





Article

Intelligent Multi-Strategy Hybrid Fuzzy K-Nearest Neighbor Using Improved Hybrid Sine Cosine Algorithm

Chengfeng Zheng¹, Mohd Shareduwan Mohd Kasihmuddin^{1,*}, Mohd. Asyraf Mansor², Ju Chen¹
and Yueling Guo¹

¹ School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia

² School of Distance Education, Universiti Sains Malaysia, Penang 11800, Malaysia

* Correspondence: shareduwan@usm.my; Tel.: +60-4-6534769

Abstract: The sine and cosine algorithm is a new simple and effective population optimization method proposed in recent years that has been studied in many works of literature. Based on the basic principle of the sine and cosine algorithm, this paper fully studies the main parameters affecting the performance of the sine and cosine algorithm, integrates the reverse learning algorithm, adds an elite opposition solution and forms the hybrid sine and cosine algorithm (hybrid SCA). Combined with the fuzzy k-nearest neighbor method and the hybrid SCA, this paper numerically simulates two-class datasets and multi-class datasets, obtains a large number of numerical results and analyzes the results. The hybrid SCA FKNN proposed in this paper has achieved good accuracy in classification and prediction results under 10 different types of data sets. Compared with SCA FKNN, LSCA FKNN, BA FKNN, PSO FKNN and SSA FKNN, the prediction accuracy is significantly improved. In the Wilcoxon signed rank test with SCA FKNN and LSCA FKNN, the zero hypothesis (significance level 0.05) is rejected and the two classifiers have a significantly different accuracy.

Keywords: meta learning; data classification; hybrid sine and cosine algorithm; Wilcoxon signed rank test; multiple application scenario datasets

MSC: 68T07; 68T27; 68T20



Citation: Zheng, C.; Kasihmuddin, M.S.M.; Mansor, M.A.; Chen, J.; Guo, Y. Intelligent Multi-Strategy Hybrid Fuzzy K-Nearest Neighbor Using Improved Hybrid Sine Cosine Algorithm. *Mathematics* **2022**, *10*, 3368. <https://doi.org/10.3390/math10183368>

Academic Editors: Niansheng Tang and Shen-Ming Lee

Received: 8 August 2022

Accepted: 5 September 2022

Published: 16 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The swarm intelligence algorithm (SI) has gained attention from many researchers in various field of sciences. SI is currently being used to provide solutions to various optimization problems. Several applications of swarm intelligence include material technology [1–3], biological system modeling [3], train assembly, high-performance graphics card [4], path planning [5] and robot control [6,7]. SI is based on the collective behavior of the elements that self-organize in order to get exposed with the solution of the optimization problem. Examples of popular SI algorithms include particle swarm optimization (PSO) [8], artificial bee colony (ABC) [9], gravitational search algorithm (GSA) [10] and whale optimization algorithm (WOA) [11]. One of the main challenges in SI is the lack of profound theoretical analysis, which requires a solid mathematical foundation that includes a proper analysis that assesses the robustness, computational complexity and parameter setting. All of these analyses are required to ensure that SI can avoid converging to a local minimum solution. Note that the local minimum solution will affect the optimality of the solution to the respective optimization problem.

The sine cosine algorithm (SCA) [12] is a new SI algorithm proposed by Mirjalili in 2016. SCA was inspired by a mathematical model of the sine cosine function used to make the oscillation of the solution converge towards the optimal solution. The random and adaptive parameters in the algorithm have the ability to balance both exploration and exploitation during solution searching. Several advantages of SCA include very few

parameters, easy implementation, a simple structure, a fast convergence speed, strong parallelism and universality and a better performance in practical applications. Therefore, it has attracted extensive attention from scholars in recent years.

In [13], a SCA with nonlinear decreasing conversion parameters was proposed. The change in parameter r_1 is controlled by a parabolic function and exponential function, respectively. The experimental results show that the adjustment of parameter r_1 by the exponential function can better balance the global and local exploration of the solution. The study in [14] proposed a method of combining quantum computing with SCA by using quantum bits, rotating gates and a non-gate, where each gate has their own specialization in exploring the solution. The proposed SCA was reported to be very effective and accurate. The study in [15] introduces the method of reverse learning to generate reverse solutions for the current individual, which expands the exploration of solution space. The study in [16] proposed a hybrid gray wolf with SCA that uses the sine cosine update formula to improve the moving direction and speed of the head wolf. By combining the benefit of SCA, the hybrid algorithm improves drastically in terms of exploration and exploitation. Another interesting work by [17] was carried out, where SCA was combined with a differential evolution algorithm (DEA). In this context, SCA was reported to help DEA jump out from the local optimal solution region. However, the application of the SCA in the above literature failed to classify a more general dataset. The test of the robust metaheuristics depends on the ability of the algorithm to classify non-bias datasets.

In 2017, Rizk M and Rizk-Allah [18] proposed a sine cosine algorithm (MOSCA) based on a multi-orthogonal search strategy to solve engineering design problems. The proposed MOSCA improves the defects of unbalanced exploration and premature convergence of the conventional SCA. MOSCA utilizes SCA during the exploration phase and uses a multi-orthogonal search strategy to find the optimal solution in the search space. MOSCA was reported to obtain a better speed of convergence with a higher solution accuracy. In another development, Elaziz et al. [15] proposed SCA based on reverse learning (OBSCA). A reverse learning strategy is an important method used to enhance the performance of the stochastic optimization algorithm. By selecting the value of the objective function according to greedy selection at the current solution and reverse solution, OBSCA enhances the diversity of the population and improves the ability of the algorithm to approach the global optimal solution. This experiment highlights the robustness of OBSCA in terms of convergence.

In 2017, Songjin and Wen [19] proposed an improved SCA (ISCA) for solving high-dimensional optimization problems. Inspired by PSO, the ISCA algorithm introduces inertia weight to improve the convergence accuracy and increase the convergence speed of the SCA. At the same time, it adopts a reverse learning strategy to generate initial individuals to improve the diversity and reconciliation quality of the population. The experimental results showed that, compared with the basic SCA, ISCA has a better optimization performance in a high-dimensional test function. In 2018, Nenavath and Jatoth [20] proposed a hybrid SCA-DE algorithm based on differential evolution to solve optimization problems and target tracking problems. The experimental results show that the hybrid SCA-DE algorithm has a higher convergence accuracy and faster convergence speed than basic SCA. In 2021, Wu et al. [21] proposed a LSCA method and FKNN method to solve biomedical problems. Compared with other methods, the proposed LSCA obtained acceptable results but the accuracy of this method still requires improvement.

In this paper, we capitalize on the mathematical properties of the SCA to balance the global and local exploration of the algorithm during the searching process. This can be achieved by adaptively changing the amplitude of the sine function and cosine function until the SCA converges towards the global optimal solution. In addition, reverse learning will be used to provide a jump mechanism to the SCA so that it can avoid a potential unwanted local solution. Both methods will be integrated into a fuzzy k-nearest neighbor (FKNN) that has the capability to classify real life datasets. Thus, the contributions of this paper are as follows:

1. In this paper, reverse learning will be implemented into an SCA model to form a hybrid SCA. In this context, the adaptive weight coupled with the reverse learning alter the position of the solution towards the global solution.
2. The proposed hybrid SCA will be implemented into a fuzzy k-nearest neighbor (SCA-FKNN). In this context, the proposed SCA-FKNN has the ability to avoid local convergence by jumping out of the current non-optimal solution.
3. The performance of the proposed SCA-FKNN will be tested using various real life datasets. SCA-FKNN will be evaluated according to the various performance metrics, such as accuracy, precision, sensitivity, specificity, Mathews correlation coefficient and Wilcoxon signed rank test. In addition, the proposed SCA-FKNN will be compared with the existing conventional state-of-the-art classifier.

The rest of this paper is organized as follows. Section 2 introduces in detail the content of the SCA model and FKNN classifier. Section 3 presents in detail the process of forming the hybrid SCA FKNN model based on the SCA model and FKNN classifier by adding a parameter adjustment and reverse learning mechanism. Section 4 introduces and analyzes 10 different data sets and evaluation indicators. Section 5 shows the prediction and classification results of 10 types of data sets under five models with extensive comparison analysis. Section 6 describes conclusions and further research.

2. Background

2.1. Sine Cosine Algorithm

The sine cosine algorithm is an algorithm based on the mathematical characteristics of sin and cos. It updates individuals through the changes in sine and cosine functions. In SCA, it is assumed that, in j -dimensional space, the population size is n , and that, in each iteration, the location update mode of the i -th individual is

$$X_i^j(t+1) = \begin{cases} X_i^j(t) + r_1 \times \sin(r_2) \times \left| r_3 X_{best}^j - X_i^j(t) \right| & r_4 < 0.5 \\ X_i^j(t) + r_1 \times \cos(r_2) \times \left| r_3 X_{best}^j - X_i^j(t) \right| & r_4 \geq 0.5 \end{cases} \quad (1)$$

where X_i^j is the position of the i -th individual in the j dimension of the t iteration; $X_{best,j}$ is the optimal position in the j dimension of the position X_i of the i -th individual; r_2 , r_3 and r_4 are random numbers subject to uniform distribution, $r_2 \in [0, 2\pi]$, $r_3 \in [0, 2]$ and $r_4 \in [0, 1]$; r_1 is the control parameter.

$$r_1 = a * \left(1 - \frac{t}{MaxFEs}\right) \quad (2)$$

where $MaxFEs$ represents the maximum number of iterations and a is a constant number and is equal to 2.

The fluctuation amplitude of $r_1 \times \sin(r_2)$ and $r_1 \times \cos(r_2)$ (sine and cosine parameters) gradually attenuates with the increase in iteration times. Its values are in the range of $(1, 2]$ and $[-2, -1)$. The algorithm performs a global search in the solution space, and the algorithm performs a local development in the range of $[-1, 1]$. The SCA algorithm flow is shown in Figure 1.

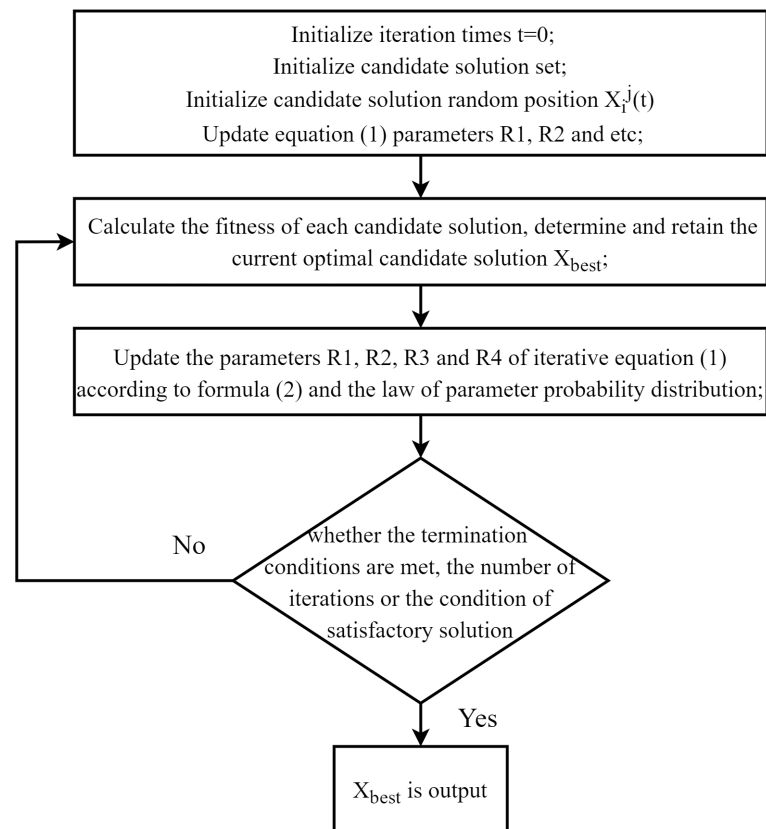


Figure 1. The flow chart of SCA.

2.2. Fuzzy K-Nearest Neighbors (FKNN)

As one of the simplest classifiers, KNN mainly infers the class of the sample according to the classes of the K training samples closest to the sample to be classified. The default of this method is that each of the k samples have the same weight, which is not the case. The KNN algorithm (nearest neighbor method) was first proposed by Cover and Hart in 1967. Many researchers have conducted in-depth theoretical research and development due to the low error rate of the nearest neighbor method, which makes it one of the important methods of pattern classification.

The FKNN algorithm (fuzzy k-nearest neighbor algorithm) was proposed by Keller et al. in 1985 [22]. He assigned different weighting coefficients to k -nearest neighbors, and then used the fuzzy decision-making method to calculate the class label with the largest coefficient as the category of test data. Because the weight coefficient based on distance is used, the recognition effect is improved. Nevertheless, the selection of fuzzy k-nearest neighbor parameter K has a great impact on the recognition effect. Choosing appropriate parameters plays an important role in improving the accuracy of the classification.

A fuzzy KNN algorithm is proposed based on the KNN algorithm. This method has the advantages of a high calculation accuracy and no data input assumption. It is a relatively mature classifier. For data sets, the membership of each member data to each class is calculated by Equation (3).

$$U_{i,k} = \begin{cases} 0.51 + \left(\frac{n_k}{K}\right) \cdot 0.49, & k = Y_k \\ \left(\frac{n_k}{K}\right) \cdot 0.49, & k \neq Y_k \end{cases} \quad (3)$$

where $i = 1, 2, 3 \dots N$ represents the i -th training sample and N represents the number of all training samples. $k = 1, 2, 3 \dots M$, where k represents the k -th class, and M denotes the number of classes. $U_{i,k}$ represents the member level of the i -th sample to the k -th class. K represents the present number of nearest neighbors, Y_k represents the class of the

i -th training sample and n_k represents the number of the i -th training sample's neighbors belonging to the k -class among the nearest K neighbors. Note that membership should meet the following:

$$U_k(x) = \frac{\sum_{j=1}^K U_{I_j,k} (x - x_{I_j})^{\frac{2}{m-1}}}{\sum_{j=1}^K (x - x_{I_j})^{\frac{2}{m-1}}} \quad (4)$$

where x stands for the test sample, $U_k(x)$ represents the test sample weight to the k -class, $j = 1, 2, \dots, K$ represents the test sample's j -th nearest neighbor, I_j represents the i index corresponding to the j -th nearest neighbor in the training samples, $U_{I_j,k}$ is the membership degree, which is calculated by Equation (3), and $x - x_{I_j}$ represents the distance measurement. m stands for fuzzy strength, which is used to control the weight of each neighbor in the membership calculation, and its range is $[1, \infty]$.

$$C(x) = \arg \max_k U_k(x) \quad (5)$$

The calculation steps of FKNN are as follows in Figure 2.

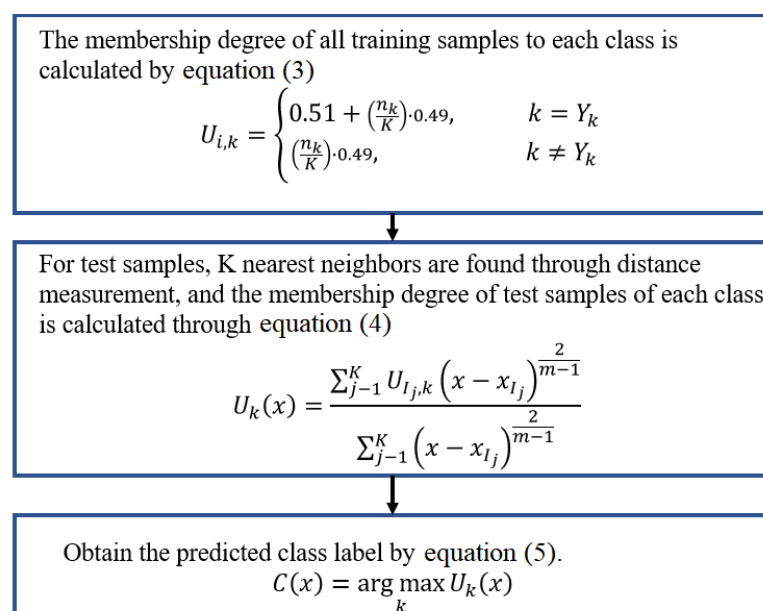


Figure 2. The flow chart of FKNN.

The above FKNN solved the problem of multivariate classification and distance weight, and the SCA will deal with the problem of a low search efficiency of the optimal solution after the distance weight.

3. The Proposed Method

At the end of the iteration, SCA will conduct a small neighborhood search near the current global optimal location and constantly try to update the optimal solution. If the search process is far from the theoretical optimal solution, it is difficult for the algorithm to converge to the global optimal solution in a short time. Therefore, the current research papers are roughly divided into two ways to improve the convergence speed and accuracy of SCA. One way is to improve the convergence speed of SCA by changing Equation (1). The other is to improve the accuracy of SCA by adding reverse learning.

In the parameter adjustment mechanism, reference [23] has introduced the adaptive weight coefficient parameter adjustment mechanism, and this mechanism has achieved good results in solving the problem of jumping out of local convergence. Based on the

parameter adjustment mechanism and combined with reverse learning, this paper forms an improved version of the sine and cosine algorithm with multiple strategies.

The combination of the swarm intelligence algorithm, lion swarm algorithm and reverse learning strategy further expands the search scope of the group, thus improving the problems of a slow convergence speed and insufficient accuracy of the group.

3.1. The Weight Factor

In this part, an adaptive weight w is used, which makes the individual position have a great impact on the individual moving direction and distance in the algorithm and effectively improves the ability of algorithm development. The value of w^{t+1} in the latter iteration is 100 times that of the previous iteration w^t , with an obvious step search. The mathematical model [23] of w is

$$w = \mu \times \sinh\left(1 - 20 \frac{t}{MaxFEs}\right)^8 \quad (6)$$

where μ is the weight factor; in most cases, the value of μ is 0.5. Adding the weight parameter w to the sine cosine algorithm in Equation (1), we obtain:

$$X_i^j(t+1) = \begin{cases} w(X_i^j(t) + r_1 \times \sin(r_2) \times |r_3 X_{best}^j - X_i^j(t)|) & r_4 < 0.5 \\ w(X_i^j(t) + r_1 \times \cos(r_2) \times |r_3 X_{best}^j - X_i^j(t)|) & r_4 \geq 0.5 \end{cases} \quad (7)$$

3.2. Reverse Learning

In SCA, the individuals of the population only rely on the current optimal solution to update their own state, so the algorithm is likely to fall into the local optimal state, resulting in the algorithm being unable to find a satisfactory solution. At this time, it is necessary to carry out a local mutation operation on the individual, and the individual reflects on the previous learning situation with the current learning results so as to increase the probability of escaping from the local area. The formula of reflective learning is

$$X_i^* = X_i^s + \omega \otimes (X_i^s - X_i^t) \quad (8)$$

where X_i^s represents the position of individual i in the t -th iteration; X_i^* represents the position after executing Equation (7); X_i^t represents the new position generated through the reflection process; ω represents a learning factor, $\omega \in [-1, 1]$; \otimes indicates dot multiplication.

In order to prevent too much randomness in the process of reflection, the learning factor was compared with ω . At the same time, in order to avoid the degradation of the learning ability and enhance the convergence of the algorithm, greedy learning was used to select the best algorithm according to the learning status before and after reflection.

$$\omega = C^{(-t/MaxFEs)} \times \cos(r_5) \quad (9)$$

where r_5 is a random number on $[0, \pi]$; C is a constant, and the effect is better when $C = 100$.

In order to reduce the possibility of the algorithm deviating from the global optimal position, the evaluation of excellent algorithms is strengthened. It is very necessary to search the space around the volume, and this improvement can improve the efficiency of the algorithm and the ability to explore new solutions. This paper integrates the strategy of elite reverse learning into SCA. The information of the elite population was used to search the space of elite individuals and their reverse solutions.

The specific operations [24] were as follows:

1. The individuals in the population were arranged after the implementation of formula *fitness*, where 10% of the excellent individuals were selected to form the elite population X_{best} ;
2. Individual $X_{best}^i \in X_{best}$ boundary $[lb_j^i, ub_j^i]$ and the dynamic boundary $[min(lb_j^i), max(ub_j^i)]$ were calculated;

3. The dynamic elite reverse population X'_{best} of individual X_{best}^i was generated according to Equation (10);
4. If the reverse population X'_{best} exceeded the limit of dynamic boundary $[min(lb_j^i), max(ub_j^i)]$, it was replaced by a new individual randomly generated in the boundary;
5. The top 50% from $[X_{best}, X'_{best}]$ was selected for the next generation according to *fitness*;
6. Steps 2 and 5 were cycled until the stop condition was reached, and the algorithm ended.

The elite inverse solution was set in d -dimensional space. $X'_{best} = (x'_1, x'_2, \dots, x'_D)$ is the inverse solution of the elite individual. $X_{best} = (x_1, x_2, \dots, x_D)$ is the inverse solution of the current population. The inverse solution is defined as

$$x'_i = k(lb_i + ub_i) - x_i \quad (10)$$

where $k \in [0, 1]$ is a random number subject to uniform distribution. Multiple inverse solutions of the elite individual can be generated by using this coefficient.

The generated elite inverse solution increases the useful information of the population converging to the global optimum, strengthens the exploration of the neighborhoods around the optimal individual and improves the local development ability of the algorithm.

3.3. The Proposed Hybrid SCA FKNN Model

In this paper, the $Fitness_i$ is equal to ACC. ACC represents the accuracy of FKNN classification, which is obtained by k-cross-validation. In this paper, five-fold validation was used. After combining hybrid SCA and FKNN, the pseudocode of the hybrid SCA FKNN is shown in Algorithm 1. Figures 3 and 4 show the operation flow of the whole hybrid SCA FKNN method in detail.

Algorithm 1 The hybrid SCA-FKNN.

```

while  $t < MaxFEs$  do
  update  $r_1, r_2, r_3, r_4$ 
   $w_t = \mu * \sinh(1 - 20 \frac{t}{MaxFEs})^8$ 
  if  $r_4 < 0.5$  then
     $X_i^s = w_t(X_i(t) + r_1 \times \sin(r_2) \times |r_3 X_{best} - X_i(t)|)$ 
     $X_i^* = X_i^s + \omega \otimes (X_i^s - X_i^t)$ 
    if  $f(X_i^s) > BF$  then
      if  $f(X_i^s) < f(X_i^*)$  then
         $X_i^{t+1} = X_i^*$ 
         $X_{best} = X_i^{t+1}$ 
         $BF = f(X_i^*)$ 
      else
         $X_i^{t+1} = X_i^s$ 
         $X_{best} = X_i^{t+1}$ 
         $BF = f(X_i^*)$ 
      end if
    end if
  end if
  for  $i=1$  to  $SupN$  do
    generate random  $k$ ,  $X'_i(t+1) = k(lb + ub) - X_i(t+1)$ 
  end for
  put all  $X'_i$  into train dataset as elite opposition solutions
else
   $X_i^s = w_t(X_i(t) + r_1 \times \cos(r_2) \times |r_3 X_{best} - X_i(t)|)$ 
  As the up,the same progress
end if
end while

```

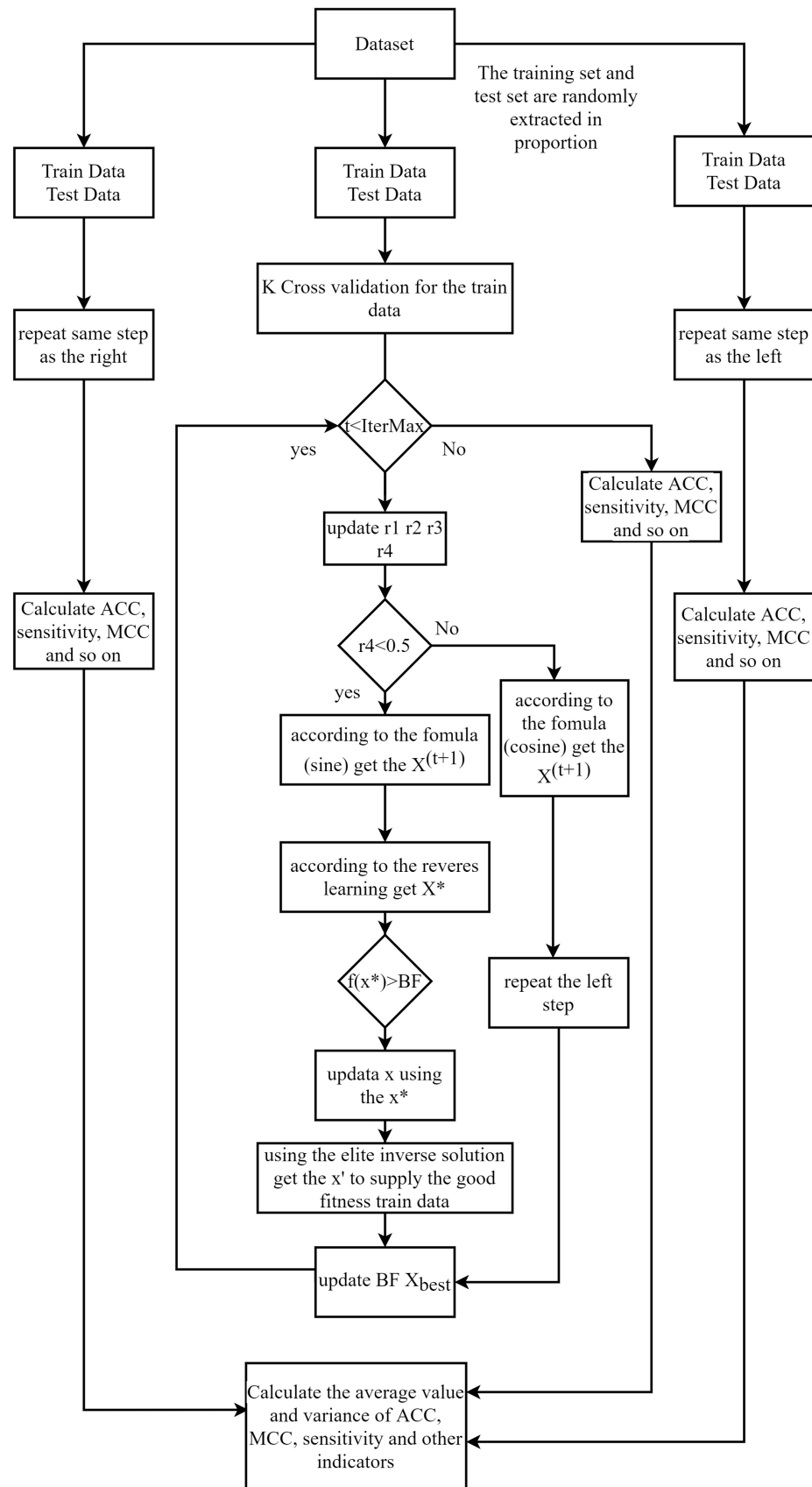


Figure 3. The flow chart of the hybrid SCA FKNN.

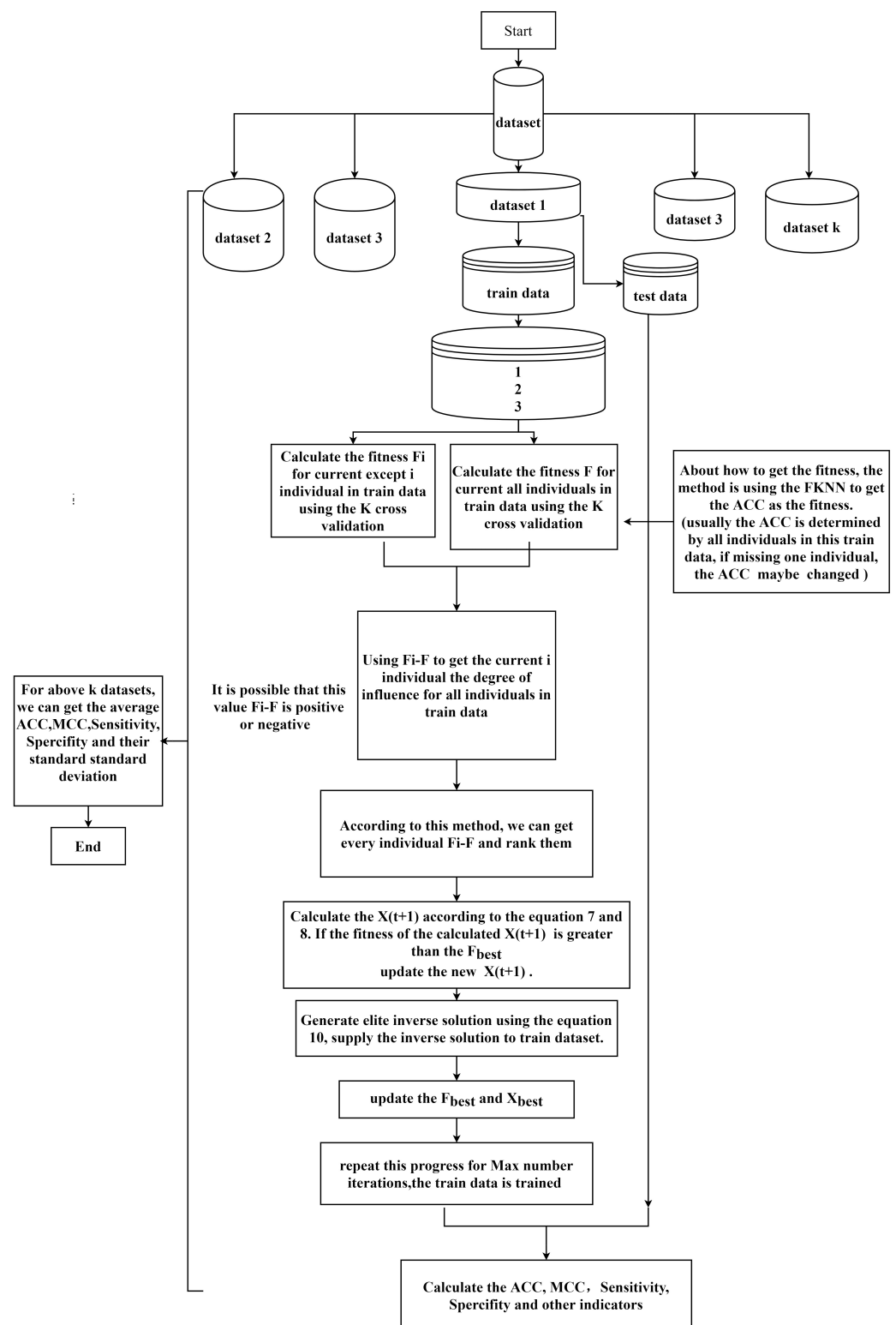


Figure 4. Numerical simulation diagram of the whole process.

4. Experiment and Discussion

4.1. Experiment Setup

In this section, the components of the experiment will be described in detail. The purpose of this experiment is to illustrate that the hybrid SCA FKNN method proposed in this paper can be used in two or more types of data sets and can achieve good numerical

results, with strong adaptability and accuracy. In order to ensure the reproducibility of the experiment, the experimental setup will be shown below.

4.2. Benchmark Datasets

The datasets used in this paper were all open source datasets; for details of the datasets used in this paper, please visit this website: <https://archive.ics.uci.edu/ml/index.php> (accessed on 30 April 2022). For all datasets, the feature was normalized to $[-1, 1]$ using maximum and minimum normalization. In order to better show the numerical results of the method proposed in this paper, this paper focused on the numerical experiments on two-class data sets and multi-class data sets.

In order to reflect the wide adaptability of the methods proposed in this paper, this paper selected 10 kinds of data sets from different application scenarios and different data types. The 10 kinds of data sets involve a variety of use scenarios with practical life significance, such as medical treatment, daily necessities, automobiles, etc. From Tables 1 and 2, it can be seen that the sample size, characteristics and categories of the 10 types of data sets cover different levels. This paper verifies the effectiveness of the method proposed in this paper from various angles according to different conditions and different use scenarios of the data sets.

Table 1. The two-class-dataset-related information.

	Categories	Samples	Features	Positive	Negative
Bupa	2	345	6	145	200
Hepatitis	2	155	19	32	123
SPECT	2	267	22	212	55

Table 2. The multi-class-dataset-related information.

Datasets	Categories	Samples	Features	Positive	Negative
Caesarian section classification dataset	2	80	4	34	46
Indian liver patient dataset (ILPD)	2	583	10	415	167
Glass identification dataset	7	214	9	69 (class 1)	145 (other classes except positive)
User knowledge modeling dataset	4	403	5	102 (class 1)	301 (other classes except positive)
Breast tissue dataset	6	106	9	20 (class 1)	86 (other classes except positive)
Car dataset	4	1728	6	1209 (class 1)	519 (other classes except positive)
QCM sensor alcohol dataset	5	125	15	24 (class 3)	101 (other classes except positive)

For two-class data sets, this paper considered the following three data sets. The basic situation of these three data sets is shown in Table 1. Table 1 describes the indicators, such as the number of data label categories, data scale and data feature quantity, of the following three datasets.

For multi-classes datasets, in order to further verify the effectiveness of the method proposed in this paper, this paper retrieved the following eight types of data sets on the

open data platform of the University of California for numerical experiments. The contents of the relevant data sets are described in Table 2. These data sets involve multiple areas of life, which is more convincing for verifying the effectiveness of the method. The relevant data categories range from the least to the most, and the diversity of data features ranges from small sample data to large sample data.

Table 2 describes the basic information of seven multi-category data sets, including the category, sample number, feature number and number of positive samples and negative samples. For example, the caesarian section classification dataset, Indian liver patient dataset (ILPD), glass identification dataset, user knowledge modeling dataset, car dataset and QCM sensor alcohol dataset. The content of data sets covers all aspects of real life, with a wider range and more complex data types. There are both large sample data sets and small sample data sets, and both multi-feature data sets and a small number of feature data sets.

4.3. Performance Metrics

The evaluation indicators used in numerical experiments include the classification accuracy (ACC), sensitivity, precision, specificity and Matthews correlation coefficient (MCC). Sensitivity refers to the ability of the model to identify positive examples. Precision indicates samples with positive prediction results. These are positive cases. The specificity measurement model is the ability to identify negative examples. The measurement range of MCC is $[-1, 1]$, and the other is $[0, 1]$. The larger the evaluation indicator is, the better the performance of the model under this indicator.

For multi-class datasets, the corresponding concerned data categories are taken as positive categories and other categories are taken as negative categories. The data sets were calculated to obtain the values of relevant evaluation indicators, such as the accuracy (ACC), sensitivity, precision, specificity and Matthews correlation coefficient (MCC).

Standard classification indicators, such as the accuracy (ACC), sensitivity, precision, specificity and Matthews correlation coefficient (MCC), were used in the experiment. According to [25–27], the true positive (TP) is the number of positive instances of correct classification, the false negative (FN) is the number of positive instances of incorrect classification, the true negative (TN) is the number of negative instances of correct classification and the false positive (FP) is the number of positive instances of incorrect classification. The basic configuration matrix of TP, FN, TN and FP is shown in Table 3.

Table 3. The Basic Confusion Matrix.

Basic Confusion Matrix		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

By referring to the papers [25–27], this paper lists the evaluation indicators of accuracy (ACC), sensitivity, precision, specificity and Matthews correlation coefficient (MCC) as follows:

$$ACC = \frac{TP + TN}{TP + TN + FN + FP} \quad (11)$$

$$Precision = \frac{TP}{TP + FP} \quad (12)$$

$$Sensitivity = \frac{TP}{TP + FN} \quad (13)$$

$$Specificity = \frac{TN}{FP + TN} \quad (14)$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (15)$$

The above evaluation indicators can comprehensively evaluate the performance of the proposed model.

4.4. Baseline Methods

In order to verify the hybrid SCA FKNN model, a large number of data experiments were conducted to verify the effectiveness of the proposed method. Firstly, in the first part, this paper tested two classes of data sets, and made a numerical comparison of nine metaheuristic-based algorithms (LSCA [21], SCA [12], PSO [28], SSA [29], SA [30], BA [31], CGSCA [32], mSCA [33] and CESCA [34]), fixing their $M = 2$ and $K = 3$ to verify the advantages of execution data results, respectively. In order to ensure the fairness of the numerical experiment, the experiment was repeated five times on the same machine. Based on the repeated experiment, the average value and standard deviation of the model were analyzed. Each experiment included five cross validation results. The average value of five cross validations was taken for performance evaluation.

4.5. Experimental Design

All numerical experiments were calculated by MATLAB 2017. All experiments were conducted on the same equipment, and 8 GB ram Intel Core i5 (Intel) equipped with windows 11 (Microsoft, Redmond, WA, USA) was used as a workstation to avoid the impact of experimental hardware during the simulation process.

5. Results and Discussion

5.1. Numerical Results for Two-Classes Datasets

For the following three types of two-class datasets, the hybrid SCA FKNN method proposed in this paper will be compared and analyzed with eight metaheuristic algorithms. The superior performance of hybrid SCA FKNN in the evaluation metrics fully shows the advantages of the proposed method, and further verifies the effectiveness of the method.

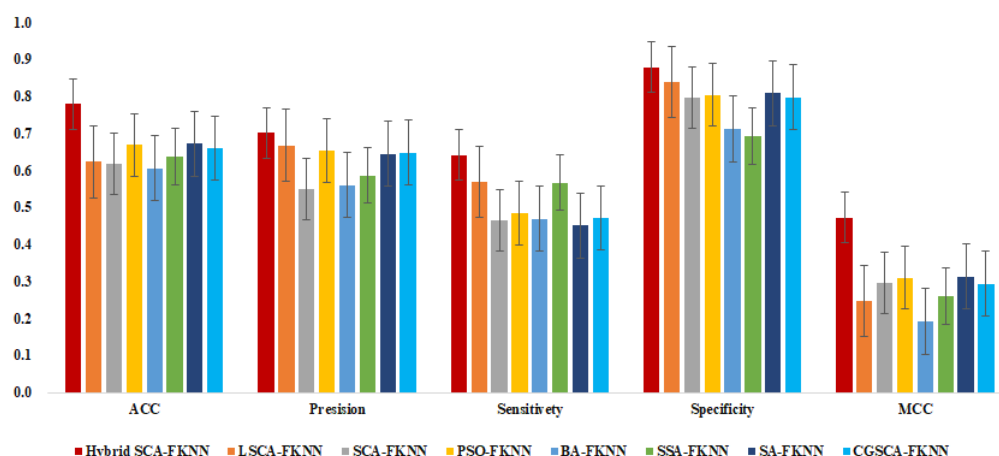
5.1.1. Experimental Results on the Bupa Dataset

The numerical results of the hybrid SCA FKNN compared with other models in the Bupa dataset are shown in Table 4. This paper carried out 10 repeated numerical experiments, and the average value and standard deviation of the 10 repeated experiments are listed in Table 4. It can be seen that the hybrid SCA FKNN model proposed in this paper achieves the best results among the four evaluation indicators. The hybrid SCA FKNN model proposed in this paper observes better results on ACC, which are approximately 8.4–25.1% higher than the comparison models. Although the standard deviation of LSCA-FKNN is lower than that of hybrid SCA FKNN in most cases, the numerical results of hybrid SCA FKNN are significantly better than LSCA FKNN in terms of average evaluation index values.

In order to show the overall benchmarking analysis results of each model, Figure 5 draws a bar graph of the performance of each model, draws the average value of 10 repeated experiments of each model and adds the standard deviation of repeated experiments as the error line. Figure 5 is a visual display of Table 4. As shown in Figure 5, except for precision, good results have been achieved in sensitivity, specificity and MCC. It can be clearly seen from the figure that the hybrid SCA FKNN method has better numerical results and stronger stability.

Table 4. Results of the hybrid FKNN and comparison models on the Bupa dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.7799	0.7015	0.6412	0.8791	0.4728
	std	0.0143	0.0284	0.0299	0.0283	0.0303
LSCA-FKNN	avg	0.6232	0.6674	0.5687	0.8393	0.2465
	std	0.0199	0.0338	0.0175	0.0177	0.0367
SCA-FKNN	avg	0.6175	0.5494	0.4645	0.7968	0.2946
	std	0.0383	0.0439	0.0487	0.0178	0.0546
PSO-FKNN	avg	0.6686	0.6531	0.4851	0.8047	0.3105
	std	0.0266	0.0344	0.0379	0.0231	0.0472
BA-FKNN	avg	0.6056	0.5600	0.4693	0.7121	0.1920
	std	0.0292	0.0479	0.0423	0.0292	0.0694
SSA-FKNN	avg	0.6377	0.5862	0.5667	0.6923	0.2601
	std	0.0586	0.0069	0.1233	0.0327	0.0895
SA-FKNN	avg	0.6721	0.6444	0.4511	0.8087	0.3131
	std	0.0142	0.0340	0.0128	0.0415	0.0279
CGSCA-FKNN	avg	0.6600	0.6486	0.4711	0.7981	0.2939
	std	0.0193	0.0356	0.0195	0.0308	0.0382

**Figure 5.** Classification performance of each model on the Bupa dataset.

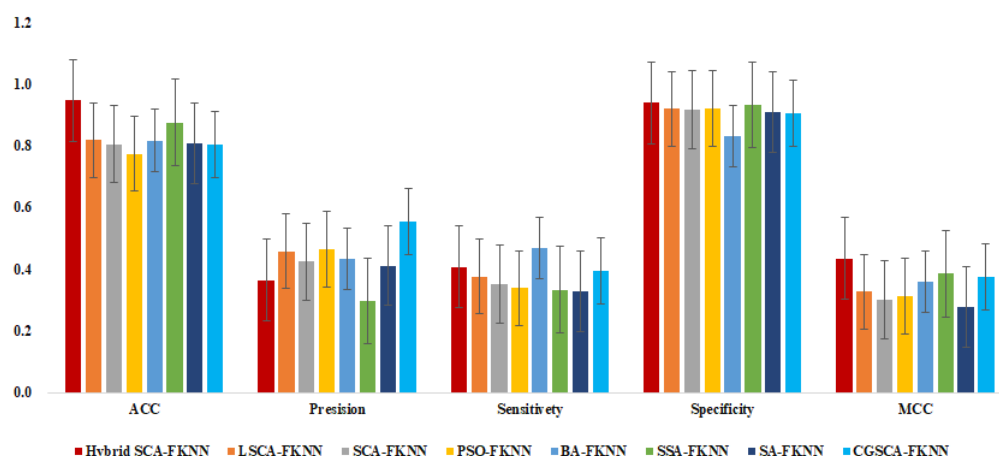
5.1.2. Experimental Results on the Hepatitis Dataset

As reflected in the above table on the Bupa dataset, Table 5 shows the benchmarking results between the hepatitis dataset and other model methods. It can be seen that the hybrid SCA FKNN model proposed in this paper achieves the best results among the three evaluation indicators. In terms of sensitivity and precision, the hybrid SCA FKNN performs worse than BA FKNN and CGSCA FKNN, but significantly better than their numerical results in terms of ACC, specificity and MCC. The hybrid SCA FKNN model proposed in this paper observes better results on ACC, which are approximately 15.2–19.6% higher than the comparison models.

Similarly, in order to further visualize the comparison of the five types of evaluation indicators, Figure 6 shows the performance of each model more intuitively. For this data set, the data size of the positive samples is small, at only 32 data, so it has a great impact on the sensitivity and MCC. In general, the hybrid SCA FKNN model has competitive advantages.

Table 5. Results of the hybrid-FKNN and comparison models on the hepatitis dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.9465	0.3638	0.4072	0.9392	0.4342
	std	0.0569	0.0937	0.0845	0.0312	0.0945
LSCA-FKNN	avg	0.8191	0.4566	0.3760	0.9192	0.3276
	std	0.0296	0.0874	0.0676	0.0303	0.0648
SCA-FKNN	avg	0.8051	0.4236	0.3512	0.9167	0.3009
	std	0.0323	0.1061	0.0745	0.0301	0.0875
PSO-FKNN	avg	0.7742	0.4641	0.3376	0.9217	0.3118
	std	0.0378	0.0969	0.0727	0.0230	0.0788
BA-FKNN	avg	0.8172	0.4333	0.4692	0.8305	0.3593
	std	0.0233	0.0702	0.0903	0.0267	0.0812
SSA-FKNN	avg	0.8750	0.2975	0.3333	0.9333	0.3846
	std	0.0432	0.0379	0.1925	0.0087	0.0098
SA-FKNN	avg	0.8076	0.4115	0.3280	0.9097	0.2771
	std	0.0242	0.0913	0.0751	0.0266	0.0797
CGSCA-FKNN	avg	0.8033	0.5533	0.3944	0.9067	0.3744
	std	0.0199	0.0740	0.0666	0.0266	0.0582

**Figure 6.** Classification performance of each model on the hepatitis dataset.

ACC is an index used to describe the accuracy of the model. The higher the value of ACC, the better the prediction result of the model. For the current hepatitis dataset, Table 5 is obtained according to the comparison with the numerical results in paper [21]. As shown in Table 5 and Figure 6, the hybrid-FKNN model proposed in this paper obtains better results in ACC, which are approximately 15.2–19.6% higher than the comparison model. However, its variance is high and its stability is poor. The data results are greatly affected by random data sampling.

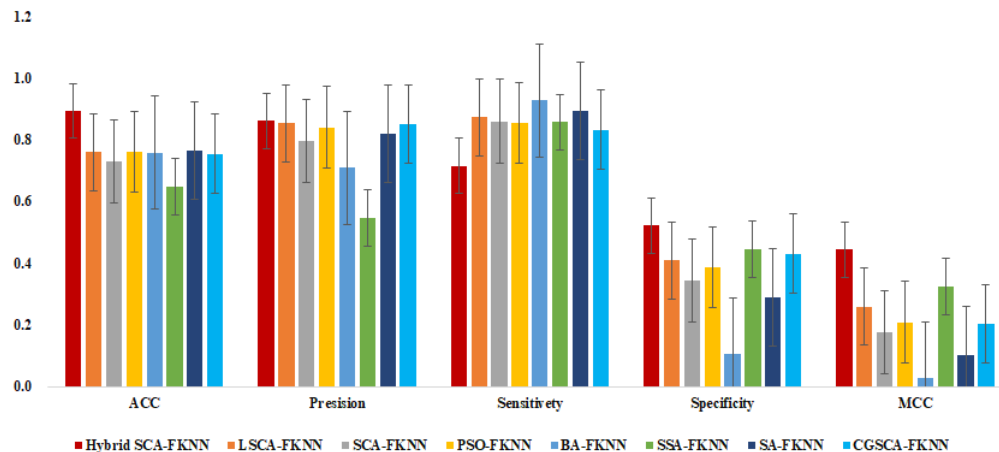
5.1.3. Experimental Results on the SPECT Dataset

Similarly, Table 6 shows the benchmarking results of the SPECT dataset and other model methods on the evaluation indicators of ACC, sensitivity, precision and MCC. In terms of sensitivity and precision, the hybrid SCA FKNN performs worse than BA FKNN and LSCA FKNN. BA FKNN has a high sensitivity and low specificity, and there is a significant difference in its ability to recognize positive cases and negative cases. However, in terms of ACC, specification and MCC, it is significantly better than their numerical results.

Table 6. Results of the hybrid-FKNN and comparison models on the SPECT dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.8936	0.8620	0.7157	0.5220	0.4436
	std	0.0195	0.0845	0.0610	0.0227	0.0605
LSCA-FKNN	avg	0.7593	0.8538	0.8730	0.4094	0.2588
	std	0.0191	0.0187	0.0357	0.0561	0.0656
SCA-FKNN	avg	0.7297	0.7953	0.8601	0.3427	0.1759
	std	0.0449	0.0283	0.0278	0.0707	0.0718
PSO-FKNN	avg	0.7615	0.8405	0.8541	0.3866	0.2079
	std	0.0351	0.0216	0.0308	0.0637	0.0975
BA-FKNN	avg	0.7585	0.7098	0.9270	0.1049	0.0259
	std	0.0287	0.0064	0.0164	0.0493	0.0702
SSA-FKNN	avg	0.6471	0.5455	0.8571	0.4443	0.3228
	std	0.0899	0.0951	0.0825	0.0673	0.0943
SA-FKNN	avg	0.7658	0.8195	0.8920	0.2891	0.1031
	std	0.0164	0.0176	0.0234	0.1013	0.0637
CGSCA-FKNN	avg	0.7546	0.8497	0.8330	0.4308	0.2034
	std	0.0229	0.0140	0.0238	0.0587	0.0709

Figure 7 vividly shows the performance of various models on different evaluation indicators. It can be seen from the figure that the sensitivity, specificity and MCC indicators of the model vary greatly, which may be due to the small number of negative samples, at only 55 negative samples. Cross validation has a great impact on the sensitivity and MCC. It can be seen from the figure that the hybrid SCA FKNN method is quite competitive in obtaining numerical results on SPECT data sets.

**Figure 7.** Classification performance of each model on the SPECT dataset.

For the current SPECT dataset, Table 6 is also obtained by the hybrid SCA FKNN. As shown in Table 6, the hybrid-FKNN model obtains better results in ACC, which are approximately 12.1–14.3% higher than the comparison models. As shown in Figure 7, except for sensitivity and precision, the best results were also achieved in specificity and MCC. However, satisfactory results have been achieved in sensitivity and precision. The numerical results are quite competitive.

In order to further verify the effectiveness of the proposed method, the standard is attached in Appendix A, which concerns the experimental results obtained in hepatitis, Bupa and SPECT datasets under the conditions of different maximum cycle test times and different cross validation numbers for researchers' reference.

5.2. Numerical Results for Multi-Class Datasets

For the above seven different application scenarios and different types of data sets, the following data results are obtained in this paper. As shown in Table 7, for multi-class datasets, the data prediction accuracy is between 0.65–0.90, and the hybrid-SCA FKNN method can still achieve good results. It has good data prediction results for datasets with multiple or few characteristics, and multiple or few samples. This method is more adaptable.

Table 7. Results of the hybrid-FKNN on multi-class datasets.

Datasets	Metric	ACC	Precision	Sensitive	Specificity	MCC
Caesarian section classification dataset	Avg	0.7026	0.7197	0.8336	0.7049	0.4694
	Std	0.0901	0.0500	0.0160	0.0927	0.0747
Indian Liver Patient Dataset (ILPD)	Avg	0.7953	0.7876	0.8255	0.3788	0.1621
	Std	0.0342	0.0379	0.0853	0.0541	0.0437
Glass Identification Dataset	Avg	0.7827	0.9016	0.9526	0.8347	0.4734
	Std	0.0355	0.0501	0.0376	0.0751	0.1037
User Knowledge Modeling Dataset	Avg	0.8606	0.9545	0.9709	0.9185	0.9042
	Std	0.0267	0.0408	0.0163	0.0864	0.0319
Breast Tissue Dataset	Avg	0.6554	0.9667	0.8883	0.9770	0.6969
	Std	0.0727	0.0577	0.0459	0.1443	0.1163
Car Dataset	Avg	0.8807	0.9188	0.9973	0.9823	0.8312
	Std	0.0916	0.0612	0.0716	0.0982	0.1616
QCM sensor Alcohol Dataset	Avg	0.9043	0.8501	0.8568	0.8477	0.8562
	Std	0.0939	0.0838	0.0719	0.0973	0.1008

In order to better demonstrate the effectiveness of the method proposed in this paper, this paper compares the numerical results of the above seven different data sets calculated by the LSCA FKNN [21], SCA FKNN [12], PSO FKNN [28], BA FKNN [31] and SSA FKNN [29] methods. The relevant numerical comparison results are shown in Tables 8–14. From the five evaluation indicators, the hybrid SCA FKNN method proposed in this paper achieved good numerical results under seven datasets.

For the current caesarian section classification dataset, Table 8 is also obtained from hybrid SCA FKNN. As shown in Table 8, the hybrid SCA FKNN model achieved better results in ACC, sensitivity and MCC, which were approximately 5.2–21.7% higher than the comparison model in ACC.

For the Indian liver patient dataset (ILPD), Table 9 shows the data results obtained by the hybrid SCA FKNN method and other numerical models. As shown in Table 9, the hybrid SCA FKNN model achieved better results in ACC, precision, specificity and MCC, which were approximately 10.9–23.2% higher than the comparison model in ACC.

Table 8. Results of the hybrid-FKNN and comparison models on the caesarian section classification dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.7026	0.7197	0.8336	0.7049	0.4694
	std	0.0901	0.0500	0.0160	0.0927	0.0747
LSCA-FKNN	avg	0.6677	0.8148	0.5741	0.7095	0.3923
	std	0.0955	0.1197	0.0986	0.0965	0.1295
SCA-FKNN	avg	0.6667	0.6766	0.6349	0.7333	0.3626
	std	0.0722	0.2384	0.0755	0.0882	0.0414
PSO-FKNN	avg	0.6675	0.8194	0.7505	0.7250	0.4045
	std	0.0701	0.1138	0.1220	0.0992	0.0939
BA-FKNN	avg	0.5625	0.5361	0.7424	0.2500	0.2655
	std	0.0625	0.0804	0.0957	0.0443	0.0995
SSA-FKNN	avg	0.5775	0.6528	0.5370	0.5952	0.2381
	std	0.0523	0.0241	0.0656	0.0591	0.0817

Table 9. Results of the hybrid-FKNN and comparison models on the Indian liver patient dataset (ILPD). (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.7953	0.7876	0.8255	0.3788	0.1621
	std	0.0342	0.0379	0.0853	0.0541	0.0437
LSCA-FKNN	avg	0.6912	0.6961	0.9856	0.1078	0.0712
	std	0.0191	0.0123	0.0220	0.0309	0.0257
SCA-FKNN	avg	0.6455	0.7440	0.8882	0.2058	0.1038
	std	0.0222	0.0732	0.0964	0.0821	0.0674
PSO-FKNN	avg	0.7173	0.7187	0.9967	0.0963	0.0953
	std	0.0139	0.0162	0.0058	0.0107	0.0641
BA-FKNN	avg	0.7108	0.7209	0.9534	0.0865	0.0532
	std	0.0085	0.0076	0.0034	0.0012	0.0125
SSA-FKNN	avg	0.7092	0.7215	0.9695	0.0502	0.0644
	std	0.0028	0.0140	0.0277	0.0457	0.0112

Table 10. Results of the hybrid-FKNN and comparison models on the glass dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.7827	0.9016	0.9526	0.8347	0.4734
	std	0.0355	0.0501	0.0376	0.0751	0.1037
LSCA-FKNN	avg	0.6589	0.5961	0.6759	0.7618	0.4242
	std	0.0355	0.2470	0.1530	0.0546	0.1972
SCA-FKNN	avg	0.6654	0.6078	0.8614	0.7028	0.5339
	std	0.0355	0.0453	0.0558	0.0337	0.0394
PSO-FKNN	avg	0.6047	0.6429	0.6923	0.7727	0.4587
	std	0.0968	0.0415	0.0994	0.0646	0.0950
BA-FKNN	avg	0.5349	0.3333	0.5000	0.6429	0.3187
	std	0.1005	0.0907	0.0874	0.0707	0.0587
SSA-FKNN	avg	0.7209	0.7778	0.9091	0.6875	0.6209
	std	0.0880	0.0128	0.1905	0.1168	0.2040

Table 11. Results of the hybrid-FKNN and comparison models on the user modeling dataset hamdi tolga dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.8606	0.9545	0.9709	0.9185	0.9042
	std	0.0267	0.0408	0.0163	0.0864	0.0319
LSCA-FKNN	avg	0.7901	0.9286	0.9286	0.9808	0.9093
	std	0.0317	0.0299	0.0469	0.0127	0.0388
SCA-FKNN	avg	0.7977	0.9107	0.9639	0.9678	0.9143
	std	0.0744	0.0233	0.0313	0.0138	0.0400
PSO-FKNN	avg	0.7713	0.8860	0.9434	0.9444	0.8876
	std	0.0681	0.0218	0.0246	0.0059	0.0395
BA-FKNN	avg	0.5179	0.6324	0.5404	0.8189	0.3828
	std	0.0890	0.1101	0.3625	0.1877	0.2242
SSA-FKNN	avg	0.8025	0.9437	0.7593	0.9541	0.7018
	std	0.0377	0.0150	0.0590	0.0153	0.0629

Table 12. Results of the hybrid-FKNN and comparison models on the breast tissues dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.6554	0.9667	0.8883	0.9770	0.6969
	std	0.0727	0.0577	0.0459	0.1443	0.1163
LSCA-FKNN	avg	0.5397	0.8667	0.6389	0.9333	0.6154
	std	0.0727	0.2309	0.1273	0.1155	0.2682
SCA-FKNN	avg	0.5373	0.7500	0.8167	0.9024	0.6471
	std	0.0550	0.0012	0.1243	0.0117	0.0937
PSO-FKNN	avg	0.5238	0.6583	0.8333	0.7500	0.5677
	std	0.0991	0.0366	0.0787	0.0605	0.0879
BA-FKNN	avg	0.4928	0.5843	0.6875	0.6742	0.4731
	std	0.0727	0.0473	0.0887	0.0751	0.0949
SSA-FKNN	avg	0.6667	0.9487	0.8196	0.6667	0.5657
	std	0.0825	0.0888	0.0239	0.0774	0.0865

Table 13. Results of the hybrid-FKNN and comparison models on the car dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.8807	0.9188	0.9973	0.9823	0.8312
	std	0.0916	0.0612	0.0716	0.0982	0.1616
LSCA-FKNN	avg	0.8691	0.9450	0.9467	0.8388	0.7922
	std	0.0161	0.0372	0.0289	0.1106	0.0454
SCA-FKNN	avg	0.8168	0.8789	0.9559	0.9428	0.7818
	std	0.0134	0.0095	0.0057	0.0271	0.0110
PSO-FKNN	avg	0.7645	0.7718	0.9977	0.3179	0.4601
	std	0.0159	0.0964	0.0434	0.0530	0.0064
BA-FKNN	avg	0.7107	0.7342	0.9468	0.6591	0.6596
	std	0.0859	0.1090	0.0350	0.1100	0.1226
SSA-FKNN	avg	0.8618	0.9461	0.8816	0.8307	0.6653
	std	0.0854	0.0939	0.0269	0.0819	0.0945

Table 14. Results of the hybrid-FKNN and comparison models on the QCM sensor alcohol dataset. (Bold indicates the best in comparison method).

Algorithm	Metric	ACC	Precision	Sensitivity	Specificity	MCC
Hybrid SCA-FKNN	avg	0.9043	0.8501	0.8568	0.8477	0.8562
	std	0.0939	0.0838	0.0719	0.0973	0.1008
LSCA-FKNN	avg	0.7600	0.7917	0.8327	0.8189	0.7269
	std	0.0367	0.0908	0.0823	0.0925	0.1415
SCA-FKNN	avg	0.7702	0.7333	0.7333	0.8841	0.7479
	std	0.0693	0.0887	0.1082	0.0275	0.1270
PSO-FKNN	avg	0.4400	0.5095	0.6656	0.6794	0.2208
	std	0.0400	0.0744	0.0672	0.0531	0.0580
BA-FKNN	avg	0.3067	0.6567	0.2611	0.7804	0.3568
	std	0.0231	0.0333	0.0674	0.0756	0.0632
SSA-FKNN	avg	0.8133	0.6640	0.7714	0.8682	0.6117
	std	0.1007	0.0856	0.0960	0.0304	0.1126

For the glass dataset, Table 10 shows the data results obtained by the hybrid SCA FKNN method and other numerical models. As shown in Table 10, the hybrid SCA FKNN model achieved better results in ACC, precision, sensitivity specificity and MCC, which were approximately 8.6–46.3% higher than the comparison model in ACC.

For the user modeling dataset hamdi tolga dataset, Table 11 shows the data results obtained by the hybrid SCA FKNN method and other numerical models. As shown in Table 11, the hybrid SCA FKNN model achieved better results in ACC, precision, sensitivity

and specificity, which were approximately 7.2–66.2% higher than the comparison model in ACC.

For the breast tissues dataset, Table 12 shows the data results obtained by the hybrid SCA FKNN method and other numerical models. As shown in Table 12, the hybrid SCA FKNN model achieved better results in precision, sensitivity specificity and MCC, which were slightly lower than SSA-FKNN and higher than the comparison model in ACC.

For the car dataset, Table 13 shows the data results obtained by the hybrid SCA FKNN method and other numerical models. As shown in Table 13, the hybrid SCA FKNN model achieved better results in ACC, specificity and MCC, which were approximately 1.3–8.6% higher than the comparison model in ACC.

For the QCM sensor alcohol dataset, Table 14 shows the data results obtained by the hybrid SCA FKNN method and other numerical models. As shown in Table 14, the hybrid SCA FKNN model achieved better results in ACC, precision, sensitivity and MCC, which were twice as large as the results of PSO FKNN and BA FKNN in ACC.

In order to better verify the method proposed in this paper, this paper analyzed the operation process of the five data sets. The change in best fitness can analyze the convergence of the data running process. If the best fitness does not change during the cycle, the data prediction results do not change in general. If a few cycles are used, the best fitness will not change, indicating that the method has a faster convergence speed.

From Figure 8, it can be seen that the optimal fitness will start to be in a stable state at approximately 12–23 iterations; that is, in the next cycle, the data of the training set will not be optimized, and the method will be in a convergent state. As shown in Figure 8, after the first iteration, the best fitness of the current training set will be obtained, and the current best fitness will not be equal to 0. The value of this best fitness gradually stabilizes with the increase in the number of iterations, but it does not start from 0. For the car dataset, QCM sensor alcohol dataset and glass identification dataset, the best fitness converges quickly and converges to a stable value at approximately the 10th iteration. For the ILPD and the breast tissue dataset, it also converges to a stable value at approximately the 15th iteration.

With the increase in the number of iterations, the best fitness gradually increases on the car dataset. When the number of iterations reaches 15, the best fitness gradually stabilizes to 0.8543. With the update and optimization of the car training dataset, the data prediction result of the car test set reaches 0.8514. For the QCM sensor alcohol dataset, when the number of iterations reaches 10, the optimal fitness reaches a stable state at 0.8400, and the data prediction result of the QCM sensor alcohol test set reaches 0.8520. For the glass identification dataset, when the number of iterations reaches 12, the optimal fitness reaches a stable state at 0.7256, and the data prediction result of the glass identification test set reaches 0.7364. The classification accuracies of the above three data sets in the cycle process, best fitness and test data sets are roughly the same, which shows that this method can find convergent data points in the simulation process effectively and obtain more accurate results.

For the ILPD, when the number of iterations reaches 20, the optimal fitness reaches a stable state at 0.7256, and the data prediction result of the ILPD test set reaches 0.6609. For the breast tissue dataset, when the number of iterations reaches 15, the optimal fitness reaches a stable state at 0.7256, and the data prediction result of the breast tissue test set reaches 0.6475. For the above two datasets, although the accuracy of the method is slightly lower than that of best fitness, the method achieves better results in precision and stability.

According to Figure 8, it can be clearly seen that the method proposed in this paper has a good convergence effect and achieves the purpose of prediction results when fewer cycles are required.

For further experiments, and to compare the results of LSCA-FKNN and SCA-FKNN for all 10 datasets, this paper will run Wilcoxon signed rank test hybrid SCA FKNN. The null hypothesis is that there is no difference between the accuracy of the two classifiers. As shown in the accuracy results of this paper, we reject the null hypothesis (significance level

0.05), and accept that the two classifiers have significantly different accuracies. This result confirms the advantages of hybrid SCA FKNN.

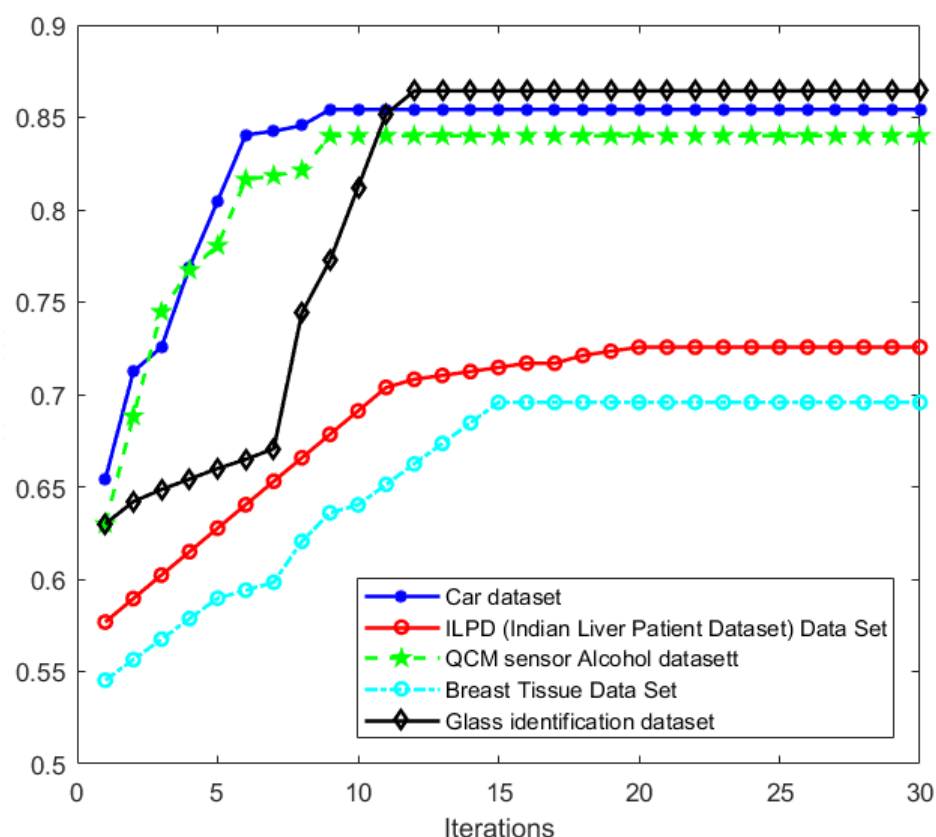


Figure 8. The process of best fitness change with increasing number of iterations on multi-class dataset.

6. Conclusions

The hybrid SCA FKNN algorithm is proposed in this paper based on the improved SCA algorithm combined with reverse learning and an FKNN classifier. It is a further combination of the swarm intelligence algorithm and classifier. This method is a multi-strategy hybrid algorithm that further optimizes the sine cosine algorithm, makes it easier to jump out of local convergence and obtains more accurate numerical solutions. In the process of implementing hybrid SCA into FKNN, this paper mainly uses the FKNN classifier to calculate the prediction accuracy by cross validation as the current best fitness to iteratively optimize the training dataset in order to obtain a more accurate classification. This way, the training set population can be optimized until the training set cannot be optimized any more and the numerical value converges, which can greatly improve the accuracy of the numerical results. After comparing the numerical results of the hybrid SCA FKNN method with five other methods in 10 data sets, and through the Wilcoxon signed rank test with SCA-FKNN and LSCA-FKNN, the numerical results were significantly improved.

In the next step, we will further consider logic mining to improve the method [27,35–38] and integrate multiple patterns to optimize swarm intelligence algorithms and obtain a more efficient method. In the process of using FKNN to calculate fitness, this method repeatedly calculates the location distance, requires a lot of numerical calculations and takes a large amount of time. The next step is to consider a more efficient classifier to reduce the time cost. In the next step, we will further optimize the model in combination with spiritual Gaussian mutation [39] to improve the accuracy.

Author Contributions: C.Z.: conceptualization, methodology, software, data curation, writing—original draft preparation; M.S.M.K.: supervision, visualization, investigation, review; M.A.M.: review, editing, supervision; J.C.: writing, validation; Y.G.: review, editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research is fully funded and supported by Universiti Sains Malaysia, Short Term Grant, 304/PMATHS/6315655.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Please refer to data availability statements in the UC Irvine Machine Learning Repository at <https://archive.ics.uci.edu/ml/> (accessed on 30 April 2022).

Acknowledgments: We would like to acknowledge “Universiti Sains Malaysia, Short Term Grant, 304/PMATHS/6315655” for the support and funding.

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

Abbreviations

Notation	Explanation
Hybrid SCA	The hybrid algorithm proposed based on the sine cosine algorithm and reverse learning
SCA	Sine cosine algorithm
LSCA	The linear population size reduction sine and cosine algorithm
PSO	Particle swarm optimization
BA	Bat algorithm
SSA	Sparrow search algorithm
SA	Salp swarm algorithm
CGSCA	Cauchy and Gaussian sine cosine optimization
FKNN	Fuzzy k-nearest neighbor

Appendix A

Table A1. Numerical results for three two-class datasets in different MaxFEs when $K = 5$ -fold cross-validation.

MaxFEs	Datasets	Metric	ACC	Precision	Sensitive	Specificity	MCC
5	Hepatitis dataset	avg	0.8280	0.5417	0.2142	0.9583	0.2426
		std	0.0660	0.1021	0.1241	0.0417	0.1102
	Bupa dataset	avg	0.6663	0.6245	0.4947	0.7913	0.2985
		std	0.0366	0.0625	0.0949	0.0367	0.0738
	SPECT dataset	avg	0.5934	0.5667	0.7164	0.4515	0.1978
		std	0.1287	0.1115	0.2763	0.1400	0.2842
10	Hepatitis dataset	avg	0.8017	0.6667	0.2564	0.9420	0.2785
		std	0.0501	0.0946	0.0943	0.1004	0.0954
	Bupa dataset	avg	0.6979	0.7000	0.5122	0.8364	0.372
		std	0.0483	0.0486	0.0643	0.0303	0.0514
	SPECT dataset	avg	0.5934	0.5515	0.7655	0.4344	0.2248
		std	0.1077	0.0973	0.1998	0.1534	0.2218
20	Hepatitis dataset	avg	0.8526	0.5833	0.4500	0.9472	0.4149
		std	0.0412	0.0174	0.0500	0.0273	0.0540
	Bupa dataset	avg	0.7576	0.6500	0.5909	0.8409	0.4429
		std	0.0582	0.0284	0.0299	0.0283	0.0303
	SPECT dataset	avg	0.6127	0.5681	0.7524	0.4601	0.2353
		std	0.1038	0.0951	0.0999	0.0706	0.0937

Table A2. Numerical results for three two-class datasets in different-fold cross-validation when $MaxFEs = 5$.

Fold Cross-Validation	Datasets	Metric	ACC	Precision	Sensitive	Specificity	MCC
K = 3	Hepatitis dataset	avg	0.8065	0.5000	0.2001	0.9872	0.2418
		std	0.0559	0.1421	0.0854	0.0222	0.1120
	Bupa dataset	avg	0.6338	0.6775	0.5295	0.8276	0.3751
		std	0.0692	0.1268	0.1104	0.0599	0.1517
	SPECT dataset	avg	0.6617	0.6553	0.7456	0.5931	0.3701
		std	0.1165	0.0998	0.2663	0.1623	0.2376
K = 5	Hepatitis dataset	avg	0.7957	0.5222	0.2762	0.9338	0.2673
		std	0.0186	0.1347	0.1288	0.0426	0.0877
	Bupa dataset	avg	0.6663	0.6245	0.495	0.7913	0.2985
		std	0.0366	0.0625	0.0949	0.0367	0.0738
	SPECT dataset	avg	0.5934	0.5667	0.7164	0.4515	0.1978
		std	0.1287	0.1115	0.2763	0.1400	0.2842
K = 8	Hepatitis dataset	avg	0.8065	0.7222	0.2050	0.9725	0.3000
		std	0.0645	0.2546	0.0556	0.0242	0.0343
	Bupa dataset	avg	0.6717	0.6310	0.4786	0.8075	0.3033
		std	0.0544	0.0938	0.0879	0.0456	0.1204
	SPECT dataset	avg	0.5826	0.5176	0.8714	0.4172	0.3164
		std	0.1134	0.2038	0.1384	0.2307	0.1509

References

- Wang, C.-N.; Yang, F.-C.; Nguyen, V.T.T.; Nguyen, Q.M.; Huynh, N.T.; Huynh, T.T. Optimal design for compliant mechanism flexure hinges: Bridge-type. *Micromachines* **2021**, *12*, 1304. [\[CrossRef\]](#) [\[PubMed\]](#)
- Nguyen, T.V.; Huynh, N.-T.; Vu, N.-C.; Kieu, V.N.; Huang, S.-C. Optimizing compliant gripper mechanism design by employing an effective bi-algorithm: fuzzy logic and anfis. *Microsyst. Technol.* **2021**, *27*, 3389–3412. [\[CrossRef\]](#)
- Chau, N.L.; Dao, T.-P.; Nguyen, V.T.T. Optimal design of a dragonfly-inspired compliant joint for camera positioning system of nanoindentation tester based on a hybrid integration of jaya-anfis. *Math. Probl. Eng.* **2018**, *2018*, 8546095. [\[CrossRef\]](#)
- Liu, H.; Wen, Z.; Cai, W. Fastpsa: Towards efficient swarm intelligence algorithm on GPUs. In Proceedings of the ICPP 2021: 50th International Conference on Parallel Processing, Lemont, IL, USA, 9–12 August 2021.
- Du, Y.; Chen, W.; Fan, B. Research and application of swarm intelligence algorithm in path planning. *Electron. Meas. Technol.* **2016**, *39*, 65–70.
- Duan, H.; Qiao, P. Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [\[CrossRef\]](#)
- Xu, H.; Guan, H.; Liang, A.; Yan, X. A multi-robot pattern formation algorithm based on distributed swarm intelligence. In Proceedings of the 2010 Second International Conference on Computer Engineering and Applications, Bali, Indonesia, 19–21 March 2010; IEEE: Piscataway, NJ, USA, 2010; Volume 1, pp. 71–75.
- Verma, O.P.; Gupta, S.; Goswami, S.; Jain, S. Opposition based modified particle swarm optimization algorithm. In Proceedings of the 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, India, 3–5 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6.
- Wang, X.; Li, Z.Y.; Xu, G.Y.; Yan, W. Artificial bee colony algorithm based on chaos local search operator. *J. Comput. Appl.* **2012**, *32*, 1033–1036. [\[CrossRef\]](#)
- Xu, Y.; Zhou, J.; Xue, X.; Fu, W.; Li, C. An adaptively fast fuzzy fractional order pid control for pumped storage hydro unit using improved gravitational search algorithm. *Energy Convers. Manag.* **2016**, *111*, 67–78. [\[CrossRef\]](#)
- Kaveh, A.; Dadras, A. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Adv. Eng. Softw.* **2017**, *110*, 69–84. [\[CrossRef\]](#)
- Mirjalili, S. Sca: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [\[CrossRef\]](#)
- Yong, L.I.U.; Liang, M.A. Sine cosine algorithm with nonlinear decreasing conversion parameter. *Comput. Eng. Appl.* **2017**, *53*, 1–5.
- Cong, C.; Liang, M.; Yong, L.; School, B. Quantum sine cosine algorithm for function optimization. *Appl. Res. Comput.* **2017**, *34*, 3214–3218.
- Abd Elaziz, M.; Oliva, D.; Xiong, S. An improved opposition-based sine cosine algorithm for global optimization. *Expert Syst. Appl. Int. J.* **2017**, *90*, 484–500. [\[CrossRef\]](#)
- Singh, N.; Singh, S.B. A novel hybrid gwo-sca approach for optimization problems. *Eng. Sci. Technol. Int. J.* **2017**, *20*, 1586–1601. [\[CrossRef\]](#)

17. Abd Elaziz, M.E.; Ewees, A.A.; Oliva, D.; Duan, P.; Xiong, S. A hybrid method of sine cosine algorithm and differential evolution for feature selection. In Proceedings of the international Conference on Neural Information Processing, Long Beach, CA, USA, 4–9 December 2017; Springer: Cham, Switzerland, 2017; pp. 145–155.
18. Rizk-Allah, R.M. Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems. *J. Comput. Des. Eng.* **2017**, *5*, 249–273. [\[CrossRef\]](#)
19. Long, W.; Wu, T.; Liang, X.; Xu, S. Solving high-dimensional global optimization problems using an improved sine cosine algorithm. *Expert Syst. Appl.* **2019**, *123*, 108–126. [\[CrossRef\]](#)
20. Nenavath, H.; Jatoth, R.K. Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. *Appl. Soft Comput.* **2018**, *62*, 1019–1043. [\[CrossRef\]](#)
21. Wu, S.; Mao, P.; Li, R.; Cai, Z.; Chen, X. Evolving fuzzy k-nearest neighbors using an enhanced sine cosine algorithm: Case study of lupus nephritis. *Comput. Biol. Med.* **2021**, *135*, 104582. [\[CrossRef\]](#)
22. Keller, J.M.; Gray, M.R.; Givens, J.A. A fuzzy k-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.* **1985**, *4*, 580–585. [\[CrossRef\]](#)
23. Lin, J.; He, Q. Mixed strategy to improve sine cosine algorithm. *Appl. Res. Comput.* **2020**, *37*, 6.
24. Wachowiak, M.P.; Smolík, R.; Zheng, Y.; Zurada, J.M.; Elmaghraby, A.S. An approach to multimodal biomedical image registration utilizing particle swarm optimization. *Evol. Comput. IEEE Trans.* **2004**, *8*, 289–301. [\[CrossRef\]](#)
25. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; Fujita, H. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl.-Based Syst.* **2018**, *154*, 43–67. [\[CrossRef\]](#)
26. Jha, K.; Saha, S. Incorporation of multimodal multiobjective optimization in designing a filter based feature selection technique. *Appl. Soft Comput.* **2021**, *98*, 106823. [\[CrossRef\]](#)
27. Kasihmuddin, M.S.M.; Jamaludin, S.Z.M.; Mansor, M.A.; Wahab, H.A.; Ghadzi, S.M.S. Supervised learning perspective in logic mining. *Mathematics* **2020**, *10*, 915. [\[CrossRef\]](#)
28. Yang, G. A modified particle swarm optimizer algorithm. In Proceedings of the 2007 8th International Conference on Electronic Measurement and Instruments, Warsaw, Poland, 1–3 May 2007; IEEE: Piscataway, NJ, USA, 2007; Volume 2, pp. 675–679.
29. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control. Eng. Open Access J.* **2020**, *8*, 22–34. [\[CrossRef\]](#)
30. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [\[CrossRef\]](#)
31. Mirjalili, S.; Mirjalili, S.M.; Yang, X.S. Binary bat algorithm. *Neural Comput. Appl.* **2014**, *25*, 663–681. [\[CrossRef\]](#)
32. Kumar, N.; Hussain, I.; Singh, B.; Panigrahi, B. Single sensor-based mppt of partially shaded pv system for battery charging by using cauchy and gaussian sine cosine optimization. *IEEE Trans. Energy Convers.* **2017**, *32*, 983–992. [\[CrossRef\]](#)
33. Gupta, S.; Deep, K. A hybrid self-adaptive sine cosine algorithm with opposition based learning. *Expert Syst. Appl.* **2019**, *119*, 210–230. [\[CrossRef\]](#)
34. Wsa, B.; Zq, B.; Aahc, D.; Hc, E.; Ht, F.; Yt, A. Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis. *Knowl.-Based Syst.* **2020**, *214*, 106728.
35. Jamaludin, S.Z.M.; Romli, N.A.; Kasihmuddin, M.S.M.; Baharum, A.; Mansor, M.A.; Marsani, M.F. Novel logic mining incorporating log linear approach. *J. King Saud-Univ. -Comput. Inf. Sci.* **2022**. [\[CrossRef\]](#)
36. Jamaludin, S.Z.M.; Kasihmuddin, M.S.M.; Ismail, A.I.M.; Mansor, M.A.; Basir, M.F.M. Energy based logic mining analysis with hopfield neural network for recruitment evaluation. *Entropy* **2020**, *23*, 40. [\[CrossRef\]](#)
37. Zamri, N.E.; Mansor, M.A.; Kasihmuddin, M.S.M.; Alway, A.; Jamaludin, S.Z. .M.; Alzaeemi, S.A. Amazon employees resources access data extraction via clonal selection algorithm and logic mining approach. *Entropy* **2020**, *22*, 596. [\[CrossRef\]](#)
38. Alway, A.; Zamri, N.E.; Kasihmuddin, M.S.M.; Mansor, A.; Sathasivam, S. Palm oil trend analysis via logic mining with discrete hopfield neural network. *Pertanika J. Sci. Technol.* **2020**, *28*, 967–981.
39. Zhou, W.; Wang, P.; Heidari, A.A.; Zhao, X.; Chen, H. Spiral gaussian mutation sine cosine algorithm: Framework and comprehensive performance optimization. *Expert Syst. Appl.* **2022**, *209*, 118372. [\[CrossRef\]](#)