Mathematical Biosciences and Engineering

MBE, 19 (5): 4547–4567.
DOI: 10.3934/mbe.2022210
Received: 15 December 2021
Revised: 30 January 2022
Accepted: 17 February 2022
Published: 04 March 2022

http://www.aimspress.com/journal/MBE

Research article

Ship power load forecasting based on PSO-SVM

Xiaoqiang Dai[1,3], Kuicheng Sheng[2], Fangzhou Shu[1,*]

[1] School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, China
[2] Jiangsu Institute of Automation, Lianyungang 222000, China
[3] Zhoushan Jiangke ship and marine engineering equipment R & D Center, Zhoushan 316021, China

* **Correspondence:** Email: 1508472012@qq.com.

Abstract: Compared with the land power grid, power capacity of ship power system is small，its power load has randomness. Ship power load forecasting is of great significance for the stability and safety of ship power system. Support vector machine (SVM) load forecasting algorithm is a common method of ship power load forecasting. In this paper, water flow velocity, wind speed and ship speed are used as the features of SVM to train the load forecasting algorithm, which strengthens the correlation between features and predicted values. At the same time, regularization parameter C and standardization parameter σ of SVM has a great influence on the prediction accuracy. Therefore, the improved particle swarm optimization algorithm is used to optimize these two parameters in real time to form a new improved particle swarm optimization support vector machine algorithm (IPSO-SVM), which reduces the load forecasting error, improves the prediction accuracy of ship power load, and improves the performance of ship energy management system.

Keywords: ship power system; load forecasting; support vector machine; improved particle swarm optimization support vector machine

1. Introduction

With the advancement of intelligent ships, the degree of ship automation is getting higher and higher. The propulsion mode of ships is changed from main engine propulsion to electric propulsion. The stability of ship power system is related to the safety of ships. Ship power system has small capacity, complex working conditions and strong randomness of load, so the prediction of ship power

load will be related to the management strategy of ship power system energy tube system and the stability of ship power system, which is of great significance to the safety of ship power system.

Load prediction algorithms in land power grid are relatively mature, including regression analysis [1–3], grey model [4–6], fuzzy prediction [7–9], autoregressive integral moving average model [10,11] and other traditional prediction methods. There is also random forest [12,13], support vector machine [14,15], expert system [16,17], artificial neural network prediction method [18–20] and other machine learning methods. Support vector machine load prediction algorithm with its small number of sample data can solve the obvious advantages of high dimensional ship power system application is relatively mature. Compared with the land power system, the ship power system has the characteristics of equal capacity and load, complex working conditions and strong randomness of load, but the load of the ship power system has the same periodicity and continuity as the land power system. These factors will affect the existence and feasibility of the solution of the adjustable parameters of the prediction model [21], as well as the global existence and local convergence of the solution [22–23]. This paper studies the load prediction algorithm of support vector machine and the parameters such as temperature, water flow velocity, relative wind speed and ship speed are innovatively used as the training characteristic values of support vector machines, which strengthens the correlation between input characteristics and output load forecast value and reduces the prediction error. At the same time, aiming at the problem that it is difficult to select parameters when using support vector machine to forecast the load, particle swarm optimization algorithm is introduced to optimize parameters, which enhances the accuracy of ship power load forecasting algorithm.

2. Support vector machine load forecasting algorithm

2.1. Basic theory of support vector machine

As a machine learning language based on statistical learning theory, support vector machine has strong advantages in solving problems such as small samples, nonlinearity and high dimensions [24]. Since the core content of support vector machine was proposed in 1992, it was initially widely used in pattern recognition to make decision rules with good generalization performance. Later, with the development of support vector machines and Vapnik introduced the insensitive loss function ε, the support vector machine algorithm is extended to some nonlinear regression problems. The basic idea of SVM is to define an optimal linear hyperplane and transform the algorithm for finding the hyperplane into an optimization problem [25]. According to the principle of structural risk minimization, the actual risk is minimized, and the value of h of empirical risk and VC dimension are minimized [26].

2.2. Support vector machine regression algorithm

Select training sample S as the training sample set for support vector machine regression prediction [27]:

$$S = \{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_l, y_l)\} \in R^n \times R \tag{1}$$

In Eq (1), S represents the training sample set of the system; $x_i \in R^n$ represents the sample input information of the ship's historical power load, $y_i \in R$ represents the output of the corresponding input, and l is the total number of training samples in the training data set. Establish the mapping set

f: $R^n \to R$ from the input space R^n to the output R, which satisfies f(x) = y, then the regression function is shown in Eq (2):

$$y = f(x) = \omega \cdot \varphi(x) + b \tag{2}$$

In Eq (2), ω represents the weight vector, b represents the bias of regression, $\varphi(x)$ represents the mapping function of samples from low-dimensional to high-dimensional space. According to the theory of modern statistics, the regression prediction error of support vector machine is mainly caused by empirical risk and $\| \omega \|^2$. The total error can be expressed as the equation shown in Eq (3):

$$R(\omega) = \frac{1}{2} \| \omega \|^2 + \sum_{i=1}^{l} e(f(x_i) - y_i) \tag{3}$$

In Eq (3), e () is called loss function, which can be specifically expressed as Eq (4):

$$(f(x_i) - y_i) = \begin{cases} 0, |f(x_i) - y_i| < \varepsilon \\ |f(x_i) - y_i| - \varepsilon, other \end{cases} \tag{4}$$

In this equation, ε is called insensitive function. When using support vector machine for regression prediction, if you want to improve its prediction accuracy, you only need to minimize the value of R(ω) within the range of constraints. According to the structural risk minimization criterion, the risk minimization of support vector machine load forecasting algorithm can be expressed as:

$$min[R(\omega)] = min\left[\frac{1}{2} \| \omega \|^2 + \sum_{i=1}^{l} e(f(x_i) - y_i)\right] \tag{5}$$

when fitting the regression model, in order to enhance the accuracy of the regression model, it is necessary to consider the possible errors in the calculation process of the model. To realize the expression of prediction model error by introducing two relaxation factors ξ_i, ξ_i^*, then the loss function of Eq (5) can be expressed as Eq (6):

$$min[\frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{l} e(\xi_i + \xi_i^*)]$$

$$s.t. \begin{cases} y_i - \omega \cdot \varphi(x) - b \geq e + \xi_i \\ \omega \cdot \varphi(x) + b - y_i \geq e + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{6}$$

In Eq (6), the first half represents the generalization ability of the model, and the latter half is used to reduce the error in the training process. C (c > 0) is the penalty parameter of regression error, which is called regularization parameter. The minimum solution problem shown is a typical optimization problem with inequality constraints. When solving the minimum value, we can consider transforming it into a dual problem by introducing Lagrange function. By introducing Lagrange multiplier α_i, α_i^*, Lagrange equation as shown in Eq (7) can be established:

$$L(\omega,\xi,\xi^*,\alpha_i,\alpha_i{}^*,\lambda_i)=\frac{1}{2}\parallel\omega\parallel^2+C\sum_{i=1}^{l}(\xi+\xi^*)-\sum_{i=1}^{l}\alpha_i\big((e+\xi_i)-y_i+f(x_i)\big)$$

$$-\sum_{i=1}^{l}\alpha_i{}^*\big((e+\xi_i{}^*)-y_i-f(x_i)\big)-\sum_{i=1}^{l}(\lambda_i\cdot\xi_i+\lambda_i{}^*\cdot\xi_i{}^*) \tag{7}$$

In Eq (7), $\alpha_i,\alpha_i{}^*,\lambda_i\geq 0,i=1,\ldots,l.$

According to the optimal condition of Lagrange equation, there is $\frac{\partial L}{\partial\omega}=0,\frac{\partial L}{\partial b}=0,\frac{\partial L}{\partial\xi_i}=0,\frac{\partial L}{\partial\xi_i{}^*}=0$, expand and simplify the equation to obtain a set of equations shown in Eq (8):

$$\begin{cases}\frac{\partial L}{\partial\omega}=\omega-\sum_{i=1}^{l}(\alpha_i-\alpha_i{}^*)\,x_i=0\\ \frac{\partial L}{\partial b}=\sum_{i=1}^{l}(\alpha_i-\alpha_i{}^*)=0\\ \frac{\partial L}{\partial\xi_i}=C-\alpha_i-\lambda_i=0\\ \frac{\partial L}{\partial\xi_i{}^*}=C-\alpha_i{}^*-\lambda_i{}^*=0\end{cases},i=1,\ldots,l \tag{8}$$

By introducing Eq (8) into Eq (7), the dual problem of the optimization problem of Eq (9) can be obtained:

$$L_{max}=-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i{}^*)\left(\alpha_j-\alpha_j{}^*\right)<x_i,x_j>$$

$$+\sum_{i=1}^{l}(\alpha_i-\alpha_i{}^*)\,y_i-\sum_{i=1}^{l}(\alpha_i-\alpha_i{}^*)\,\varepsilon$$

$$s.t.\quad\sum_{i=1}^{l}(\alpha_i-\alpha_i{}^*)=0 \tag{9}$$

In this equation, $0\leq\alpha_i,\alpha_i{}^*\leq C.$

By solving the quadratic programming problem of Eq (9), we can obtain the further expression of the regression function f(x), which can be divided into two categories: linear optimization and nonlinear optimization [28].

When the training sample is a linear model:

$$L_{max}=-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i{}^*)\left(\alpha_j-\alpha_j{}^*\right)x_ix_j+\sum_{i=1}^{l}(\alpha_i-\alpha_i{}^*)\,y_i-\sum_{i=1}^{l}(\alpha_i-\alpha_i{}^*)\,\varepsilon$$

$$s.t.\quad\sum_{i=1}^{l}(\alpha_i-\alpha_i{}^*)=0 \tag{10}$$

By further solving Eq (10), we can get:

$$f(x)=\sum_{x_i\in N_{sv}}^{l}(\alpha_i-\alpha_i{}^*)\,x_ix_j+b \tag{11}$$

N_{sv} is the set of support vectors in the regression model.

When the regression model is nonlinear:

When solving the regression problem of nonlinear model, we usually use a nonlinear mapping to

map the original training samples to the high-dimensional space, and realize the linear regression of training samples in the high-dimensional space. In the above process, the inner product operation $< \varphi(x), \varphi(y) >$ is used in the high-dimensional space because the samples in the low-dimensional space are mapped to the high-dimensional space, then Eq (10) can be specifically expressed as follows when the regression model is nonlinear:

$$L_{max} = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)\left(\alpha_j - \alpha_j^*\right)K\left(x_i, x_j\right)$$

$$+\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\, y_i - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\,\varepsilon$$

$$s.t. \quad \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \tag{12}$$

$K\left(x_i, x_j\right) = \varphi(x)\varphi(y)$ represents the kernel function. Then the regression model:

$$f(x) = \sum_{x_i \in N_{sv}}^{l}(\alpha_i - \alpha_i^*)\, K\left(x_i, x_j\right) + b \tag{13}$$

2.3. Least squares support vector machine load forecasting model

Least squares support vector machine (LSSVM) is a new algorithm based on support vector machine. The analysis shows that the process of solving SVM model is actually to solve a quadratic programming problem with inequality constraints. In LSSVM algorithm, the square error term is selected as the optimization index, and the inequality constraint in SVM algorithm is replaced by equality constraint, so as to convert the quadratic programming problem of SVM into the solution of linear model [29].

The basic idea of LSSVM regression is the same as that of nonlinear SVM, which maps samples into high-dimensional space through a nonlinear mapping, and performs linear regression on samples in high-dimensional space. Its basic expression is the same as that of SVM:

$$y = f(x) = \omega \cdot \varphi(x) + b \tag{14}$$

Then the optimization problem of LSSVM can be expressed as Eq (15):

$$\min J(\omega, b, e) = \frac{1}{2}\parallel \omega \parallel^2 + \frac{1}{2}C\sum_{i=1}^{l} {e_i}^2$$

$$s.t. \quad \omega^T\varphi(x_i) + b + e_i = y_i, i = 1, \ldots, l \tag{15}$$

In this equation: $e_i \in R^{l\times 1}$ is the regression error, R is the error vector composed of errors e_i. And C represents the regularization parameter, which is used to control the penalty degree of error in the regression process.

Similarly, Lagrange multipliers λ are introduced into Eq (15), Eq (15) can be expressed as:

$$L(\omega, \mathrm{b}, \mathrm{e}, \lambda) = J(\omega, b, e) - \sum_{i=1}^{l}\lambda_i\left(\omega^T\varphi(x_i) + b + e_i - y_i\right) \tag{16}$$

For solving the minimum problem in the specified scope with constraints, the KKT condition can be used to obtain:

$$\begin{cases} \frac{\partial J}{\partial \omega} = \sum_{i=1}^{l} \lambda_i \varphi(x_i) = 0 \\ \frac{\partial J}{\partial b} = \sum_{i=1}^{l} \lambda_i = 0 \\ \frac{\partial J}{\partial e_i} = \lambda_i - C e_i = 0 \\ \frac{\partial J}{\partial \lambda_i} = \omega^T \varphi(x_i) + b + e_i - y_i = 0 \end{cases}, i = 1, \ldots, l \tag{17}$$

Eliminate ω and e in the equation, after solving the Eq (17), we can get:

$$\begin{bmatrix} b \\ C \end{bmatrix} = \begin{bmatrix} 0 & E \\ E & K + \frac{I}{Y} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ Y \end{bmatrix} \tag{18}$$

In this equation, $\mathrm{E} = [1,1, \ldots,1]^T$, $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$ represents the kernel function under suitable conditions, I stands for identity matrix, $\mathrm{Y}=[y_1, y_2, \ldots, y_l]^T$, then the prediction model of LSSVM can be expressed as:

$$y = \sum_{i,j=1}^{l} \lambda_i K(x_i, x_j) + b \tag{19}$$

λ_i and b can be solved by linear equations, $K(x_i, x_j)$ represents the kernel function mapping samples from low-dimensional space to high-dimensional space. The commonly used kernel functions generally include linear kernel function, RBF kernel function, polynomial kernel function, etc. Gaussian radial basis kernel function (RBF) is widely used in SVM regression estimation because of its simple expression, radial symmetry and good analytical properties [30,31]. In this paper, RBF is selected as the kernel function:

$$K(x_i, x_j) = \exp\left[-\frac{\left||x_i - x_j|\right|^2}{2\sigma^2}\right] \tag{20}$$

In this this equation, x_i represents the input vector, x_j represents the center of the j-th Gaussian basis function, σ represents the standardized parameters that determine the width of the center point surrounded by the Gaussian function. $\left||x_i - x_j|\right|$ represents the norm of the vector $(x_i - x_j)$, which can specifically represent the distance between the vector x_i and the vector x_j.

3. Improved particle swarm optimization support vector machine algorithm

When using LSSVM for load forecasting, the value of C and σ will have a great impact on the accuracy of prediction. In order to improve the prediction accuracy of load data regression model, particle swarm optimization algorithm is introduced to optimize the parameters of LSSVM load prediction algorithm, so as to form a new improved particle swarm optimization support vector machine algorithm, which enhances the accuracy of prediction model [32].

3.1. Particle swarm optimization algorithm

Particle swarm optimization (PSO) is a swarm intelligence algorithm that simulates the predation behavior of birds [33]. The algorithm searches the solution space of the problem by simulating the

foraging behavior of birds. Each particle in this algorithm has the function of information sharing, and the particle can change its position according to the fitness value in the environment. The position of each particle in the search space represents a feasible solution of the problem. Each particle moves in the search space at a certain speed, which represents the solution seeking process. The fitness value of the objective function of the problem is often used to judge the quality of the particle position. The population obtains the best particle position in the search space through multiple iterations, which represents the process of obtaining the optimal solution of the problem.

Firstly, some particles are randomly generated in a space, and the scale of the example group (i.e., the number of particles) and the search dimension are set. If the total number of particles is n and the dimension is Q [34], each particle can be expressed as $x_i=(x_{i1}, x_{i2}, x_{i3}, \dots, x_{iQ})$, the corresponding speed can be expressed as $v_i=(v_{i1}, v_{i2}, v_{i3}, \dots, v_{iQ})$, where $i \in 1,2, \dots, n$. Each particle needs to pay attention to the historical optimal solution $p_i=(p_{i1}, p_{i2}, p_{i3}, \dots, p_{iQ})$ searched by its individual iterative process and the the global optimal solution $p_g=(p_{g1}, p_{g2}, p_{g3}, \dots, p_{gQ})$ searched by all particles in the population.

In the iterative updating process of particles, the Eq (21) needs to be used to update the particle velocity and position in the population by PSO [35]:

$$\begin{cases} {v_{id}}^{k+1} = \omega {v_{id}}^{k} + c_1\xi\left({p_{id}}^{k} - {x_{id}}^{k}\right) + c_2\eta\left({p_{gd}}^{k} - {x_{id}}^{k}\right) \\ {x_{id}}^{k+1} = {x_{id}}^{k} + \gamma {v_{id}}^{k+1} \end{cases} \tag{21}$$

In this equation: ${v_{id}}^{k}$ and ${v_{id}}^{k+1}$ represent the velocity of particles at time k and time k+1. Similarly, ${x_{id}}^{k}$ and ${x_{id}}^{k+1}$ represent the position of particles at that two moments. ω is inertia weight, c1 and c2 are learning factors, ξ and η represent random numbers evenly distributed between [0,1]. γ is the constraint factor when updating the position of particles, which is generally set to 1.

It can be seen from Eq (21) that the speed of particles in the search space is affected by three factors: the first part is that ${v_{id}}^{k}$ is the speed of particles in the previous iteration, represents the flight inertia of particles, and has the ability to balance local search and global search. When ${v_{id}}^{k}$ is small, the global search ability of particles is weak, but the local search ability is strong. When ${v_{id}}^{k}$ is large, the local search ability of particles is weak, but the global search ability is strong; The second part is the thinking and learning of the particle on its own flight experience, that is, the current search tendency of the particle is changed by the attraction of the historical optimal solution ${p_{id}}^{k}$ to the particle in the flight process, in which the learning factor c_1 is used to adjust the moving step of the particle to the historical optimal solution ${p_{id}}^{k}$. The random adjustment of the value of ξ can reduce the situation that particles fall into local optimization and improve the global search ability of particles; The third part is the "social cognitive ability" of particles, that is, the process of particles learning the flight experience of the whole population. This process is directly related to the global optimal solution ${p_{gd}}^{k}$ of the population. The learning factor c_2 is used to control the step size of particles flying to the global optimal solution ${p_{gd}}^{k}$. Affected by the random adjustment of the value of η, particles can achieve mutual cooperation and information sharing. Under the joint action of these three factors, particles constantly adjust their position and flight speed through their own flight experience and population information sharing mechanism, and finally reach the optimal particle position to obtain the optimal solution of the problem to be solved [36].

When $\omega = 0$, there is no memory term in the velocity formula. The velocity formula becomes:

$${v_{id}}^{k+1} = 0 + c_1\xi\left({p_{id}}^{k} - {x_{id}}^{k}\right) + c_2\eta\left({p_{gd}}^{k} - {x_{id}}^{k}\right) \tag{22}$$

Particles do not have inertial motion, and they do not have the ability to expand space. Particles may not be able to search a certain area in space. In the whole optimization process, the particle group shows too much local search, and does not show the ability of particle to search the whole search space [37].

When $c_1 = 0$, the individual cognition term in the velocity formula does not exist, the particle loses the ability of self-evaluation and judgment, and retains the ability of 'social cognition' of the best position in the particle group history.

$$v_{id}{}^{k+1} = \omega v_{id}{}^{k} + 0 + c_2\eta\left(p_{gd}{}^{k} - x_{id}{}^{k}\right) \tag{23}$$

Particles move rapidly in the search space, and through information exchange and mutual learning between particles, particles can enter the new search space for optimization, and the algorithm quickly enters the convergence state. However, when calculating some complex problems, the optimization performance is not good, and the calculation result may be a local optimal value in the space rather than the global optimal value [38].

When $c_2 = 0$, that is, the group cognitive item in the velocity formula does not exist, the particles cannot communicate and learn, and there is no information transmission between the entire population. In the entire search space, the particles are independent individuals.

$$v_{id}{}^{k+1} = \omega v_{id}{}^{k} + c_1\xi\left(p_{id}{}^{k} - x_{id}{}^{k}\right) + 0 \tag{24}$$

Equivalent to a certain number of particles in the search space for blind search, each particle immediately search results are the extreme value of each particle, so often cannot get satisfactory results [39].

When $c_1 = 0, c_2 = 0$, there is neither individual cognition nor group cognition in the velocity formula, only the memory part of the particle itself.

$$v_{id}{}^{k+1} = \omega v_{id}{}^{k} + 0 + 0 \tag{25}$$

The basic particle swarm optimization process is as follows:

Step1: Initialization: randomly generate the position and velocity of particles in the D dimension of the problem space.

Step2: Evaluate particles: evaluate the applicable value of q-dimensional optimization function for each particle.

Step3: Update the optimal position: 1) Compare the current fitness value of the particle with its individual history optimal value. If it is better than p_i, its p_i is updated to the current particle position. 2) Comparing the current fitness value of particles with the global optimal value of population. If it is better than p_g, the global optimal solution position is updated to the current particle position.

Step4: Update particles: Update particle velocity and position according to Eq (21).

Step5: Stop condition: Loop back to Step 2 until the termination condition is satisfied, which is usually to meet the maximum iteration algebra.

Corresponding to the above algorithm flow, the basic framework of particle swarm algorithm is shown in Figure 1.

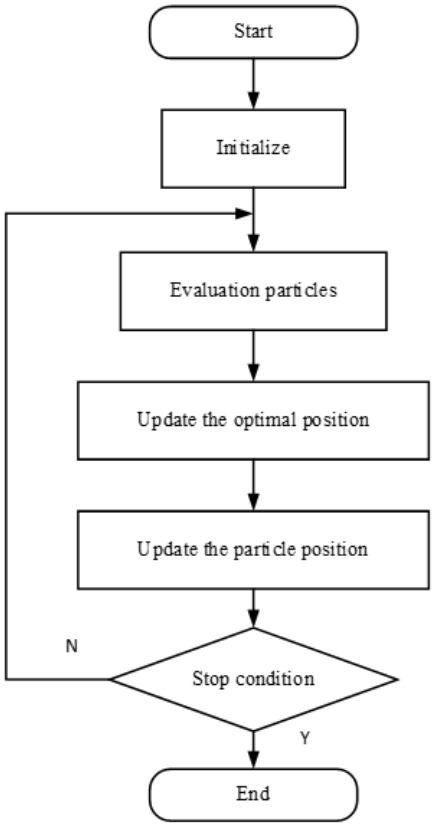


Figure 1. The basic framework of particle swarm algorithm.

During the operation of particle swarm optimization algorithm, the parameters affect the change of particle velocity and the update of particle position, and finally affect the result of particle optimization. The performance of particle swarm optimization algorithm mainly depends on the two key parts of local search and global search. When solving the actual optimization problem, it is necessary to balance global search and local search, and take into account the local search in the global search. In order to better solve this problem [40], the inertia weight ω is introduced into the flight inertia part ${v_{id}}^k$ that affects the particle's flight speed to balance the local search and global search. The larger the value of ω, the more particles inherit the previous speed and the faster the particle moves, which is conducive to avoiding falling into a local state during optimization and the global optimization of the algorithm. On the contrary, the smaller the value of ω, the weaker the ability of particles to explore space, and the particles are easy to carry out rapid optimization in a local area of space, which is not conducive to the global optimization of particles. If the inertia weight ω is not a constant but a dynamic decreasing law in the process of particle iteration, the inertia weight ω of the particle in the search space is large in the early stage of the algorithm, and the particle can carry out rapid exploratory optimization in the whole search space, so that the particle approaches the area around the global optimal position in a short time; In the later stage of the algorithm, the inertia weight ω of the particles is small so that the particles focus on the area around the optimal position for centralized mining optimization.

In this paper, a commonly used concave function decreasing strategy is taken as shown in Eq (26) to make the inertia weight ω keep large enough at the beginning of particle optimization so that the particles can quickly approach the optimal solution. In the later stage of the algorithm, the inertia

weight ω is constantly reduced, so that the algorithm can be more stable and the optimization effect is better.

$$\omega = min\omega \times \left(\frac{max\omega}{min\omega}\right)^{\frac{1}{1+10*\frac{CurCount}{LoopCount}}} \tag{26}$$

In this formula: $min\omega$ and $max\omega$ represent the maximum and minimum values of ω respectively. CurCount represents the current iteration number of the particle. LoopCount represents the total iteration number of the particle.

3.2. Improve particle swarm optimization algorithm

In order to avoid particle swarm optimization falling into local optimization, the average particle distance and fitness variance are used as evaluation indexes to reinitialize particles [41–43]. During particle swarm initialization, the density of particles in the space is judged by the average particle spacing. If the average particle spacing is less than the set value, it means that the particle distribution is too dense and not evenly distributed in the space.

In the later stage of particle swarm optimization, the particle converges to a certain optimal solution, and the average particle spacing of the particle is small. The fitness variance is used to evaluate whether the particle converges to the nonlocal optimal solution. When the average particle spacing of the particle is small and the fitness variance is small, the particle loses its activity prematurely and falls into the local optimal solution. At this point, reinitialize the particles and make the particles search again.

3.2.1. Average particle distance

During the particle initialization of particle swarm optimization, the distribution of particles in space is random, which indicates that most particles may concentrate near a certain value and affect the final optimization result. In order to make the particles search the global space and prevent the particles from falling into the local optimal solution in the early stage, the concept of particle average particle distance is introduced in the process of parameter optimization with particle swarm optimization, which guides the distribution of the initial population of particles in the space of particle distribution. Its basic definition is shown in Eq (27):

$$D(t) = \frac{1}{|n||L|}\sum_{i=1}^{n}\sqrt{\sum_{j=1}^{Q}\left(p_{ij} - \overline{p_j}\right)^2} \tag{27}$$

In this formula: n is the number of particles of the particle swarm; L is the diagonal value of the search space during the particle search process, which can be expressed in the program as: $L = \sqrt{(200-0.1)^2 + (20-0.1)^2}$; Q is the dimension of the particle; p_{ij} represents the coordinate value of the j-th dimension of the i-th particle; $\overline{p_j}$ represents the average of all particles in the j-th dimension. The average particle distance represents the distribution of particles in the search space. The smaller the value of $D(t)$, the more dense the particles are; On the contrary, it indicates that the particles are more dispersed. Through the average particle distance, the particles can more fully optimize the parameters in the search space.

3.2.2. Fitness variance

In the process of particle iteration, the particles will approach towards the global optimal solution. At the initial stage of particle operation, the convergence speed of particles is fast. When the particles encounter the local extreme point, the particles may approach the local extreme point, resulting in the rapid decline of particle speed. Particle swarm optimization loses the ability of iterative evolution, and the algorithm falls into the local optimal solution and cannot jump out. Therefore, fitness variance is added to evaluate the concentration degree of particle fitness and judge the current state of particle population. If the number of current iterations is k, the fitness value of the i-th particle in this generation population can be represented by $f^k{}_i$（$i \in 1,2,\ldots,n$）, $\bar{\bar{f}}$ is the current average fitness value of the population, which can be expressed as:

$$\bar{\bar{f}} = \frac{1}{n}\sum_{i=1}^{n} f^k{}_i \tag{28}$$

$\sigma^2{}_k$ is the fitness variance of the k-th generation of the population, then $\sigma^2{}_k$ can be defined as:

$$\sigma^2{}_k = \sum_{i=1}^{n}(f^k{}_i - \bar{\bar{f}})^2 \tag{29}$$

Let the population fitness variance after introducing the normalization factor be σ^2,which is used to characterize the convergence degree of particles, then σ^2 can be defined as:

$$\sigma^2 = \sum_{i=1}^{n}\left(\frac{f^k{}_i - \bar{\bar{f}}}{f}\right)^2 \tag{30}$$

In this equation: f represents the normalized scaling factor, which is used to limit the size of variance, and is defined as:

$$f = \begin{cases} \max\left|f_i - \bar{\bar{f}}\right|, \max\left|f_i - \bar{\bar{f}}\right| > 1 \\ 1, \text{other} \end{cases} \tag{31}$$

Let the population reach the global extreme point p_{gd} in the t-th generation, that is

$$\lim_{t\to\infty} p_g(t) = p_{gd} \tag{32}$$

Obviously, if the global extremum is p_{gd}, when the particle is in $c_2\eta\left({p_{gd}}^k - {x_{id}}^k\right)$, that is, the third part of the particle flight speed update formula——the process of the particle learning the flight experience of the whole population, the particle will keep moving closer to p_{gd}, and its individual extreme value p will be updated continuously. If the particle does not find a better position than p_{gd}, the individual extreme value of the particle itself will be equal to p_{gd}, that is

$$\lim_{t\to\infty} p_i(t) = p_{gd} \tag{33}$$

The above formula shows that for any particle in the particle swarm, its final convergence position will be the optimal extreme value found by the whole particle swarm, and all particles will gather to this position.

Let the population reach the global optimum at t-th generation, that is, p_{gd} is the global optimum, and each particle converges to p_{gd} under the influence of "social cognitive ability", that is, the global

optimum. At this time, the fitness value of the particle is

$$f^k{}_i = f(p_{gd}) \tag{34}$$

Therefore, the average fitness of particles is

$$\bar{\bar{f}}^k = f(p_{gd}) \tag{35}$$

At this time:

$$\sigma^2 \to min \tag{36}$$

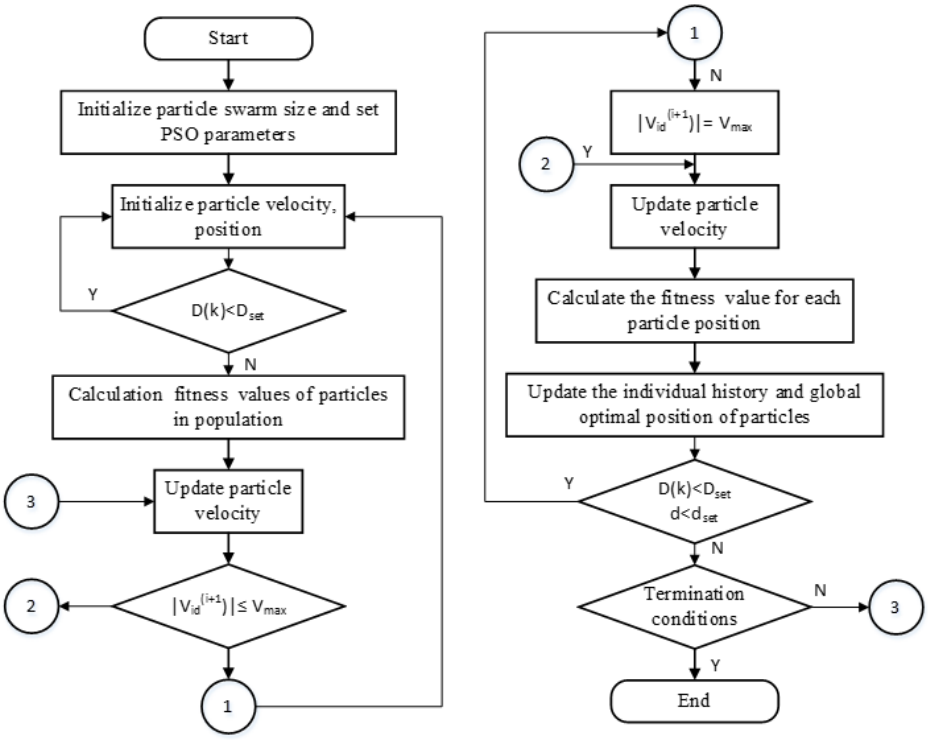


Figure 2. Improving the process of PSO optimization algorithm.

To sum up, in the process of particle iteration, the fitness of particles will get closer and closer, and the value of σ^2 will become smaller and smaller, which reflects the continuous aggregation of particles. In order to make each particle traverse the search space as much as possible and continuously update the global optimal solution of the population, a threshold is set in the algorithm. When the average particle distance $D(t)$ of the population and the fitness variance σ^2 are both less than the given threshold, it indicates that the particle has fallen into the global extreme point, the particle flight speed calculated from Eq (21) is very small, which makes the particle difficult to change the position significantly. At this time, v_{max} is forced to update the flight velocity of particles in the algorithm, and the n particles in the population are mutated by v_{max}, so that particles are separated from the global extremum point in different directions with v_{max}, and are redistributed in all positions of the search space. Since the particles have memory function, the optimal position found by the particles so

far is still stored in memory, and then the particles will search the solution space for a new round. In the search process, $D(t)$ and σ^2 begin to decrease again. When they are reduced to the given threshold, the algorithm will enter the mutation stage again. The whole evolutionary process of particles is to continuously search and mutate until the end of iterations to find the best position. The specific algorithm flow is shown in Figure 2.

4. Case simulation and analysis

4.1. Instance object and data processing

Table 1. Typical load data**.**

Date	Time	Average air temperature (°C)	Flow velocity (m/s)	Relative wind speed (m/s)	Average speed (kn)	Load power (kw)
2.14	0	7.2	0.73	2.21	8.56	802
2.14	2	7.1	0.75	2.23	8.51	821
…	…	…	…	…	…	…
2.14	23	7.6	0.82	2.45	7.84	865
2.15	0	7.5	0.74	2.25	8.25	793
…	…	…	…	…	…	…
3.27	15	13.3	0.58	1.25	8.62	856
…	…	…	…	…	…	…
3.27	23	9.3	0.63	0.95	8.15	816

In this paper, the 42-day load forecasting data of a 60 m Zhenyang steam ferry is selected to form the training data set of its load forecasting model. In the data set, weather information and load information on the day of ship operation are included. Weather information specifically includes relative flow velocity and relative wind velocity; The load information is collected every hour for the ship's load power information during the daily operation time. Select the data of a certain day as an example to display the load data, as shown in Table 1.

In the typical load data shown in Table 1, such as 2.14, the average temperature at time 0 is 7.2 °C, the flow velocity is 0.73 m/s, the relative wind speed is 2.21 m/s, the average sailing speed is 8.56 kn, and the total load is 802 kw. Among them, the load power of two propulsion motors is about 600kw, and the other loads such as ship lighting, control and pump are 202 kw in total. At time 23, the total demand power of the two propulsion motors is 625 kw, and the total demand power of the other ship lighting, control and other pump loads is 196 kw.

Due to the differences in units of different data types, in order to avoid the adverse impact of a certain data on the load forecasting results due to its magnitude, it is necessary to normalize the data. The data in Table 1 mainly includes temperature, flow velocity, relative wind speed, average speed and load data. Among them, the flow velocity data are already between 0 and 1, there is no need to normalize them. So it is only need to normalize the parameters affecting the ship's power load such as average temperature, relative wind speed, average sailing speed and load value [44].

1) Normalization of parameters affecting ship power load

Typical normalization of data：

$$T_{ij}{}' = \frac{T_{ij}-T_{jmin}}{T_{jmax}-T_{jmin}} \tag{37}$$

In this equation： $T_{ij}{}'$ represents normalized data; T_{ij} represents the original data; T_{jmin} represents the lowest value of any day in the collected data; T_{jmax} represents the highest value at any time in the collected data, $i = 1,2,\dots,n,\ j = 1,2,\dots,m.$

2) Normalization of load value

When normalizing the load data, the logarithmic processing method is adopted [45]:

$$x_{ij}{}' = \lg(x_{ij}) \tag{38}$$

In this equation： $x_{ij}{}'$ is the normalized load value; x_{ij} represents the original data at time i on day j, $i = 1,2,\dots,n,\ j = 1,2,\dots,m.$

After collecting and processing the historical load data of ships, the load forecasting model of ships can be fitted and regressed. In the establishment of load forecasting model, considering the continuity and periodicity of load, the input training characteristics of load are summarized.

Therefore, the selected input characteristics include the following : the average temperature, flow velocity, relative wind speed, average speed, load value at the same time of the previous day, the load value at the first two moments of the current forecast time of the forecast day, the load value at the first moment of the current forecast time of the forecast day, the average temperature, flow velocity, relative wind speed and average speed at the first moment of the forecast day. A total of 11 data are used as the input characteristics of training, and the output is the load value at the forecast time in the forecast day.

4.2. Simulation and analysis of LSSVM algorithm

When LSSVM is used for load forecasting, since the RBF kernel function has radial symmetry and good smoothness, and its expression is simple, especially in the case of multivariable input, it can also realize the mapping of input vector to high-dimensional space through nonlinear transformation and deal with the prediction problem of nonlinear relationship between input and output. Therefore, RBF is selected as the kernel function in the forecasting process [46], and two parameter values that the penalty parameter C and the standardized parameter σ of the kernel function need to be selected to carry out the regression of load data.

In this LSSVM training, C = 50, σ = 3 are selected, and the load data of dates 2.14 to 3.26 in Table 1 are selected as the training data set to predict the load data of dates 3.27. The results after operation are shown in Figure 3. In order to evaluate the prediction error of the prediction curve, the prediction error curve and the average relative error value are selected as the evaluation indexes, and the obtained curve is shown in Figure 4.

It can be seen from Figures 3 and 4 that the maximum error can reach 8%, the minimum error is around 2%, and the average relative error is about 3.2% when LSSVM is used for prediction.

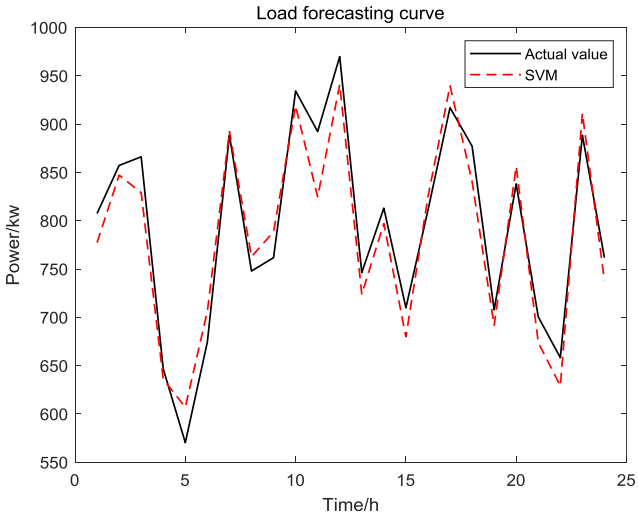


Figure 3. LSSVM load forecasting curve.

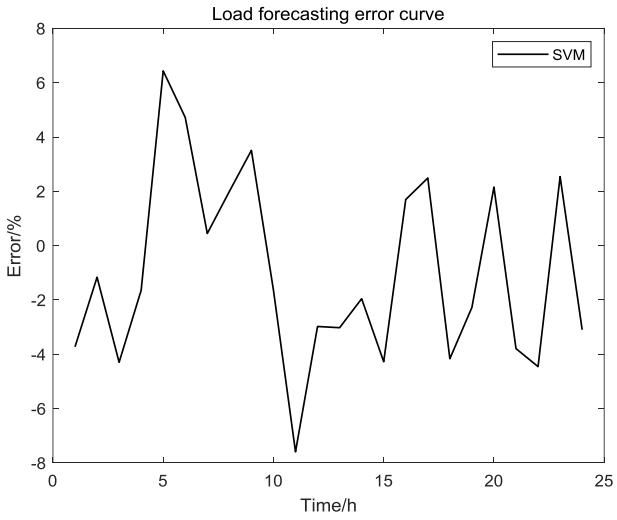


Figure 4. Error curve.

4.3. Simulation and analysis of PSO-SVM algorithm

The particle dimension is set to be 2D, the total number of particles is 50, the number of iterations is 10, the value range of ω is [0.4, 0.9], the value range of parameter C is [0.1, 200], and the value range of σ is [0.1, 20]. Taking the error as the evaluation index of fitness and running the LSSVM regression model optimized by particle swarm optimization, the function curve in Figures 5 and 6 can be obtained.

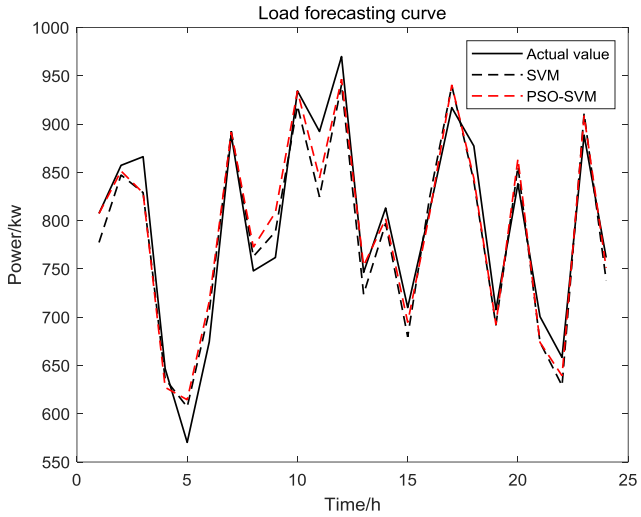


Figure 5. PSO-SVM load forecasting curve.

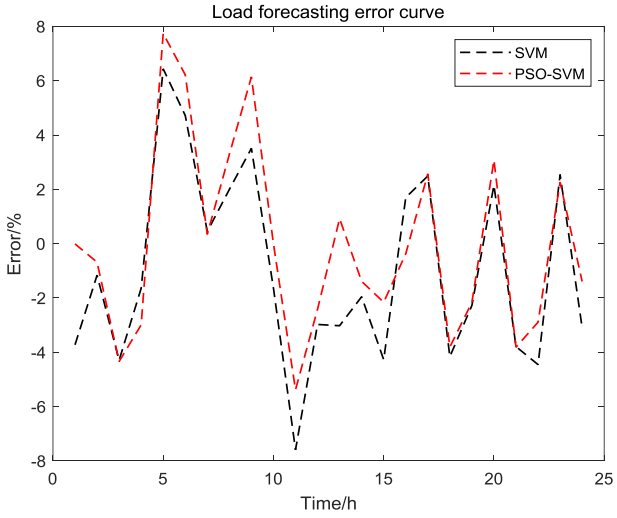


Figure 6. Load forecasting error curve.

It can be seen from Figures 5 and 6 that the regression model obtained by using PSO algorithm to optimize the two parameters of LSSVM is more accurate. It can be seen from the error curve in Figure 6 that although the maximum error of prediction increases, the minimum error of prediction can reach around 0. The average relative error also shows that the improved SVM algorithm based on PSO can obtain better prediction results. In addition, when the prediction model is established, online performance and offline performance indicators are added as the evaluation basis for optimization, and the expression is shown in Eq (39):

$$\begin{cases} OnLine(1,k) = \dfrac{sum(MeanAdapt(1,1{:}k))}{k} \\ OffLine(1,k) = \max(MeanAdapt(1,1{:}k)) \end{cases} \tag{39}$$

In this equation: $k = 1,2,\ldots,LoopCount$. $MeanAdapt(1,1{:}k)$ represents the average fitness value of each generation of particles from the first generation to the k-th generation.

At the beginning of optimization, the prediction error increases, the maximum average error of iteration can reach 2.87%, and the maximum average error of single iteration can reach 2.89%. Through continuous iteration, it can be seen that the error after iteration is getting smaller and smaller, and the average error is getting smaller and smaller, indicating that the optimization results are constantly optimizing.

4.4. Simulation and analysis of IPSO-SVM algorithm

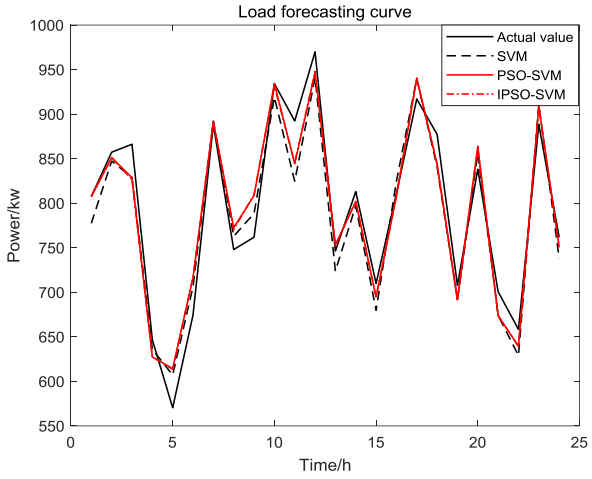


Figure 7. Load forecasting curve.

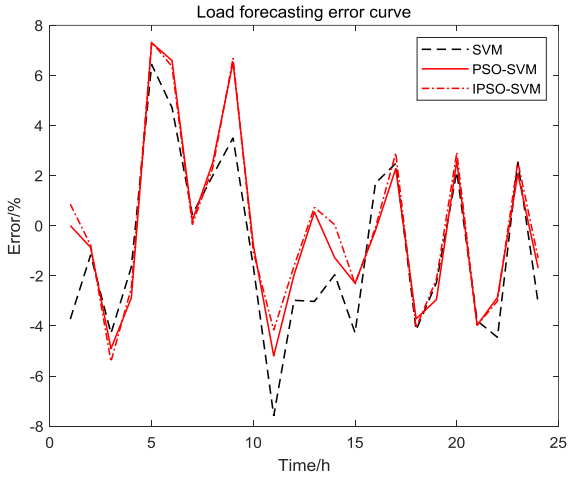
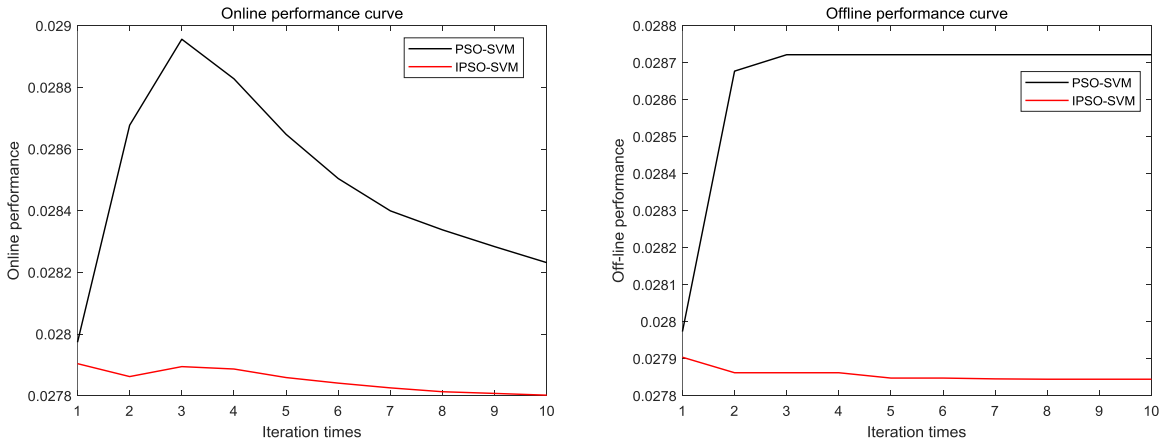


Figure 8. Load forecasting error.

IPSO-SVM can not only make the space optimization of particles sufficient, but also avoid particles falling into local optimal solution. The simulation diagram of load forecasting is shown in Figures 7 and 8.



a) Online performance curve. b) Offline performance curve.

Figure 9. Online and offline performance.

From the Figures 7–9, it can be seen that the accuracy of prediction has been further improved by adding the average particle distance and fitness variance to the particle swarm algorithm. At this time, the maximum error of prediction is about 8%, the minimum error is about 0, and the average relative error is reduced by about 0.013%. In the prediction iteration, the maximum average error of single iteration can reach 2.79%, and the maximum average error of iteration can reach 2.81%. In this paper, the two-dimensional optimization of two parameters is mainly realized, and the optimization range is small, and the effect is not obvious. It can be imagined that when the parameter dimension becomes larger and the optimization range becomes wider, the parameter optimization method will have more advantages. At the same time, it can be seen that after parameter optimization, the load prediction accuracy of IPSO-SVM is more accurate and the prediction error is smaller.

5. Conclusions

Load forecasting of ship power system is an important component of energy management of ship power system, which is related to the stability, security and economy of ship power system. In order to further reduce the error of load forecasting, the support vector machine model and method for ship power load forecasting are designed. The parameters such as temperature, flow velocity, relative wind speed and speed are innovatively used as the training eigenvalues of support vector machine, which strengthens the correlation between input characteristics and predicted output load and improves the prediction accuracy. Aiming at the problem that the regularization parameter C and the standardized parameter σ of the support vector machine are difficult to select, the particle swarm optimization algorithm is introduced to optimize. At the same time, the average particle distance and the fitness variance are used as the evaluation indexes. When the initial selection of particles is unreasonable and falls into local optimum, the particles are re-initialized to optimize to avoid particle swarm optimization falling into local optimum, and the accuracy of IPSO-SVM load forecasting is further improved.

Acknowledgments

This work was supported by the Zhoushan Science and Technology Project(2022C13034) and Jiangsu Key R & D plan (BE2018007).

Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. G. G. Box, G. M. Jenkins, G. C. Reinsel, Time series analysis: forecasting and control, *J. Time*, **31** (1976), 238–242. https://doi.org/10.1111/j.1467-9892.2009.00643.x
2. J. He, G. Wei, L. L. Xiong, Fuzzy improvement of linear regression analysis for load forecasting, *East China Electr. Power*, **11** (2003), 21–23.
3. C. B. Yuan, T. Zhang, J. L. Zhu, Short-term power load forecasting based on MRA and regression analysis, *Inf. Technol.*, **10** (2007), 88–90.
4. Q. Li, Z. Y. Wang, Z. R. Wang, Uncertainty evaluation for the dynamic calibration of pressure transducer, *J. Bei-jing Univ. Aeronaut. Astronaut.*, **41** (2015), 847–856. https://doi.org/10.13700/j.bh.1001-5965.2014.0356
5. S. L. Guo, Q. L. Shui, X. Y. Gu, Summary of application of gray system theory in load forecasting, *Ind. Instrum. Autom.*, **3** (2017), 24–27.
6. Y. Xue, N. Zhang, H. Wu, Z. Yu, R. Li, Short-term load forecasting method for user side microgrid based on UTCI-MIC and amplitude compression grey model, *Power Syst. Technol*.,**44** (2020), 556–563. https://doi.org/10.13335/j.1000-3673.pst.2019.1870
7. G. J. Zhang, J. J. Qiu, J. H. Li, Multi-factor short-term load forecasting based on fuzzy inference system, *Autom. Electr. Power Syst*., **26** (2002), 49–53.
8. S. K. Ha, K. B. Song, B. S. Kim, Short-term load forecasting for the holidays using fuzzy linear regression method, *IEEE Trans. Power Syst*., **20** (2005), 96–101. https://doi.org/10.1109/TPWRS.2004.835632
9. L. Feng, J. J. Qiu, Short-term load forecasting for anomalous days based on fuzzy multi-objective genetic optimization algorithm, *Proc. CSEE*, **25** (2005), 29–34. https://doi.org/10.1109/PES.2006.1708902
10. X. AI, Z. Y. Zhou, Y. P. Wei, H. Zhang, L. Li, Bidding strategy of transferable load based on autoregressive integrated moving average model, *Autom. Electr. Power Syst.*, **41** (2017), 26–31. https://doi.org/10.7500/AEPS20170119009
11. L. L. Liu, *Short-term power load forecasting based on SARI-MA and SVR*, Ph.D thesis, East China University of Technology, 2018.
12. X. Y. Wu, J. H. He, P. Zhang, J. Hu, Power system short-term load forecasting based on improved random forest with grey relation projection, *Autom. Electr. Power Syst.*, **39** (2015), 50–55. https://doi.org/10.7500/AEPS20140916005
13. N. T. Huang, G. B. Lu, D. G. Xu, A permutation importance-based feature selection method for short-term electricity load forecasting using random forest, *Energies*, **9** (2016), 767. https://doi.org/10.3390/en9100767
14. Y. C. Li, T. J. Fang, E. K. Yu, Study of support vector machine method for short-term load forecasting, *Proc. CSEE*, **6** (2003), 55–59. https://doi.org/10.1109/ICNC.2007.659
15. D. F. Zhao, W. C. Pang, J. S. Zhang, X. F. Wang, Based on Bayesian theory and online learning SVM for short term load forecasting, *Proc. CSEE*, **25** (2005), 8–13. https://doi.org/10.1109/ICNC.2007.659

16. D. F. Zhao, X. F. Wang, L. Zhou, T. Zhang, D. Z. Xia, Short-term load forecasting using radial basis function networks and expert system, *J. Xi'an Jiaotong Univ.*, **4** (2001), 331–334.
17. H. D. Zhao, *Research on intelligent power short-term load forecasting system based on fuzzy expert system*, Master thesis, South China University of Technology, 2001.
18. L. Li, J. Wei, C. B. Li, Y. Cao, B. Fang, Prediction of load model based on artificial neural network, *Trans. China Electrotech. Soc*., **30** (2015), 225–230. https://doi.org/10.19595/j.cnki.1000-6753.tces.2015.08.028
19. J. R. Zhang, Research on power load forecasting based on the improved elman neural network, *Chem. Eng. Trans*., **51** (2016), 589–594. https://doi.org/10.3303/CET1651099
20. R. Hu, S. Wen, Z. Zeng, T. Huang, A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm, *Neurocomputing*, **221** (2017), 24–31. https://doi.org/10.1016/j.neucom.2016.09.027
21. X. Dai, C. Yang, S. Huang, T. Yu, Y. Zhu, Finite time blow-up for wave equation with dynamic boundary condition at critical and high energy levels in control systems, *Electron. Res. Arch.*, **28** (2020), 91–102. https://doi.org/10.3934/era.2020006
22. X. Q. Dai, Global existence of solution for multidimensional generalized double dispersion equation, *Boundary Value Probl.*, **1** (2019), 1–4. https://doi.org/10.1186/s13661-019-1266-1
23. X. Q. Dai, W. K. Li, Non-global solution for visco-elastic dynamical system with nonlinear source term in control problem, *Electron. Res. Arch.*, **29** (2021), 4087–4098. https://doi.org/10.3934/era.2021073
24. N. Y. Liu, F. Mu, Short-term power load forecasting based on least squares support vector machine optimized by NRS and PSO algorithm, *Mod. Electr. Tech.*, **42** (2019), 115–118. https://doi.org/10.16652/j.issn.1004-373x.2019.07.028
25. W. L. Gong, *Short-term load forecasting based on least squares support vector machine*, Master thesis, Hunan University, 2014.
26. X. G. Zhang, About statistical learning theory and support vector machine, *J. Autom*., **26** (2000), 32–42. https://doi.org/10.16383/j.aas.2000.01.005
27. T. Liu, Y. Wang, W. Liu, Research on Least Squares Support Vector Machine Combinatorial Optimization Algorithm, in *2009 International Forum on Computer Science-Technology and Applications,* (2009), 452–454. https://doi.org/10.1109/IFCSTA.2009.116
28. W. D. Chang, An improved PSO algorithm for solving nonlinear programing problems with constrained conditions, *Int. J. Model. Simul. Sci. Comput.*, **12** (2021), 2150001. https://doi.org/10.1142/S179396232150001X
29. H. Yang, *SVM kernel parameter optimization research and application*, Master thesis, Zhejiang University, 2014.
30. J. Dong, Y. Zhao, C. Liu, Z. F. Han, C. S. Leung, Orthogonal least squares based center selection for fault-tolerant RBF networks, *Neurocomputing*, **339** (2019), 217–231. https://doi.org/10.1016/j.neucom.2019.02.039
31. Y. Bai, *Kernel function-based interior-point algorithms for conic optimization*, Science Press, 2010.
32. W. Jiang, *SVM parameter optimization and application based on improved particle swarm optimization*, P. D. thesis, Jiangsu University of Science and Technology, 2020.
33. P. W. Li, J. Zhao, Intelligent single particle optimization and particle swarm optimization fusion algorithm, *Int. J. Appl. Math. Stat.*, **45** (2013), 395–403.

34. J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.
35. Y. Shi, R. Eberhart, Empirical study of particle swarm optimization, in *International Conference on Evolutionary Computation*, **3** (1999), 1945–1950. https://doi.org/10.1109/CEC.1999.785511
36. J. D. Tang, X. Y. Xiong, Y. W. Wu, Reactive power optimization of power system based on improved PSO algorithm, *Power Autom. Equip.*, **7** (2004), 81–84.
37. J. D. C. Little, The use of storage Water in a Hydroelectric System, *Oper. Res.*, **3** (1995), 187–197. https://doi.org/10.1287/opre.3.2.187
38. K. De Jong, *An analysis of the behavior of a class of genetic adaptive systems*, P. D. thesis, University of Michigan, 1975.
39. J. X. Hu, J. H. Zeng, Adjustment strategy of inertia weight in particle swarm optimization, *Comput. Eng.*, **11** (2007), 193–195.
40. Y. Shi, R. Eberhart, A modified particle swarm optimizer, in *IEEE International Congress on Evolutionary Computation Proceedings*, (1998), 69–73. https://doi.org/10.1109/ICEC.1998.699146
41. H. Y. Liu, Y. E. Lin, J. S. Zhang, Hybrid particle swarm optimization algorithm based on chaotic search to solve premature convergence, *Comput. Eng. Appl.*, **42** (2006), 77–79.
42. Z. S. Lu, Z. R. Hou, Adaptive mutation particle swarm optimization algorithm, *J. Electr.*, **32** (2004), 416–420.
43. X. L. Zhang, S. H. Wen, H. N. Li, Q. Lu, M. Wu, X. Wang, Chaos particle swarm optimization algorithm based on Tent mapping and its application, *China Mech. Eng.*, **19** (2008), 2108–2112.
44. J. P. He, *Application of support vector machine in short-term power load forecasting*, Master thesis, Three Gorges University, 2014.
45. D. Wang, *Short-term load forecasting of power system based on improved least squares support vector machine*, Master thesis, Xi'an University of Technology, 2015.
46. H. B. Wang, *Electric propulsion ship load forecasting research*, Master thesis, Jiangsu University of Science and Technology, 2013.