Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5885235, 15 pages
https://doi.org/10.1155/2022/5885235

Research Article

Improvement and Application of Fractional Particle Swarm Optimization Algorithm

Jing Li and Chunna Zhao

School of Information Science and Engineering, Yunnan University, Kunming 650500, China

Correspondence should be addressed to Chunna Zhao; zhaochunna@ynu.edu.cn

Received 25 February 2022; Revised 7 April 2022; Accepted 11 April 2022; Published 26 May 2022

Academic Editor: Nagarajan Deivanayagampillai

The convergence performance of existing fractional particle swarm optimization algorithm directly depends on a single fractional-order operator. When its value increases, the convergence speed of particles gets slower. When its value decreases, the probability of the particle swarm trapping into the local optimum increases. In order to solve this problem, an improved fractional particle swarm optimization (IFPSO) algorithm is proposed in this paper. New variables are introduced in this paper to redefine the formula. The fractional-order operator is added to the velocity update formula and the position update formula. The IFPSO algorithm has linearly decreasing inertia weights. Then the proposed IFPSO algorithm is used to optimize the parameters of support vector machine (SVM) and the clustering center of the K-means classifier is selected. Experimental results show that the IFPSO algorithm can effectively avoid falling into the local optimal solution. It has a faster convergence rate and better stability than the original algorithm, which proves the effectiveness of the algorithm. Examples verify that the IFPSO algorithm can improve the classification accuracy of SVM in practical application. The IFPSO algorithm effectively solves the problem that K-means algorithm is overly dependent on the initial center and may have empty classes.

1. Introduction

Particle swarm optimization (PSO) algorithm originated from the simulation of the foraging behavior of birds. It is widely used to solve multiobjective optimization problems because of simple concept, easy implementation, and fast convergence. However, the PSO algorithm still faces many problems such as slow local convergence speed, trapping into a local optimum in the multimodal function, and uneven distribution of the obtained solutions. Therefore, many scholars are committed to improving the performance of the PSO algorithm. Krohling [1] introduced the Gaussian function into the PSO algorithm and proposed the Gaussian particle swarm optimization (GPSO) algorithm. GPSO does not require inertial weight, and the acceleration factor is generated by a random number that obeys the Gaussian distribution. It can overcome the problem that the searchability and convergence performance of the traditional PSO algorithm depends on a large extent on the acceleration factor and inertial weight setting. Tillett et al. [2] proposed an algorithm that uses the evolutionary ideas of natural selection, called Darwinian particle swarm optimization (DPSO), which dynamically divides the population into several subgroups, and each subgroup searches independently to increase the diversity of particles. It can enhance the global optimization capability of the algorithm. Solteiro Pires et al. [3] introduced fractional calculus into PSO and proposed a fractional order particle swarm optimization (FOPSO) algorithm, which controls the convergence speed of the algorithm by introducing fractional operators to the velocity formula of the particle swarm. On this basis, Couceiro et al. [4] proposed a fractional order Darwinian particle swarm optimization (FODPSO) algorithm to control the convergence speed of the DPSO algorithm. Experiments showed that the FODPSO algorithm is superior to the basic PSO, DPSO, and FOPSO algorithms in terms of calculation accuracy and convergence speed. But, like the FOPSO algorithm, the convergence performance of the FODPSO algorithm also directly depends on the fractional operators α.

In response to this shortcoming, based on the FOPSO algorithm, an improved fractional particle swarm optimization (IFPSO) algorithm is proposed in this paper. First, a new fractional operator is introduced into the velocity formula, and a linearly decreasing strategy is adopted for the inertia weight factor. Based on this, a new update formula of particle swarm optimization is constructed, and the comparison results of multiple sets of test functions show that the new optimization algorithm enhances the particle diversity. The improved algorithm shows the controllability of local convergence and the efficiency of global convergence and improves the ability of the algorithm to capture the global optimal solution. In this paper, the IFPSO algorithm is applied to the parameter optimization of support vector machines and K-means. The effectiveness and stability of the IFPSO algorithm are verified by simulation experiments.

Inspired optimization algorithms have been used in a variety of fields (such as image detection [5–7], image segmentation [8, 9], parameter optimization [10–13], PID control [8, 14], feature selection [15], scheduling problem [16, 17], K-means (KM) [18, 19], and clustering [20, 21]). The improved PSO algorithms are also used in classification, such as adaptive particle swarm optimization for parameter optimization in classification models [22] and multiobjective particle swarm optimization approach for feature selection in classification [23]. At the same time, the medical decision support system based on machine learning has a good development prospect as described in the articles in [24, 25]. In particular the support vector machine (SVM) training results are stable, and the number of samples required is relatively small. However, its prediction accuracy is not high enough; the improved fractional particle swarm optimization algorithm is applied into optimizing the parameters of SVM in this paper. The IFPSO-SVM is constructed in heart disease prediction model and the red wine classification prediction model. Simulation experiments verify the superiority of the model based on IFPSO-SVM in terms of diagnosis efficiency and accuracy. Diagnosis error is reduced significantly and prediction results have certain practical significance.

The IFPSO algorithm is used to optimize the clustering center of the KM clustering. It effectively improves the K-means algorithm's problems about excessive dependence on the selection of the initial center and sensitivity to noise data. Simulation experiments also confirm that the classification effect of the IFPSO-KM algorithm is more accurate and stable than K-means and PSOK-means, and there is no empty class phenomenon, the classification accuracy is higher, and the clustering effect is relatively stable.

The rest of this paper is organized as follows. In Section 2, firstly the definition and properties of particle swarm optimization algorithm with a linearly decreasing inertia weight are introduced. Then fractional calculus is given briefly. Lastly, the IFPSO algorithm is derived in this part. Section 3 describes the application examples of IFPSO-SVM. Classification method based on IFPSO-KM algorithm is described in Section 4. Experiments are illustrated and analyzed in Section 5. The conclusions are illustrated in Section 6.

2. Improved Fractional Particle Swarm Optimization Algorithm

This section briefly introduces the inertial weighted particle swarm optimization algorithm and fractional calculus, and then the IFPSO algorithm is derived in this part.

2.1. Weight Particle Swarm Optimization (WPSO) Algorithm.

The standard integer-order PSO is inspired by the swarm's behavior, where the particles have a synchronized motion during maneuvers such as for searching food and for defense. The $(k+1)-th$ iteration of the velocity of the id − th particle V_{id} can be determined as

$$V_{id}^{k+1} = w \times V_{id}^{k} + c_1 \times r_1 \times \left(P_{id}^{k} - X_{id}^{k}\right) + c_2 \times r_2 \times \left(P_{g\,d}^{k} - X_{id}^{k}\right), \tag{1}$$

where P_{id}^{k} is its best position found so far in $k-th$ iteration, $P_{g\,d}^{k}$ is the best position found by neighbourhoods of the current particle, and the random values r_1 and r_2 are between $[0,1]$. c_1 and c_2 are the cognitive and social coefficients that are used to determine the effects of individual and group experience on the particle trajectory. w is the inertia weight factor.

After calculating the velocity, the new position of each particle can be computed as

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1}. \tag{2}$$

If the inertia weight w in PSO algorithm is set to a fixed value, oscillations are likely to occur around the global optimal solution in the later stage of convergence. To solve this phenomenon, the paper in [26] uses the method of linearly decreasing inertia weight in the PSO algorithm to control the velocity from exploding, and the formula of the WPSO algorithm is expressed as

$$w(k) = w_{\max} - \frac{\left(w_{\max} - w_{\min}\right) * k}{k_{\max}}, \tag{3}$$

where k is the current iteration number, $k_{\max}$ is the maximum iteration number, $w_{\max}$ is the initial inertia weight value, and $w_{\min}$ is the final inertia weight value.

2.2. Fractional Calculus.

Fractional calculus is the theory of any order of differentiation and integration, which is derived from the generalization of integer-order calculus. It is closer to the actual situation of the application background. The fractional differential operator is a nonlocality overall operator [27]. The integer differential is only related to the current sampling point and the sampling value of the previous sampling point at the current moment, while the fractional differential is related to the values of all previous sampling points related. The fractional derivative of order α ($\alpha \in R$), also known as the Grunwald-Letnikov (G-L) definition, defined by the series, can be expressed as

$$D^{\alpha} f(x) = \lim_{h\longrightarrow 0}\left[\frac{1}{h^{a}}\sum_{k=0}^{+\infty}\frac{(-1)^{k}\Gamma(\alpha+1)f(x-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)}\right]. \tag{4}$$

In the formulation of fractional PSO, the discrete-time approximation is utilized in equation (4); we have

$$D^{\alpha} f(x) = \frac{1}{T^{a}}\sum_{k=0}^{r}\frac{(-1)^{k}\Gamma(\alpha+1)f(x-kh)}{\Gamma(k+1)\Gamma(\alpha-k+1)}, \tag{5}$$

where T is the sampling period, while r is the truncation order. $\Gamma(\bullet)$ represents the standard Euler gamma function, which is mathematically defined by the following relation for a convergent improper integral: $\Gamma(z) = \int_0^{\infty} e^{-t}t^{z-1}\mathrm{d}t$.

According to the definition of GL, it can be found that the integer-order differential is only a special case of the fractional-order differential. For the fractional calculus of a continuous function at a certain point, not only is it to find the limit at that point but also it is related to the value of the function at all times from the initial moment to the point, so the fractional calculus has memory.

2.3. Improved Fractional Particle Swarm Optimization (IFPSO) Algorithm. Since the particle swarm algorithm is based on the long-term dynamic evolution of individual biological populations, the inherent long memory characteristic of the fractional-order model is suitable to describe the optimization process of particle swarms. Solteiro Pires et al. [3] proposed the FOPSO algorithm. The algorithm corrects the order of the velocity derivative by rearranging the original velocity of the particle swarm to control the convergence speed of the algorithm. However, the algorithm does not consider the influence caused by the fixed value of the inertia weight factor. The convergence performance of the existing fractional particle swarm optimization directly depends on the fractional-order α. When the value α increases, the convergence speed of particles becomes slower, and when the value α decreases, the probability of the particle swarm trapping into the local optimum becomes higher. In this paper, new variables are added, and fractional derivation is considered for both velocity and position formulas at the same time to reduce the dependence of the update formula on α, and the inertia weight factor w uses the principle of linear decrease. This method reduces the oscillation while increasing the randomness of the particles and also reduces the probability of the articles falling into a local optimum solution.

According to equation (1), the following can be obtained by transforming left and right:

$$V_{id}^{k+1} - V_{id}^{k} = (w-1)\times V_{id}^{k} + c_1\times r_1\times\left(P_{id}^{k} - X_{id}^{k}\right) + c_2\times r_2\times\left(P_{g\,d}^{k} - X_{id}^{k}\right), \tag{6}$$

where $V_{id}^{k+1} - V_{id}^{k}$ is the fractional derivative of the discrete state and, assuming that the sampling period $T = 1$, equation (6) can be extended to the fractional-order derivative:

$$D^{a}\left[V_{id}^{k+1}\right] = (w-1)\times V_{id}^{k} + c_1\times r_1\times\left(P_{id}^{k} - X_{id}^{k}\right) + c_2\times r_2\times\left(P_{g\,d}^{k} - X_{id}^{k}\right). \tag{7}$$

Due to the memory characteristics of fractional calculus and considering that the relationship between the particles in the current iteration and the particles of the first few generations has gradually faded, we choose to keep the vector in the current 4 generations (the truncation order $r = 4$) and let $T = 1$. Using the G-L derivative (equation (5)) to extend the speed formula of the particle swarm algorithm from the first order to any order, one has

$$\begin{aligned} D^{a}\left[V_{id}^{k+1}\right] = V_{id}^{k+1} - aV_{id}^{k} - \frac{1}{2}a(1-a)V_{id}^{(k-1)} - \frac{1}{6}a(1-a)(2-a)V_{id}^{(k-2)} \\ - \frac{1}{24}a(1-a)(2-a)(3-a)V_{id}^{(k-3)}. \end{aligned} \tag{8}$$

Combining equations (7) and (8), we obtain the speed formula of the fractional-order particle swarm algorithm with linearly decreasing weight factors:

$$\begin{aligned} V_{id}^{k+1} =& (w-1+a)V_{id}^{k} + \frac{1}{2}a(1-a)V_{id}^{(k-1)} + \frac{1}{6}a(1-a)(2-a)V_{id}^{(k-2)} \\ &+ \frac{1}{24}a(1-a)(2-a)(3-a)V_{id}^{(k-3)} + c_1 \times r_1 \times \left(P_{id}^{k} - X_{id}^{k}\right) + c_2 \times r_2 \times \left(P_{g\,d}^{k} - X_{id}^{k}\right). \end{aligned} \tag{9}$$

Through the introduction of the fractional differential operator, the current particles are connected with the velocity of the particles of the previous stage, which makes the algorithm have memory. Next, the position formula is also improved in fractional order, and the term of equation (2) can be shifted to obtain

$$X_{id}^{k+1} - X_{id}^{k} = V_{id}^{k+1}. \tag{10}$$

Using the G-L definition to extend it to the fractional differential, we get

$$D^{\beta}\left[X_{id}^{k+1}\right] = V_{id}^{k+1}. \tag{11}$$

When $\beta \neq 1$, choose to keep the current 4 generation vectors and take $T = 1$; using the G-L definition (equation (5)) to extend the position formula of the particle swarm algorithm from the first order to any order, we have

$$\begin{aligned} D^{\beta}\left[X_{id}^{k+1}\right] =& X_{id}^{k+1} - \beta X_{id}^{k} - \frac{1}{2}\beta(1-\beta)X_{id}^{k-1} - \frac{1}{6}\beta(1-\beta)(2-\beta)X_{id}^{k-2} \\ &- \frac{1}{24}\beta(1-\beta)(2-\beta)(3-\beta)X_{id}^{k-3}. \end{aligned} \tag{12}$$

Combining equations (11) and (12) can get the position update formula of the fractional particle swarm algorithm:

$$\begin{aligned} X_{id}^{k+1} =& V_{id}^{k+1} + \beta X_{id}^{k} + \frac{1}{2}\beta(1-\beta)X_{id}^{k-1} + \frac{1}{6}\beta(1-\beta)(2-\beta)X_{id}^{k-2} \\ &+ \frac{1}{24}\beta(1-\beta)(2-\beta)(3-\beta)X_{id}^{k-3} \\ =& (w-1+a)V_{id}^{k} + \frac{1}{2}a(1-a)V_{id}^{(k-1)} + \frac{1}{6}a(1-a)(2-a)V_{id}^{(k-2)} \\ &+ \frac{1}{24}a(1-a)(2-a)(3-a)V_{id}^{(k-3)} + c_1 \times r_1 \times \left(P_{id}^{k} - X_{id}^{k}\right) \\ &+ c_2 \times r_2 \times \left(P_{g\,d}^{k} - X_{id}^{k}\right) + \beta X_{id}^{k} + \frac{1}{2}\beta(1-\beta)X_{id}^{k-1} \\ &+ \frac{1}{6}\beta(1-\beta)(2-\beta)X_{id}^{k-2} + \frac{1}{24}\beta(1-\beta)(2-\beta)(3-\beta)X_{id}^{k-3}. \end{aligned} \tag{13}$$

It can be seen from equation (13) that the position of the particle is no longer only affected by the fractional-order α. The introduction of the fractional-order β makes the updated position also have memory and related to the previous position, and due to the inertia weight factor w uses a linearly decreasing strategy, which protects the diversity of particles.

3. Prediction Model Based on IFPSO-SVM

In this section, the improved fractional particle swarm optimization algorithm is applied to the parameter optimization of SVM, and the radial basis function (RBF) is used as the kernel function. The improved fractional particle swarm optimization algorithm is used to determine the appropriate error penalty factor and the parameters of the kernel function.

3.1. Parameter Optimization of SVM. SVM is a unified framework constructed based on the principle of structural risk minimization to solve the problem of small sample learning. It can learn the optimal prediction results under limited information conditions and can better solve the classification problems of nonlinearity, overlearning, high latitude, and so forth and has good generalization ability. The main point of SVM is to map data to a high-dimensional space with a kernel function. Its classification performance depends on the selection of kernel function type, the setting of kernel function parameters, and the error penalty factor. The setting of the error penalty factor C adjusts the ratio of the confidence range of the learning machine to the nuclear experience risk and ensures that the trained SVM has a better generalization ability. Therefore, choosing the appropriate parameters (C and σ) affected the precision of the SVM significantly. However, there is no mature theory to guide the parameter optimization of SVM. The parameter optimization problem limits the classification effect of SVM in practical applications. Traditional SVM parameter optimization methods mainly include experimental methods, grid search methods, and gradient descent methods [28, 29]. These methods involve a large number of calculations and are easy to fall into local optimal solutions and can no longer meet the current requirements in terms of solving speed and accuracy. In recent years, with the rise of artificial intelligence algorithms, many scholars have applied a variety of metaheuristic algorithms to the parameter optimization problem of SVM, such as the genetic algorithm and the PSO algorithm [30]. However, these optimization methods also have defects such as being easy to fall into local optimum solutions, slow convergence speed, and insufficient accuracy. Therefore, this paper considers using the IFPSO algorithm to encapsulate SVM to build a prediction model, aiming to improve the prediction accuracy and stability of the model.

The RBF is as follows:

$$K(x_i, y_i) = \exp\left(-\frac{\|x_i - y_i\|^2}{\sigma^2}\right), \sigma \text{ is the nuclear parameter.} \tag{14}$$

The value of the parameter affects its learning ability and generalization ability. From equation (14), it can be found that the radial basis function only needs to determine one parameter, which is beneficial to parameter optimization. The paper in [31] also believes that RBF is a more suitable kernel function. Therefore, this paper selects RBF as the kernel function and then uses the IFPSO algorithm to find the best parameter combination (σ, C) for SVM.

3.2. Predictive Model Algorithm.

In this model, the SVM uses RBF as the kernel function, uses the IFPSO algorithm to optimize the parameters of the SVM, determines the appropriate error penalty factor C and kernel function parameter σ, and then builds a complete prediction model. The steps of the algorithm are as follows:

Step 1: Set the individual extremum of each particle as the current position, use the fitness function to calculate the fitness value, and take the individual extremum with the best fitness value as the global extremum.

Step 2: Calculate and update the current inertia weight factor according to the weight update formula (equation (3)).

Step 3: Calculate according to the velocity formula (equation (9)) and position update formula (equation (13)) of the improved particle swarm algorithm to update the velocity and position of the particles.

Step 4: Use the fitness function of the particles to calculate the fitness value of each particle after each iteration. Compare the fitness value of each particle with its individual extreme value *pbest*; if the fitness value is better, update the individual extreme value; otherwise, keep the original value; compare the updated individual extreme value of each particle with the global extreme value *gbest*; if the updated individual extreme value is better, update the global extreme value; otherwise, keep the original value.

Step 5: Judge whether the iteration value and accuracy range match the optimization conditions. If they are satisfied, this iteration ends, and the next iteration is performed; otherwise, go back to step 3.

Step 6: After the iteration, use the best optimized parameters (σ,C) to establish IFPSO-SVM prediction model.

The algorithm flow chart of the IFPSO-SVM prediction model is shown in Figure 1.

In the particle update process, the inertia weight factor decreases linearly. The new fractional-order update formula is used when updating the velocity and position of the particle. In subsequent experiments, the genetic algorithm mutation operations were added to the particles to increase the randomness and diversity of the particles. The improved fractional particle swarm optimization algorithm is used to optimize the SVM parameters to improve the prediction accuracy of the model.

4. Clustering Model Based on IFPSO-KM

This section uses the improved fractional particle swarm algorithm to select the clustering centers of the KM algorithm. The model effectively solves the excessive dependence of the KM algorithm on the selection of the initial clustering centers.

4.1. Improvement Idea of K-Means Clustering.

The basic principle of the KM algorithm is a clustering algorithm based on partition. Given a dataset X and the total number of data samples n, first randomly select k initial clustering centers (cluster points), and assign each object to the nearest cluster point to get a set of clusters. Then calculate the average value of each cluster as the new cluster point, and redistribute each data sample to the nearest aggregation point. This process is performed in a loop until the termination condition is met and the algorithm ends. In the KM algorithm, the Euclidean distance formula is used to measure the distance, which is the square root of the sum of the squares of the difference of each attribute. The most commonly used objective function is $f = \sum_{j=1}^{k} \sum_{x_i \in C_j} d(x_i, z_j)$, where x_i is the i-th object and z_j is the center of the j-th cluster. The purpose of clustering is to find a set of cluster centers which minimizes the above objective function value.

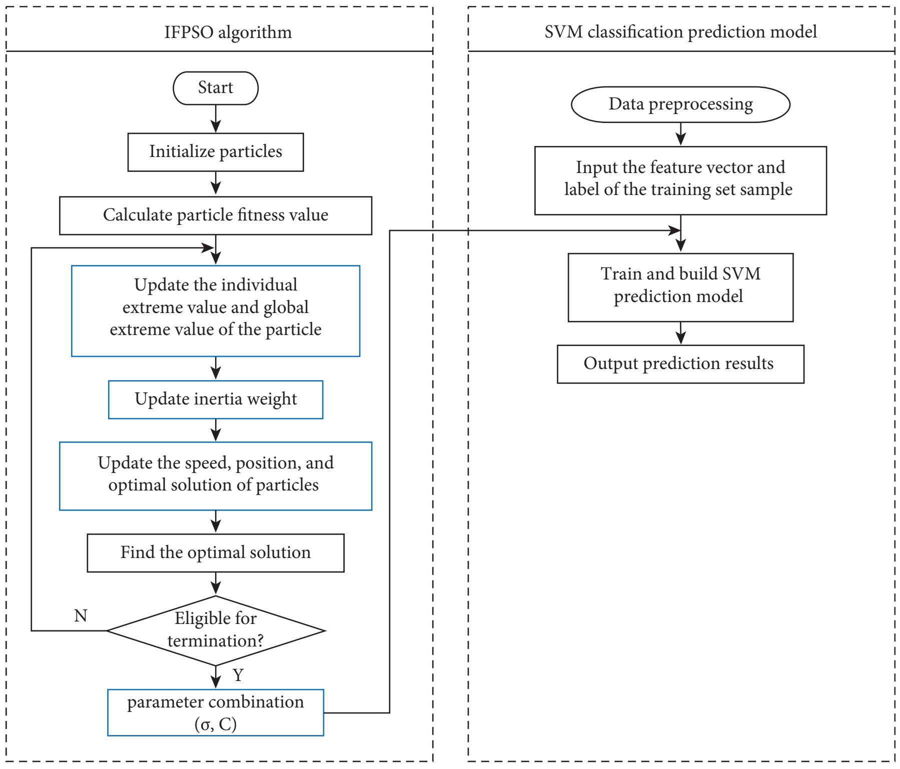
IFPSO algorithm
Start
Initialize particles
Calculate particle fitness value
Update the individual extreme value and global extreme value of the particle
Update inertia weight
Update the speed, position, and optimal solution of particles
Find the optimal solution
Eligible for termination?
N
Y
parameter combination (σ, C)
SVM classification prediction model
Data preprocessing
Input the feature vector and label of the training set sample
Train and build SVM prediction model
Output prediction results

Figure 1: Algorithm flow chart of the IFPSO-SVM prediction model.

However, due to the existence of local extreme points and the greedy nature of the heuristic algorithm, the traditional KM algorithm has the shortcomings of being sensitive to the initial clustering center and easily converging to the local extreme value. In this paper, the IFPSO algorithm is used to find cluster centers, and mutation operation is added in the particle update process, which increases the diversity of particles and dynamically adjusts particles. It avoids the phenomenon of premature convergence of particles in the PSO algorithm and has strong global search capabilities. It also eliminates the dependence of the KM algorithm on the selection of the initial cluster centers. Through the algorithm, the global optimal N cluster centers are obtained and then used as the initial cluster centers to implement the improved algorithm to obtain the ideal cluster division.

4.2. Principle of IFPSO-KM Algorithm. Given a sample set X, the total number of data samples is n, which is divided into k clusters to form a partition $C = \{C_1, C_2, \ldots, C_k\}$. For each cluster center, calculation formula and sample set are obtained. The total cumulative dispersion and formula are as follows:

$$z_{rj} = \frac{1}{n_j} \sum_{x_i \in C_j} x_i. \tag{15}$$

$$f(Z_r) = \sum_{j=1}^{k} \sum_{x_i \in C_j} d\left(x_i, z_{rj}\right). \tag{16}$$

The fitness value of the particle is calculated according to equation (16), where Z_r is the r-th particle $(1 \le r \le N)$, N is the number of particles, and z_{rj} is the cluster center position of the j-th class of the r-th particle. $d(x_i, z_{rj})$ is the distance from the i-th sample data to the corresponding cluster center. The clustering criterion function $f(Z_r)$ is the sum of the distances from various samples to the corresponding cluster centers and is also the fitness function of the particles. The particle velocity and position update formula are as follows:

$$\begin{aligned} V_r^{(t+1)} =& (w - 1 + a)V_r^t + \frac{1}{2}a(1-a)V_r^{(t-1)} + \frac{1}{6}a(1-a)(2-a)V_r^{(t-2)} \\ &+ \frac{1}{24}a(1-a)(2-a)(3-a)V_r^{(t-3)} + c_1 \times r_1 \times \left(P_r^t - Z_r^t\right) + c_2 \times r_2 \times \left(P_g^t - Z_r^t\right) \\ &+ c_3 \times r_3 \times \left(P_g^t - P_r^t\right). \end{aligned} \tag{17}$$

```
Input: data set $X$
Output: divided $k$ cluster center solution sets
Step 1: Initialize the population and parameters, determine the $k$ value according to the SSE and Silhouette Coefficient, randomly select $k$ data sample points as the initial center point, and calculate the fitness value of each particle according to the formula.
Step 2: Compare the fitness value of each particle with its individual extreme value $P_r$, and update $P_r$ if the new value is better.
Step 3: Compare the fitness value of each particle with the population extreme value $P_g$, and update $P_g$ if the new value is better.
Step 4: According to the velocity formula (equation (17) and position formula (equation (18)) the velocity and position of the particles are updated respectively. At the same time, the experiment uses the basic idea of GA to randomly perform mutation operations on some particles, randomly encode particles within the set number of dimensions and perform single-point mutation operations to generate new groups. Then recalculate the fitness value of the particles, and update the fitness value according to equtaion (16).
Step 5: According to the update process of the above algorithm, the optimal cluster center point is generated.
Step 6: Using the nearest neighbor rule in the KM algorithm, reclassify each sample to obtain a new cluster division.
Step 7: Determine whether the termination conditions are met. If it is satisfied, output the optimal solution; otherwise, return Step 2. The termination condition is that the cluster centers obtained within a given number of iterations do not change, or the maximum number of iterations is reached.
```

Algorithm 1: IFPSO-KM algorithm.

$$\begin{aligned} Z_r^{(t+1)} =& V_r^{(t+1)} + \beta Z_r^t + \frac{1}{2}\beta(1-\beta)Z_r^{t-1} + \frac{1}{6}\beta(1-\beta)(2-\beta)Z_r^{t-2} \\ &+ \frac{1}{24}\beta(1-\beta)(2-\beta)(3-\beta)Z_r^{t-3}. \end{aligned} \tag{18}$$

Based on the standard particle swarm optimization algorithm, in the process of updating the particle speed, we consider increasing the connection between the global optimum and the local optimum and adding the $c_3 \times r_3 \times (P_g^t - P_r^t)$ variable on the basis of the previous algorithm (Algorithm 1).

4.3. Improved Clustering Model Algorithm. The steps involved in improved clustering model algorithm are given below.

5. Analysis of the Experiment

This part is mainly divided into three parts of simulation experiments, which are as follows: (1) The contrast experiment of the convergence characteristics of the IFPSO algorithm and the FOPSO algorithm is to test whether the efficiency of the algorithm is effectively improved. (2) Use the IFPSO algorithm and support vector machine to optimize parameters to construct two-category and multicategory prediction models. Analyze and evaluate the classification effect of the model. (3) Combine the IFPSO algorithm with K-means clustering to obtain the best combination of cluster centers. Then analyze and evaluate the clustering effect of the algorithm.

5.1. Convergence Analysis of IFPSO Algorithm

5.1.1. Introduction to Test Functions. To verify the performance of the IFPSO algorithm given in this paper, we compare the IFPSO algorithm and the FOPSO algorithm through several commonly used unimodal and multimodal test functions. Analyze whether the IFPSO algorithm effectively improves the convergence performance through the solution of the test functions. The characteristics of the four selected test functions are as follows:

Holder function is as follows:

$$f(\mathbf{x}) = -\left|\sin(x_1)\cos(x_2)\exp\left(\left|1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right|\right)\right|. \tag{19}$$

Inputting domain is $x_i \in [-10, 10]$, $i = 1, 2$.

Global minimum is $f(x^*) = -19.2085$. There are four minimum points, namely, $x^* = (8.05502, 9.66459), (8.05502, -9.66459), (-8.05502, 9.66459), (-8.05502, -9.66459)$. This function has many local minima and four global minimums. It is a multimodal function.

Generalized Rastrigin function is

$$f(\mathbf{x}) = 10\,d + \sum_{i=1}^{d}\left[x_i^2 - 10\cos(2\pi x_i)\right]. \tag{20}$$

Inputting domain is $x_i \in [-5.12, 5.12]$, $i = 1, \ldots, d$.

Global minimum is $f(x^*) = 0$, $x^* = (0, \ldots, 0)$.

The Rastrigin function has multiple local minima, but the positions of the minima are regularly distributed. It is a highly multimodal function, so it is not easy to search for the global best solution.

Schaffer function is

$$f(\mathbf{x}) = 0.5 - \frac{\sqrt{\sin\left(x_1^2 + x_2^2\right)} - 0.5}{1 + 0.001 \times \left(x_1^2 + x_2^2\right)^2}. \tag{21}$$

Inputting domain is $x_i \in [-4, 4]$, $i = 1, 2$.

Global maximum is $f(x^*) = 1$, $x^* = (0, \ldots, 0)$.

This function has a global maximum point, and there are infinite subglobal maximum points within a range of 3.14 from the global maximum point. In the simulation experiment, if the opposite value is taken, the extreme point should be -1.

Sphere function is

$$f(\mathbf{x}) = \sum_{i}^{d} x_i^2. \tag{22}$$

Inputting domain is $x_i \in [-5.12, 5.12], \quad i = 1, \ldots, d$.
Global minimum is $f(x^*) = 0$, $x^* = (0, \ldots, 0)$.

This function has a unique global minimum point. It is a continuous, convex, unimodal function.

5.1.2. Analysis of Convergence Results. In this paper, the IFPSO algorithm and FOPSO algorithm are simulated in Python, and the parameters of the comparison algorithm involved in this paper are given in Table 1.

In the experiment, the parameter settings of the IFPSO algorithm are maintained as the combination of the parameter settings of IFPSO-SVM. Set the parameters in the FOPSO algorithm to the parameter settings in literature [33], and change the inertia weight to linear inertia weight. The population size and number of iterations of the two algorithms are the same, and the optimization and convergence effects of the four test functions are shown in Figures 2–5.

In these four figures, the red line is the convergence curve of the IFPSO algorithm, and the gray line is the convergence curve of the FOPSO algorithm.

From the comparison chart of the optimization simulation curves of the four test functions above, it can be seen that the IFPSO algorithm in this paper has better optimization capabilities in the process of solving unimodal functions or multimodal functions. Moreover, its convergence speed and the ability to search for the best value are faster and more accurate than the algorithm before the improvement. Its performance is improved more obviously in complex multimodal functions. It can be seen from Figure 3 that, in the optimization process of the Generalized Rastrigin function, the FOPSO algorithm falls into the local optimum solution, and it fails to capture the global minimum. The position update of the IFPSO algorithm also has the characteristics of long-term memory of the fractional differential, which makes the algorithm's particle diversity higher, with better ability to jump out of the local optimal solution. Experiments show that the IFPSO algorithm not only improves the convergence speed but also enhances the ability of the particle global search.

5.2. Prediction Model Based on IFPSO-SVM

5.2.1. Experimental Data of the Prediction Model. This experiment uses the Statlog (heart) dataset. The dataset is obtained from the UCI machine learning database. This dataset contains a total of 303 cases, and each row records 13 characteristics and a label of the sample. Preprocess the data, and use 242 cases (about 80% of the total number of samples) as the training set and 61 cases (about 20% of the total number of samples) as the test set. The main attributes of the dataset are shown in Table 2.

The physical meaning, data unit, and magnitude of each attribute in the selected dataset are different. The original data is normalized and mapped in [0, 1], and the indicators are in the same order of magnitude, which is convenient for comprehensive comparison and evaluation.

5.2.2. Heart Disease Prediction Model Based on IFPSO-SVM. The modeling environment is Python, and the SVM model used is the SVC module of the sklearn tool. The full name is C-Support Vector Classification, which is a support vector machine based on LibSVM. The main purpose of this experiment is to test the effectiveness and progress of classification models and algorithms through simulation experiments of predictive models.

Use the IFPSO algorithm to build a machine learning binary classification model. Input the known heart disease data (training set) into the model, let the model learn, and obtain the ability to predict new unknown data (test set). The model is evaluated by comparing the prediction results of the test set with the real labels of the test set, and the classification performance of the model is evaluated. The kernel function selects the RBF function, and the most important parameters of the model are optimized through the IFPSO algorithm. The SVM model constructed by the IFPSO algorithm greatly improves the prediction accuracy of the model and reduces the time consumption of model calculations.

In the IFPSO algorithm, through experiments, it can be found that when the inertia weight value is less than 0.5, the search accuracy of the algorithm is not high, and it is easy to fall into the local optimal solution later. After many experiments, the accuracy of the algorithm is higher when $w_{\min} = 0.4$ and $w_{\max} = 0.8$. Compared with traditional algorithms, the calculation speed is faster, and the global search capability is also good. The introduction of the fractional-order algorithm also enriches the search behavior and avoids falling into the local optimum solutions. At the same time, the change of the current particle depends not only on the moment but also on the previous state, and the particle can obtain a better fitness value. The literature [20] shows that the accuracy of the algorithm is better when the value of the fractional operator α is between $[0.3, 0.8]$. After several experiments, this paper takes $\alpha = 0.4$ and $\beta = 0.7$, and the results show that the algorithm's performance at this time is relatively balanced and stable. In SVM, the kernel function uses the RBF function, and we obtain a set of optimal parameters (σ, C) through the IFPSO algorithm. After the IFPSO-SVM model is trained on the training set, the data in the test set can be predicted. Compare the test results with the real labels in the test set. Evaluate the model by drawing a confusion matrix, calculating evaluation indicators such as Precision, Recall, and F1-Score, and drawing ROC curves.

5.2.3. Multiclass Prediction Experiment Based on IFPSO-SVM. The dataset used in the experiment is UCI's red wine dataset. This dataset includes three types of wine, with 13 different composition characteristics and a total of 178 rows

Table 1: Algorithm parameter settings.

Algorithm	Parameter settings	Reference
PSO	$\omega \in [0.4, 0.9]; c_1 = c_2 = 2.0$	[32]
FOPSO	ω: 0.9; $a = 0.632; c_1 = c_2 = 1.5; r_{1\,d} = r_{2\,d} = [0, 1]$	[3]
IFPSO	$\omega \in [0.4, 0.8]; a = 0.4; \beta = 0.7$	

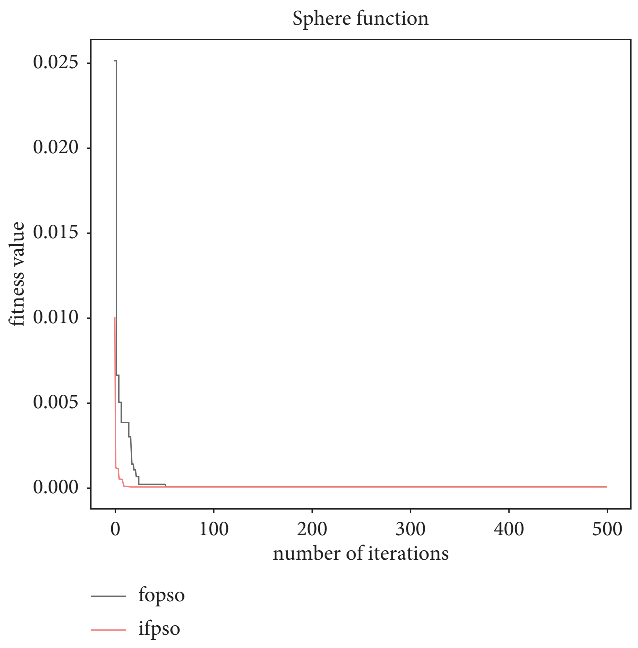

Figure 2: Simulation of sphere function.

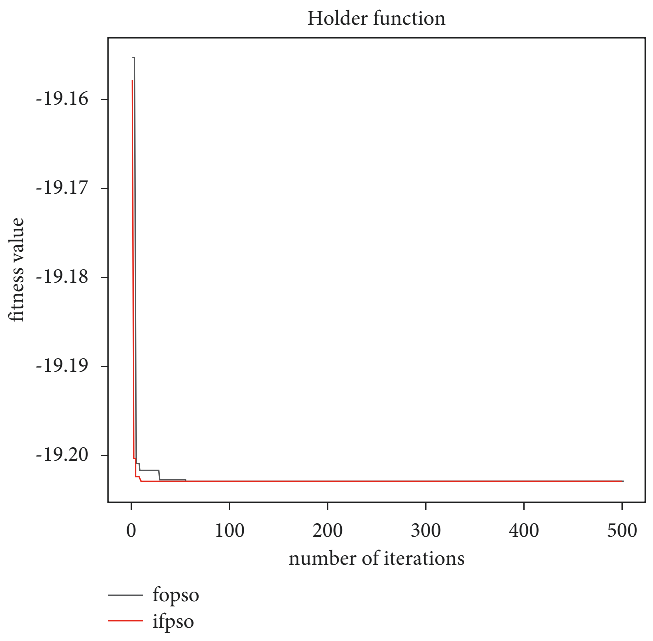

Figure 4: Simulation of Schaffer function.

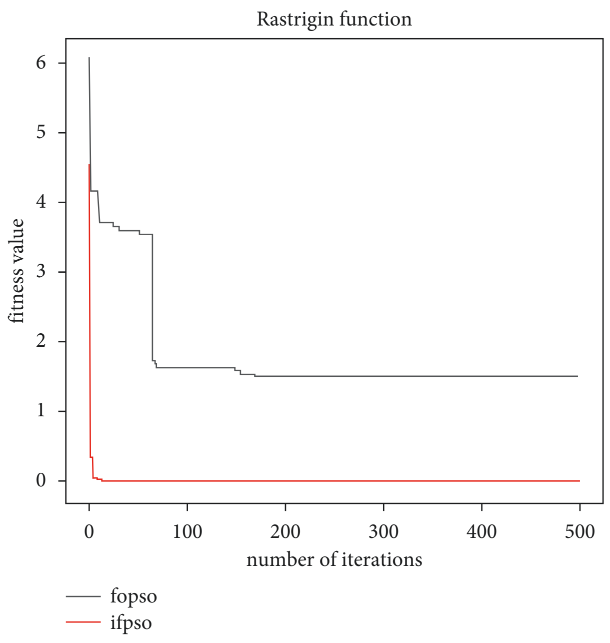

Figure 3: Simulation of Rastrigin function.

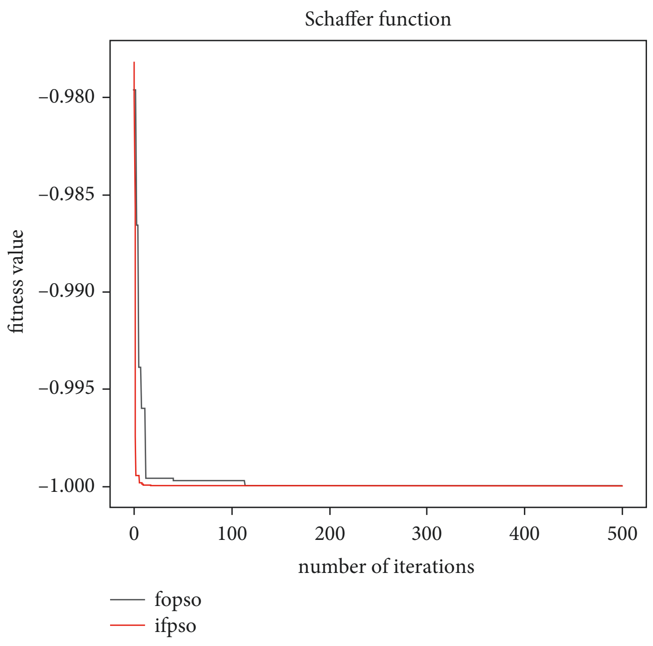

Figure 5: Simulation of Holder function.

of data. The three types of wine are marked as “1,” “2,” and “3.” There are 59 samples in the first category, 71 samples in the second category, and 48 samples in the third category. Preprocess the data, use 142 cases (about 80% of the total number of samples) as the training set and 36 cases (about 20% of the total number of samples) as the test set. The experimental parameters are the

Table 2: The attributes of the heart disease dataset.

Number	Attribute name	Types
1	Age	Continuous
2	Sex	Two categories
3	cp (types of chest pain)	Four categories
4	trestbps (blood pressure)	Continuous
5	chol (cholesterol)	Continuous
6	fbs (fasting blood glucose)	Two categories
7	restecg (ECG results)	Three categories
8	thslach (maximum heart rate)	Continuous
9	exang (whether angina during exercise)	Two categories
10	oldpeak (ST depression)	Continuous
11	slope (ST segment)	Ordered, three categories
12	ca (the number of blood vessels)	Continuous
13	thal (types of defects)	Three categories
14	Target	0, healthy 1, diseases

parameters of the two-category prediction model in the previous section.

5.3. Clustering Model Based on IFPSO-KM. In the simulation experiment, the classic dataset in the UCI database is used as the test dataset to verify the performance of the KM, PSO-K, ADPSO-AKM, and IFPSO-KM algorithms. The test dataset description is shown in Table 3. There are four attributes in the Iris dataset, namely, calyx length, calyx width, petal length, and petal width. There are three types of iris flowers, namely, *Iris setosa*, *Iris versicolor*, and *Iris virginica*. The Wine quality dataset contains two data subtypes. Sets are samples of red wine and white wine. Among them, there are 1599 samples in the red wine dataset and 4,858 samples in the white wine dataset. There are 11 physical and chemical properties of red wine (white wine) and the quality (score from 0 to 10) of red wine (white wine).

In the experiment, take the particle population $m = 80$ and the maximum number of iterations $T_{\max} = 15$. In the IFPSO algorithm, the inertia weight is also linearly decreased. Through many experiments, set $c_1 = c_2 = 1.49$, $\omega_{\max} = 0.8$, $\omega_{\min} = 0.3$, $\alpha = 0.6$, and $\beta = 0.4$. Through the IFPSO algorithm, we obtain a set of particle combinations of cluster centers and use the KM algorithm to cluster according to this particle combination. Use common clustering evaluation indicators (Rand index, homogeneity score) for model evaluation. The clustered data can be compared with the real data in the way of supervised learning to get the accuracy of the model.

5.4. Analysis of Results

5.4.1. Confusion Matrix. A confusion matrix can effectively measure the accuracy of a classifier's classification. It is a situation analysis table for summarizing and predicting the results of classification models in data science, data analysis, and machine learning. In the form of a matrix, the records in the dataset are summarized according to the two criteria of the real category and the classification judgment made by the classification model. Take the binary classification problem as an example. There are two types of records in the dataset: positive and negative. The classification model will make positive judgments (judgment records belong to the positive category) or negative judgments (judgment records belong to the negative category). The row represents the reality, and the column represents the forecast.

The evaluation indicators extended by the confusion matrix as follows:

Accuracy: the percentage of correctly classified samples in the total sample.

Precision: the probability of a positive sample among the samples predicted to be positive.

Recall: the probability that the model predicts a positive sample in the actual positive sample.

F1-score: it can be regarded as the harmonic mean of model Accuracy and Recall.

5.4.2. Evaluation Index of Clustering Model. SSE (sum of the squared errors) is a measure of cluster looseness. As an objective function, it is actually a strict Coordinate Decedent process. SSE cannot guarantee finding the global optimal solution but can only guarantee the local optimal solution. In other words, it may cause a variety of k cluster divisions.

Silhouette Coefficient combines the degree of cohesion and the degree of separation to evaluate the effect of clustering. The value range of the average profile coefficient is $[-1, 1]$. The larger the coefficient, the better the clustering effect. After each clustering, each sample will get a contour coefficient. When it is 1, it means that the point is far away from the surrounding clusters, which is a good classification. When the value is 0, it means that the point is on the boundary of the two clusters. When the value is negative, it means that the point may be misclassified.

Adjusted Rand index is proposed to achieve "in the case of clustering results randomly generated, the index should be close to zero," which has a higher degree of discrimination; the calculation formula is

Table 3: Properties of the test dataset.

Test dataset	Number of data samples	Number of clusters	Data dimension
Iris	150	3	4
Wine quality-white	4858		12
Wine quality-red	1599		12

$$\mathrm{ARI} = \frac{RI - E[RI]}{\max(RI) - E[RI]}. \tag{23}$$

The value range of ARI is [−1, 1]. The larger the value is, the more consistent the clustering result is with the real situation. In a broad sense, ARI measures the degree of agreement between the two data distributions.

5.4.3. Evaluation of Predictive Models. Figure 6 is an illustration of the confusion matrix drawn by the prediction results after the dataset is trained with IFPSO-SVM. Each row represents the real situation, each column represents the prediction situation, and the main diagonal represents the correct sample size of the prediction.

It can be seen from the figure that the total number of data samples in the test set is 61, of which 29 are health data samples and 32 are disease data samples. 26 of the health data samples were correctly predicted, and 29 of the disease data samples were correctly predicted.

Table 4 shows the values of some evaluation indicators calculated from the confusion matrix obtained by the prediction model.

To compare the performance of the heart disease prediction model based on the IFPSO-SVM proposed in this paper, several other algorithms [34, 35] are used to predict the same dataset, as well as the Precision, Recall, and F1-Score. The experimental results are shown in Table 5.

Through the comparison experiment, we can see that, for the basic PSO-SVM classification model, the prediction accuracy is about 80%. The accuracy of the model built by the PSO algorithm that only updates the velocity in fractional order is about 83%. The prediction accuracy of the IFPSO-SVM model in this article has reached 90%, which greatly improves the effectiveness and accuracy of the model. All indicators are better than those in other algorithms.

Finally, draw the ROC (Receiver Operating Characteristic) curve of the method in this article. Combining the two variables (1 − specificity and sensitive) with a graphical method can more clearly and accurately integrate the accuracy of the reaction model. The horizontal axis is the false positive rate, and its value is proportional to the false positive rate. The vertical axis is the true positive rate, and its value is proportional to the accuracy rate. Therefore, the closer to the upper left corner of the ROC space, the better the judgment effect, as shown in Figure 7.

AUC is the area under the ROC curve. In the experiment, the larger the AUC value, the better the model effect. Usually, (1) AUC ≈ 1.0 means the model is the most ideal, (2) $0.7 \leq \mathrm{AUC} \leq 0.9$ means the accuracy of the model is high, and (3) AUC = 0.5 means the model is meaningless. In general, an AUC of 0.9 or more can be regarded as a highly accurate judgment experiment. The prediction at this time is of practical significance. In the above model, AUC = 0.9385 (the parameter combination is as follows: gamma = 0.055; $c = 0.7915$). It shows that the accuracy of the model has reached the standard of practical significance, and the introduction of the IFPSO algorithm greatly improves the accuracy of the model.

In the multiclassification application of IFPSO-SVM, the model also has a better improvement effect. Table 6 shows

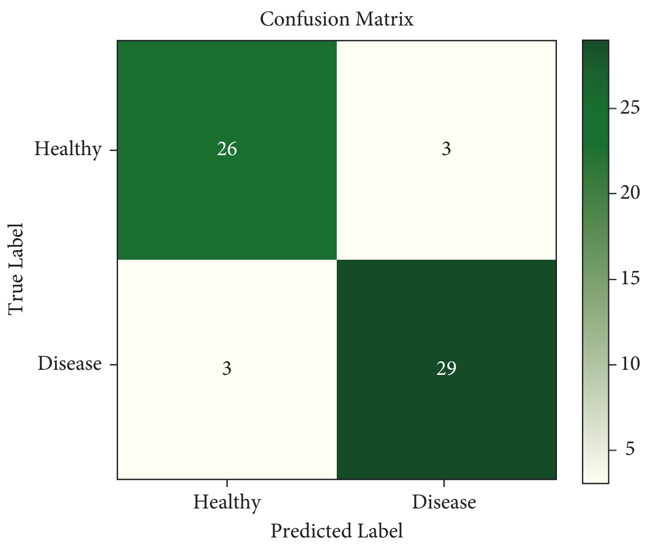

Figure 6: Confusion matrix.

Table 4: Evaluation index value of IFPSO-SVM algorithm (heart disease dataset).

	Precision	Recall	F1-Score	Support
Healthy	0.90	0.90	0.90	29
Disease	0.91	0.91	0.91	32
Accuracy			0.90	61
Macro avg.	0.90	0.90	0.90	61
Weighted avg.	0.90	0.90	0.90	61

Table 5: Comparison of prediction results of different algorithms.

Algorithms	Precision	Recall	F1-Score
PSO-SVM	0.809	0.759	0.783
BP	0.831	0.783	0.806
XGBoost	0.865	0.804	0.833
FOPSO-SVM	0.865	0.845	0.855
IFPSO-SVM	0.910	0.910	0.900

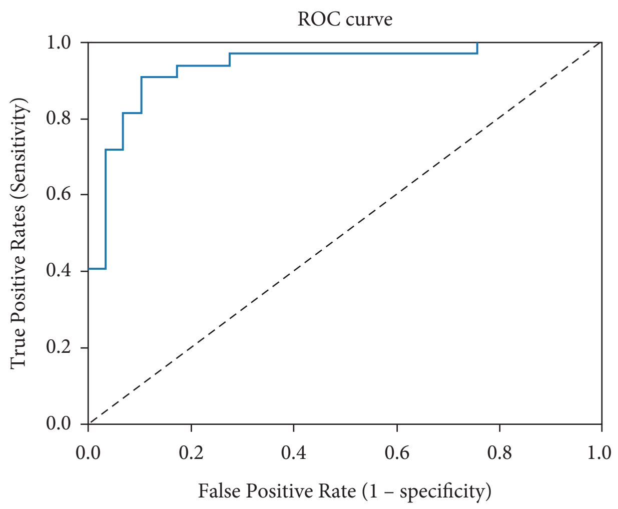

Figure 7: ROC curve graph.

the algorithm evaluation of the model to classify the Wine dataset.

It can be seen that we selected 20% of the dataset as the test set, and the Precision, Recall, and F1-Score obtained from the simulation experiment all reached 1 (the parameter combination is as follows: gamma = 0.158; $c = 6.5811$). Explain that the predictions for all test data are accurate. At the same time, it shows that the IFPSO algorithm effectively improves the prediction accuracy and accuracy of SVM, not only for accidental datasets. The algorithm has a good classification effect on two-category and multicategory problems.

In summary, the IFPSO algorithm in this paper has better performance in the parameter optimization problem of SVM. The time complexity of the algorithm is significantly reduced, and the ability to find the global optimal solution is improved. The IFPSO-SVM prediction model constructed in this paper has achieved a higher diagnostic success rate than other models in the diagnosis of heart disease, which makes the research of the model have certain practical reference significance. The accuracy of the multicategory model has also reached 100%. The effect of the algorithm has improved significantly.

Table 6: Evaluation index value of IFPSO-SVM algorithm (Wine dataset).

	Precision	Recall	F1-Score	Support
Type 1	1.00	1.00	1.00	14
Type 2	1.00	1.00	1.00	14
Type 3	1.00	1.00	1.00	8
Accuracy			1.00	36
Macro avg.	1.00	1.00	1.00	36
Weighted avg.	1.00	1.00	1.00	36

5.4.4. Evaluation of the Clustering Model. According to the change curve of SSE and profile coefficient, the number of clusters is dynamically obtained. Use the standard dataset Iris as the test data. Set the range of the number of clusters k to [2, 10], and draw the SSE and Silhouette Coefficient corresponding to different numbers of clusters, to select the value of k more intuitively, as shown in Figure 8.

Combining the graph, it can be seen that when $k = 3$, the SSE value decreases significantly, and the Silhouette Coefficient value is also higher. The actual number of classifications in the Iris dataset is also 3, which is consistent with the calculated k value.

The IFPSO-KM algorithm is verified through experiments on UCI's Iris and Wine dataset. The clustering effect of the Iris dataset can be represented by three-dimensional images. Figure 9 shows the clustering effect diagram of the IFPSO-KM algorithm, and Figure 10 shows a diagram of scattering clustering according to the true value. It can be seen that the effect of the improved algorithm clustering is better. Its clusters are relatively close and the distance between clusters is large, which meets the clustering requirements.

When using IFPSO-KM to cluster two datasets, the parameters are set as follows: take $c_1 = c_2 = 1.49$, $\omega_{\max} = 0.8$, $\omega_{\min} = 0.3$, $\alpha = 0.6$, and $\beta = 0.4$. Table 7 shows the number of error points for training on the Iris and Wine datasets by different clustering methods. Since the Wine dataset has more attribute values, only the corresponding initial center points of Iris are given in the table. The optimal initial center points obtained by the IFPSO- KM algorithm are the 49th, 98th, and 129th data points in the sample set, respectively. After experiments, we can see that the KM algorithm is very sensitive to the initial center point. Although the number of misclassifications in some experiments is relatively small, the improved algorithm combined with the PSO algorithm is better than the KM algorithm in terms of the degree of dependence on the initial center point and the accuracy of the classification. Effectively reduce the randomness of clustering.

It is worth pointing out that the standard KM algorithm is not very effective for clustering datasets with more data attributes and a relatively large amount of data, and the classification accuracy can only reach about 80%. The

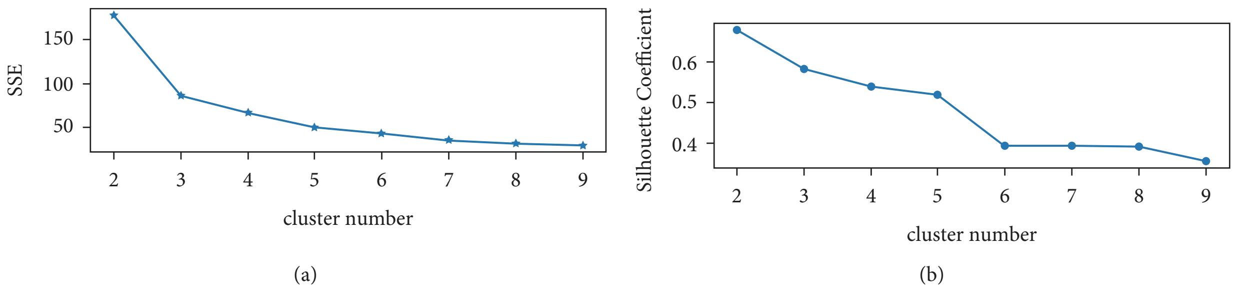

Figure 8: Change curve of SSE and Silhouette Coefficient. (a) K-means SSE. (b) K-means Silhouette Coefficient.

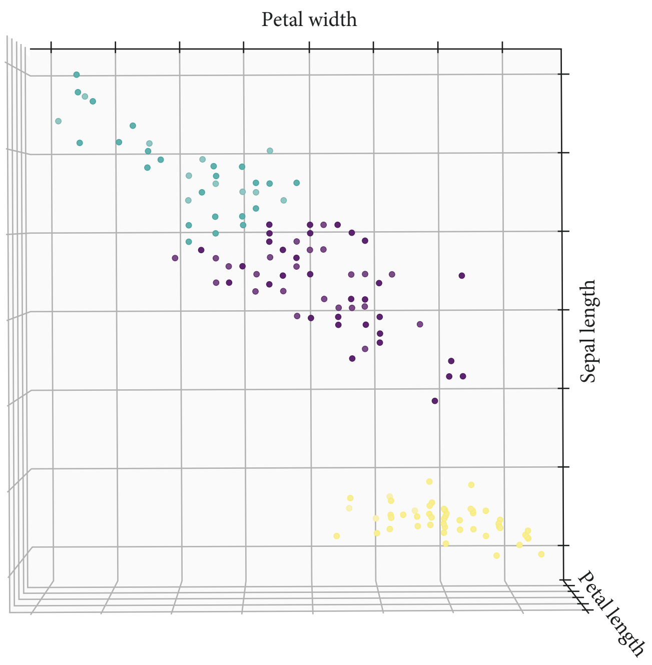

Figure 9: Cluster graph of IFPSO-KM (Iris).

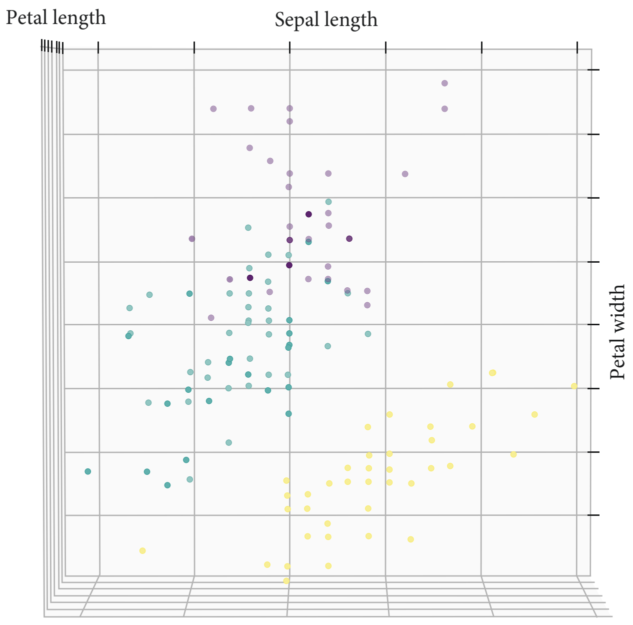

Figure 10: Iris' real cluster graph.

Table 7: Clustering result.

Algorithm	Initial cluster center of Iris dataset			Misclassifications	
				Iris dataset	Wine dataset
K-means	11	55	122	12	650
K-means++	35	89	122	11	602
PSOK-means	41	72	109	6	286
IFPSO-KM	49	98	129	6	48

classification accuracy of the IFPSO algorithm in this article combined with KM reached 98.183%, and the Rand index also reached 0.924, with a homogenization score of 0.905. It is a clustering model with a better clustering effect.

6. Conclusion

For the problems of FOPSO algorithm and FODPSO algorithm, which only improve the velocity update formula with fractional factor, we consider adding new variables. The fractional derivation is performed on both velocity and position formulas. It can reduce the dependence on moperator α and increase the randomness of the particles. The probability of the population falling into the local optimum is reduced. Experiments have also proved that the improved algorithm has better convergence performance and better ability to find the global optimal solution.

Aiming at the problem of low prediction accuracy of SVM, this paper establishes the prediction model of IFPSO-SVM. The IFPSO algorithm is used to optimize the error penalty factor of the SVM and the parameter of the kernel function. Combining the heart dataset for simulation experiments and comparing the prediction results with the previous model, the results show that the IFPSO-SVM model has the characteristics of higher prediction accuracy and faster convergence speed. Moreover, it has a good classification effect on two-category and multi-category datasets. Due to the shortcomings of the KM algorithm, such as excessive reliance on the selection of the initial center, difficulty to obtain the global optimal solution, and empty class, this paper uses the IFPSO algorithm to select the clustering center of the dataset to optimize the clustering effect. The IFPSO-KM algorithm has a fast convergence speed and a stable clustering result. It is worth pointing out that the combination of fractional particle swarm algorithm and mutation operation can prevent empty classes and improve clustering accuracy at the same time.

In the future research, adopting an adaptive algorithm for the fractional order to better enhance the diversity and randomness of particles shall be considered. For the dataset, it is possible to consider setting different size impact factors on the input feature vector according to the degree of importance to facilitate better training of the model.

Data Availability

All data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation under Grants 61862062 and 61104035.

References

[1] R. A. Krohling, "Gaussian particle swarm with jumps," in *Proceedings of the 2005 IEEE Congress On Evolutionary Computation*, vol. 1-3, pp. 1226–1231, Edinburgh, UK, September 2005.

[2] J. Tillett, T. M. Rao, F. Sahin, and R. Rao, "Darwinian particle swarm optimization," in *Proceedings of the Indian International Conference on Artificial Intelligence*, Pune, India, December 2005.

[3] E. J. Solteiro Pires, J. A. Tenreiro Machado, P. B. de Moura Oliveira, J. Boaventura Cunha, and L. Mendes, "Particle swarm optimization with fractional-order velocity," *Nonlinear Dynamics*, vol. 61, no. 1-2, pp. 295–301, 2010.

[4] M. S. Couceiro, R. P. Rocha, N. M. Fonseca Ferreira, and J. A. Tenreiro Machado, "Introducing the fractional-order darwinian PSO," *Signal, Image and Video Processing*, vol. 6, no. 3, pp. 343–350, 2012.

[5] Y. Y. Wang, W.-X. Peng, C.-H. Qiu, J. Jiang, and S.-R. Xia, "Fractional-order darwinian PSO-based feature selection for media-adventitia border detection in intravascular ultrasound images," *Ultrasonics*, vol. 92, 2018.

[6] A. U. Rehman, A. Islam, and S. B. Belhaouari, "Multi-cluster jumping particle swarm optimization for fast convergence," *IEEE Access*, vol. 8, pp. 189382–189394, 2020.

[7] M. Sugisaka and X. Fan, "An effective search method for NN-based face detection using PSO," in *Proceedings of the Sice Conference*, Sapporo, Japan, August 2004.

[8] H. Feng, W. Ma, C. Yin, and D. Cao, "Trajectory control of electro-hydraulic position servo system using improved PSO-PID controller," *Automation in Construction*, vol. 127, Article ID 103722, 2021.

[9] J. Wang, Y. Di, and X. Rui, "Research and application of machine learning method based on swarm intelligence optimization," *Journal of Computational Methods in Science and Engineering*, vol. 19, no. 2, pp. 179–187, 2019.

[10] W. Xiaokai, S. Guan, L. Hua, B. Wang, and X. He, "Classification of spot-welded joint strength using ultrasonic signal time-frequency features and PSO-SVM method," *Ultrasonics*, vol. 91, 2018.

[11] Z. Wei, D. Ma, J.-j. Wei, and H.-f. Liang, "A parameter selection strategy for particle swarm optimization based on

particle positions," *Expert Systems with Applications*, vol. 41, 2014.

[12] I. Y. Chung, W. Liu, D. A. Cartes, and K. Schodr, "Control parameter optimization for a microgrid system using particle swarm optimization," in *Proceedings of the IEEE International Conference on Sustainable Energy Technologies*, Singapore, November 2008.

[13] Q. Wu, S. Wu, and J. Liu, "Mechanical fault diagnoses approach based on Fv-SVM," *Systems Engineering-Theory & Practice*, vol. 30, no. 7, pp. 1266–1271, 2010.

[14] S. C. Yao and H. X. Pan, "Fractional order PID controller for synchronous machine excitation using particle swarm optimization," *Proceedings of the Csee*, vol. 30, 2010.

[15] W. Chen and F. Han, *An Improved Multi-Objective Particle Swarm Optimization with Adaptive Penalty Value for Feature Selection*, Springer Singapore, Singapore, 2020.

[16] X. Zhu and D. Wang, *A Hybrid Ant Colony Optimization Algorithm for the Fleet Size and Mix Vehicle Routing Problem with Time Windows*, Springer Singapore, Singapore, 2020.

[17] S. Qiu, B. Hu, Y. Quan, X. Jian, and H. Ouyang, *Ant Colony Algorithm Based on Upper Bound of Nodes for Robot Path Planning Problems*, Springer Singapore, Singapore, 2020.

[18] A. Kaur, S. Kumar Pal, and A. P. Singh, "Hybridization of chaos and flower pollination algorithm over K-means for data clustering," *Applied Soft Computing*, vol. 97, 2019.

[19] N. Kamel, I. Ouchen, and K. Baali, "A sampling-PSO-K-means algorithm for document clustering," *Advances in Intelligent Systems and Computing*, vol. 238, pp. 45–54, 2014.

[20] J. Raitoharju, S. Kiranyaz, and M. Gabbouj, "Evolutionary feature synthesis by multi-dimensional particle swarm optimization," in *Proceedings of the 2014 5th European Workshop on Visual Information Processing (EUVIP)*, December 2014.

[21] F. Hamdaoui, A. Sakly, and A. Mtibaa, "An efficient multi level thresholding method for image segmentation based on the hybridization of modified PSO and otsu's method," *Computational Intelligence Applications in Modeling and Control*, Springer, Berlin, 2015.

[22] Y. Xu and F. Wang, "Gesture recognition method based on sEMG by APSO/CS-SVM," *Journal of Electronic Measurement and Instrument*, vol. 34, no. 7, pp. 1–7, 2020.

[23] Y. Zhang, D.-w. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 64–75, 2017.

[24] H. Yin and N. K. Jha, "A health decision support system for disease diagnosis based on wearable medical sensors and machine learning ensembles," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 3, no. 4, pp. 228–241, 2017.

[25] Y. Sun, D. He, and J. Li, "Research on the fatigue life prediction for a new modified asphalt mixture of a support vector machine based on particle swarm optimization," *Applied Sciences*, vol. 11, no. 24, p. 11867, 2021.

[26] J. Kennedy, "Parameter selection in particle swarm optimization," in *Proceedings of the.IEEE Int.conf.neural Network*, Australia, May 1998.

[27] V. E. Tarasov, "On history of mathematical economics: Application of Fractional Calculus," *Mathematical Economics: Application of Fractional Calculus*, vol. 8, no. 4, 2019.

[28] D. P. Kanungo, B. Naik, J. Nayak, S. Baboo, and H. S. Behera, *An Improved PSO Based Back Propagation Learning-MLP (IPSO-BP-MLP) for Classification*, Springer, New Delhi, 2015.

[29] J. B. Park, Y. W. Jeong, J. R. Shin, and K. Y. Lee, "An improved particle swarm optimization for nonconvex economic dispatch problems," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 156–166, 2010.

[30] E. Addeh, V. Ranaee, and R. Ghaderi, "Application of the PSO-SVM model for recognition of control chart patterns," in *Proceedings of the International Conference on Control*, Yokohama, Japan, July 2010.

[31] J. Shu, S. Liu, L. Liu, L. Zhan, and G. Hu, "Research on link quality estimation mechanism for wireless sensor networks based on support vector machine," *Chinese Journal of Electronics*, vol. 26, no. 2, pp. 377–384, 2017.

[32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the Icnn95-international Conference on Neural Networks*, Perth, WA, Australia, December 1995.

[33] C. H. Yang, G. U. Li-Shan, and W. H. Gui, "Particle swarm optimization algorithm with adaptive mutation," *Computer Engineering*, vol. 34, no. 16, pp. 188–190, 2010.

[34] Z. Jiang, J. Zhou, Z. Yuanpeng, and W. Shitonga, "A novel multi-view SVM based on consistent hidden density distributions between views for face recognition," *Journal of Intelligent and Fuzzy Systems*, vol. 36, no. 6, pp. 5245–5259, 2019.

[35] Y. Liu and M. Qiao, "Heart disease prediction based on clustering and XGboost," *Computer Systems & Applications*, vol. 28, 2019.