Load details of the selected benchmark function
[lb,ub,dim,fobj]=Get_Functions_details(Function_name);

[Best_score,Best_pos,cg_curve]=SCA(SearchAgents_no,Max_iteration,lb,ub,dim,fobj);

figure('Position',[284   214   660   290])
%Draw search space
subplot(1,2,1);
func_plot(Function_name);
title('Test function')
xlabel('x_1');
```

```
ylabel('x_2');
zlabel([Function_name,'( x_1 , x_2 )'])
grid off

%Draw objective space
subplot(1,2,2);
semilogy(cg_curve,'Color','b')
title('Convergence curve')
xlabel('Iteration');
ylabel('Best flame (score) obtained so far');

axis tight
grid off
box on
legend('SCA')

display(['The best solution obtained by SCA is : ', num2str(Best_pos)]);
display(['The best optimal value of the objective funciton found by SCA is : ', num2str(Best_score)]);

function [Destination_fitness,Destination_position,Convergence_curve]=SCA(N,Max_iteration,lb,ub,dim,fobj)

display('SCA is optimizing your problem');

%Initialize the set of random solutions
X=initialization(N,dim,ub,lb);

Destination_position=zeros(1,dim);
Destination_fitness=inf;

Convergence_curve=zeros(1,Max_iteration);
Objective_values = zeros(1,size(X,1));

% Calculate the fitness of the first set and find the best one
for i=1:size(X,1)
    Objective_values(1,i)=fobj(X(i,:));
    if i==1
        Destination_position=X(i,:);
        Destination_fitness=Objective_values(1,i);
    elseif Objective_values(1,i)<Destination_fitness
        Destination_position=X(i,:);
        Destination_fitness=Objective_values(1,i);
    end

    All_objective_values(1,i)=Objective_values(1,i);
end

%Main loop
t=2; % start from the second iteration since the first iteration was dedicated to calculating the fitness
while t<=Max_iteration

    % Eq. (3.4)
    a = 2;
    Max_iteration = Max_iteration;
    r1=a-t*((a)/Max_iteration); % r1 decreases linearly from a to 0

    % Update the position of solutions with respect to destination
    for i=1:size(X,1) % in i-th solution
        for j=1:size(X,2) % in j-th dimension

            % Update r2, r3, and r4 for Eq. (3.3)
            r2=(2*pi)*rand();
            r3=2*rand;
            r4=rand();

            % Eq. (3.3)
            if r4<0.5
                % Eq. (3.1)
                X(i,j)= X(i,j)+(r1*sin(r2)*abs(r3*Destination_position(j)-X(i,j)));
            else
                % Eq. (3.2)
                X(i,j)= X(i,j)+(r1*cos(r2)*abs(r3*Destination_position(j)-X(i,j)));
            end

        end
    end
```

Code sample 7[50] *– Matlab code implementation of SCA algorithm's main loop and optimal solution*

С отчетом ознакомлена.
Научный руководитель ВКР,
к.ф.-м.н, доцент И.Н. Зенкова
08.06.2022

ANTIPLAGIAT
PLAGIARISM DETECTION SYSTEM

Text reuse report No.1

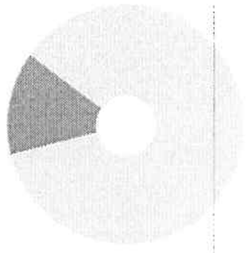
Author: Hu Zhichao 2903539308zhichao@gmail.com / ID: 9754220
Checked by: Hu Zhichao (2903539308zhichao@gmail.com / ID: 9754220)
Report presented by the Antiplagiat service - http://users.antiplagiat.ru

DOCUMENT INFORMATION

Document No.: 3
Uploading start: 08.06.2022 11:27:58
Uploading duration: 00:00:09
Initial file name: Bachelor Thesis(10).pdf
Document name: Bachelor Thesis(10)
Text size: 1 KB
Number of characters: 65582
Number of words: 8623
Number of sentences: 331

REPORT INFORMATION

Latest report (edit.)
Check start: 08.06.2022 11:28:07
Check duration: 00:00:45
Comments: not specified
Search with editing factored in: yes
Search modules: Legal information system «Adilet» search module, Bibliography separation module, ELS joint collection, Search module INTERNET PLUS, RSL collection, Citations, Cross language search (RuEn), Cross language search on eLIBRARY.RU (EnRu), Cross language search on eLIBRARY.RU (KkRu), Cross language search on eLIBRARY.RU (KyRu), Cross language search over the Internet (EnRu), Cross language search over the Internet (KkRu), Cross language search over the Internet (KyRu), Cross language search (KkEn), Cross language search (KyEn), Wiley Open Library Cross language, eLIBRARY.RU collection, GARANT collection, Medical collection, Dissertations and abstracts of National Library of Belarus, Search module of eLIBRARY.RU paraphrases, Search module of Internet paraphrases, Search module of Wiley Open Library paraphrases, Patents collection, Media collection, Search module of common phrases, Institutes Unity collection, Wiley Open Library, Cross language search



REUSE	SELF-CITATIONS	CITATIONS	ORIGINALITY
13,98%	0%	0%	86,02%

SUSPICIOUS DOCUMENT

The following cheating methods may have been used: **СВЯЗНОСТЬ ТЕКСТА**

Reuse is a share of all found overlapping text fragments, with the exception of those categorized as citations by the system, as compared to the entire document size.
Self-citations refer to the share of text fragments (relative to the whole document) in the document being checked for reuse that are completely/partially identical to the text fragments from a source which was authored/co-authored by the individual whose document is being checked.
Citations are a share of overlapping text which has not been created by the respective author, but the system considered its use correct as compared to the entire document size. These include citations complying with the GOST standard; common phrases; text fragments found in legal and regulatory documentation source collections.
Overlapping text is a text fragment of the checked document which is identical or almost identical to a source text fragment.
Source is a document indexed by the system and contained in the search module which is used for check.
Originality is a share of text fragments in the checked document that have not been found in any checked sources as compared to the entire document size.
Reuse, self-citations, citations, and originality are separate indicators. Their sum is equal to 100%, with respect to the entire text of the checked document.
Please note that the system finds overlapping texts in the checked document and text sources indexed by the system. At the same time, the system is an auxiliary tool. Correctness and adequacy of reuse or citations, as well as authorship of text fragments in the checked document must be determined by the verifier.

№	Report share	Source	Valid from	Search module	Comments
[01]	0,84%	Composite quantile regression extreme learning machine with feature selection for short-term wind speed forecasting: A new approach https://doi.org	12 May 2018	Search module INTERNET PLUS	
[02]	0,68%	http://orca.cf.ac.uk/124512/1/energies-11-00697-v3.pdf http://orca.cf.ac.uk	17 Dec 2019	Search module INTERNET PLUS	
[03]	0,39%	A Hybrid Model for Forecasting Sunspots Time Series Based on Variational Mode Decomposition and Backpropagation Neural Network Improved by Firefly Algorithm https://hindawi.com	26 Apr 2021	Search module INTERNET PLUS	
[04]	1,28%	A prediction approach using ensemble empirical mode decomposition-permutation entropy and regularized extreme learning machine for short-term wind speed https://doi.org	22 Jan 2021	Wiley Open Library	
[05]	0,46%	https://www.ijitee.org/wp-content/uploads/Souvenir_IJITEE_Volume-8_Issue-6_April_2019.pdf https://ijitee.org	04 Aug 2020	Search module INTERNET PLUS	
[06]	0,57%	Hybrid multiscale wind speed forecasting based on variational mode decomposition https://doi.org	31 Jan 2018	Wiley Open Library	
[07]	0,23%	Precise Feature Extraction from Wind Turbine Condition Monitoring Signals by using Optimized Variational Mode Decomposition http://eprint.ncl.ac.uk	06 Feb 2017	Search module INTERNET PLUS	