Journal on Internet of Things
DOI: 10.32604/jiot.2021.014980
Article

New Solution Generation Strategy to Improve Brain Storm Optimization Algorithm for Classification

Yu Xue[1,2,*] and Yan Zhao[1]

[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China
[2]Jiangsu Key Laboratory of Data Science and Smart Software, Jinling Institute of Technology, Nanjing, China
[*]Corresponding Author: Yu Xue. Email: xueyu@nuist.edu.cn

Received: 25 May 2021; Accepted: 16 July 2021

Abstract: As a new intelligent optimization method, brain storm optimization (BSO) algorithm has been widely concerned for its advantages in solving classical optimization problems. Recently, an evolutionary classification optimization model based on BSO algorithm has been proposed, which proves its effectiveness in solving the classification problem. However, BSO algorithm also has defects. For example, large-scale datasets make the structure of the model complex, which affects its classification performance. In addition, in the process of optimization, the information of the dominant solution cannot be well preserved in BSO, which leads to its limitations in classification performance. Moreover, its generation strategy is inefficient in solving a variety of complex practical problems. Therefore, we briefly introduce the optimization model structure by feature selection. Besides, this paper retains the brainstorming process of BSO algorithm, and embeds the new generation strategy into BSO algorithm. Through the three generation methods of global optimal, local optimal and nearest neighbor, we can better retain the information of the dominant solution and improve the search efficiency. To verify the performance of the proposed generation strategy in solving the classification problem, twelve datasets are used in experiment. Experimental results show that the new generation strategy can improve the performance of BSO algorithm in solving classification problems.

Keywords: Brain storm optimization (BSO) algorithm; classification; generation strategy; evolutionary classification optimization

1 Introduction

Optimization problem is to solve the maximum (minimum) value of an objective function under certain constraints, which widely exists in scientific research and engineering applications. Data classification is a classic research topic in the field of machine learning, and its purpose is to extract a function or model from training samples to predict the label of unknown instances, which is similar to optimization problem. Some important classification methods such as support vector machine (SVM) [1], artificial neural network (ANN) [2], k-nearest neighbor (KNN) [3], decision tree (DT) [4], naive Bayesian classification (NBC) [5] have been proposed. These methods enrich the classification patterns and successfully solve various complex practical problems. However, many existing methods are fixed structures, and a considerable part of them are still fall into local optimal solutions.

Swarm intelligence evolutionary algorithm uses evolutionary method to iteratively to find the global most accurate solution, which cannot be calculated by traditional methods. The classic evolutionary algorithms are particle swarm optimization (PSO) [6], genetic algorithm (GA) [7], artificial bee colony (ABC) [8], brain storm optimization (BSO) [9] and ant colony optimization (ACO) [10], etc. Compared with traditional methods, swarm intelligence evolutionary algorithm has great advantages in algorithm performance and is widely used in solving practical problems. For example, Holden et al. [11] proposed a hybrid particle swarm optimization/ant colony optimization (PSO/ACO) algorithm to solve the classification problem. Cervantes et al. [12] introduced PSO algorithm into SVM classifier, which combining the best optimization technology and classification technology to achieve better classification performance. Dalal et al. [13] used genetic algorithm to optimize the kernel extreme learning machine, which has higher accuracy in classification of various forms of arrhythmia in Electrocardiogram. Therefore, the introduction of evolutionary algorithm into classification method can significantly improve the performance of classification method, but the improvement of evolutionary algorithm in classification method has limitations.

In the past few decades, evolutionary algorithms only improve the performance of classification by optimizing the parameters or structure of classifiers, and there are few other applications. Evolutionary classification optimization model is a new classification method proposed by Xue et al. [14], it uses BSO algorithm to solve the classification problem directly. This method verifies the effectiveness of evolutionary algorithm which can be directly used to solve the classification problem. However, Xue et al. [14] only focuses on how to construct the classification model, and the improvement of the classification performance of this method has not been involved too much. In addition, the experimental datasets used in [14] is still very limited, and the analysis of the method to solve the classification problem is not comprehensive enough. Therefore, the study of optimization evolutionary classification optimization model has a bright future.

The structure of evolutionary classification optimization model depends on the size of dataset dimension. Therefore, in large-scale datasets, complex structure may affect the performance of classification model. Feature selection is an effective way to solve such problems. For instance, Xue et al. [15] proposed an adaptive particle swarm optimization (SAPSO) algorithm for feature selection method, which achieves the ideal classification performance. Saravanan et al. [16] simplified feature subset obtained by feature selection is applied to monitor phishing websites, and its classification performance is superior to existing classifiers. Considering this, we introduce two feature selection methods to optimize the structure of the model.

BSO algorithm is a new evolutionary algorithm proposed by Shi [17]. It has the ability of local search and global search, and has better optimization ability in solving practical problems. However, BSO also has a series of problems; for example, there is no direct interaction between different solutions. Thus, when solving the optimal solution problem, the information of other better solutions is not used well. Moreover, although BSO has the ability of local search and global search, there are still some deficiencies in the balance between global exploration and local exploration. In view of the shortcomings of BSO algorithm, most of the existing optimization work of BSO algorithm is limited to modifying parameters to improve its performance. Such as, Li et al. [18] designed a vector grouping learning BSO (VGLBSO) methods. The method used VGL schemes to generate individuals for improve population diversity to coordinate global exploration and local exploitation capabilities. Li et al. [19] also proposed a BSO algorithm with multiple information interactions (MIIBSO) which using multiple information interaction (MII) strategies to enhance various information interactions between individuals; Moreover, the original K-means algorithm of BSO algorithm is replaced by random grouping (RG) strategy, which further enhances the ability of information exchange between individuals, and introduced dynamic differential step size (DDS) for improve the search range and coordinate global exploration and local exploitation capabilities. Yang et al. [20] proposed a new method of combining BSO with chaotic local search (CLS), which has randomness and periodicity. The diversification of the CLS search method breaks the premature convergence of the BSO algorithm and maintains the diversity of the population, thus achieving a better balance between global exploration and local exploitation. These show that there is little research on the improvement of BSO search strategy. Therefore, this paper uses three generation methods: global optimization, local optimization and nearest neighbor to better retain the information of the dominant solution to improve the search efficiency and the performance of the algorithm. The improved algorithm is applied to the evolutionary classification optimization model, and is tested on 12 experimental datasets, which from UCI Machine Learning Repository.

The rest of the paper is organized as follows: Section 2 describes the outline of the BSO algorithm. Section 3 focuses on three generation strategies. Section 4 introduces the evolutionary optimization classification model. Section 5 introduces the design of experimental parameters and Section 6 gives the experimental results and analysis. Section 7 summarizes the research and prospects the future trend.

2 Brain Strom Optimization Algorithm

The BSO algorithm was proposed in 2011 by Shi [21], and it simulates the collective behavior of human beings to solve the problem by brainstorming. In BSO, each individual in the algorithm represents the solution of a potential problem, and the individual is updated through the evolution and fusion of individuals.

The implementation of the BSO algorithm is very simple: Firstly, n (population size) solutions are generated for the problem to be solved, and then, these n individuals are divided into m (preset parameters) clusters by clustering algorithm. Through the evaluation of n individuals, the individuals in each cluster are sorted, and the optimal individual in each cluster is selected as the central individual of the cluster. The central individual of a cluster is selected randomly, and whether it is replaced by a randomly generated individual is determined according to the probability value P. In the process of individual update iteration, $P_{cluster}$ is used to control whether the new individual is generated by one individual or two individuals fusion. In addition, $P_{central}$ is used to control whether the new individuals is generated by central individuals or common individuals. The development area of the algorithm can be controlled by selecting individuals from one cluster to generate new individuals. In addition, when the selected individuals in multiple clusters are fused to generate new individuals, the new individuals are based on the information of multiple individuals, which can maintain the population diversity of the algorithm. New individuals are generated according to Eq. (1) and Eq. (2):

$$x_{new}^{i} = x_{old}^{i} + \xi(t) * N(\mu, \sigma^2) \tag{1}$$

$$\xi(t) = \log sig\left(\frac{0.5 * \max_{iteration} - t}{k}\right) * rand() \tag{2}$$

where x_{new}^{i} is the i^{th} new individual, and x_{old}^{i} represents the i^{th} individual to be updated; $N(\mu, \sigma^2)$ is a random number based on normal distribution and \$rand()\$ is a random function for generating random values; T and t represent the maximum number of iterations the current number of iterations, respectively; k is the coefficient that controls the $\log sig()$ function and it is used to change the search step-size of $\xi(t)$ to balance the convergence speed of the algorithm. The definition of transfer function $\log sig()$ is shown in Eq. (3):

$$\log sig(\alpha) = \frac{1}{1 + \exp(-\alpha)} \tag{3}$$

Finally, the new individuals are compared with the old ones in the current position, leaving the individuals with better fitness values.

The BSO algorithm framework is described in Algorithm 1:

Algorithm 1 The framework of BSO algorithm

Input: Parameters including the population size n, the number of clusters m, and the maximum number of iterations $max\_iteration$

Output: The best individual and fitness value.

```
initialize: Randomly initialize n individuals and evaluate the fitness value of each individual;
Disrupt cluster centers: Determine whether to replace a central individual randomly;
while stop criterion == false do
Clustering: Cluster n individuals into m clusters by k-means algorithm;
New individuals' generation: Randomly select an individual or combine two individuals to generate n new individuals by Eq.(1);
Selection: Compare the newly generated individual and the i^th individual, and select the superior individual to enter the next iteration;
end while
```

3 New Solution Generation Strategy to Improve Brain Strom Optimization Algorithm (IBSO)

To improve the performance of BSO algorithm, researchers have proposed different improvement strategies. These improvements can be divided into three categories: individuals clustering, new individual generation and hybrid algorithm. Among them, the new individual generation strategy greatly affects the efficiency of BSO. By improving the new individual generation strategy, the search efficiency of BSO algorithm can also be improved. In order to solve different types of optimization problems, the algorithm needs to use the real-time search information adaptively. In the aspect of new individual generation, there are also a lot of research work that can be studied, such as modifying search step size and new individual generation method. In this paper, we use three new individual generation strategies, which have achieved good performance in previous applications.

1) Cluster optimal strategy: the central individual of clusters guide the generation of new individuals, where $x_{central}^{icluster}$ is the central individual in the cluster where the i^{th} individual is located.

$$x_{new}^{i} = x_{old}^{i} + (x_{central}^{icluster} - x_{old}^{i}) * N(\mu, \sigma^2) \tag{4}$$

2) Nearest neighbor strategy: neighboring individuals guide the generation of new individuals, where x_{near} is the nearest individual to the i^{th} individual found by the nearest neighbor algorithm.

$$x_{new}^{i} = x_{old}^{i} + (x_{near} - x_{old}^{i}) * N(\mu, \sigma^2) \tag{5}$$

3) Global optimal strategy: global optimal individuals guide the generation of new individuals, where $x_{optimal}$ is the global optimal individual obtained by comparing the cluster central individuals.

$$x_{new}^{i} = x_{old}^{i} + (x_{optimal} - x_{old}^{i}) * N(\mu, \sigma^2) \tag{6}$$

There is a probability value $P_{cluster}$ to decide whether the new individual is generated by the global optimal strategy or the others. If the new individual is not generated by the global optimal strategy, roulette is used to determine which cluster the new individual is generated from. In addition, the probability value P_{one} used to select the cluster optimal strategy or the nearest neighbor strategy to generate new individuals.

The new individual generation strategy is described in Algorithm 2.

Algorithm 2 The new individual generation strategy

```
Input: Clustered population and central individuals
for i < n do
    if rand() < P_cluster then
        if rand() < P_one then
            Select the cluster optimal strategy to generate new individuals by Eq.(4);
        else
            Select the nearest neighbor strategy to generate new individuals by Eq.(5);
        end if
    else
        Select the global optimal strategy to generate new individuals by Eq.(6);
    end if
    Calculate the fitness of the new individual
    if x_i^new is better than x_i then
        Replace x_i with x_i^new;
        i = i + 1;
    end if
end for
```


4 Evolutionary Classification Optimization Model

Given a dataset D, in which 70% of the samples are used as training sets T, write as:

$$\begin{bmatrix} x_{11}, x_{12}, \cdots, x_{1d}, y_1 \\ x_{21}, x_{22}, \cdots, x_{2d}, y_2 \\ \cdots \quad \cdots \quad \cdots \\ x_{m1}, x_{m2}, \cdots, x_{md}, y_m \end{bmatrix} \tag{7}$$

where (x_i, y_i) is the i^{th} samples, $x_i = x_{i1}, x_{i2}, \ldots, x_{id}$ is the information of the i^{th} sample, and y_i represents the label of the i^{th} sample.

Then, the weight vector $W = \{w_1, w_2, \cdots, w_d\}$ is introduced to transform the classification problem into the following optimization problem for solving linear equations:

$$\begin{cases} w_1 x_{11} + w_2 x_{12} + \cdots + w_d x_{1d} = y_1 \\ w_1 x_{21} + w_2 x_{22} + \cdots + w_d x_{2d} = y_2 \\ \qquad \cdots \\ w_1 x_{m1} + w_2 x_{m2} + \cdots + w_d x_{md} = y_m \end{cases} \tag{8}$$

In practical problems, it is usually difficult to find the exact solutions of the linear equations. Fortunately, for classification problem, we only need to search the approximate solutions of the linear equations:

$$\begin{cases} w_1 x_{11} + w_2 x_{12} + \cdots + w_d x_{1d} \approx y_1 \\ w_1 x_{21} + w_2 x_{22} + \cdots + w_d x_{2d} \approx y_2 \\ \qquad \cdots \\ w_1 x_{m1} + w_2 x_{m2} + \cdots + w_d x_{md} \approx y_m \end{cases} \tag{9}$$

When we find such an approximate solution W, we can predict the label of the sample by the information of this sample. The process is as follows, and the δ an acceptable error range.

$$\begin{cases} y_1 - \delta \le w_1 x_{11} + w_2 x_{12} + \cdots + w_d x_{1d} \le y_1 + \delta \\ y_2 - \delta \le w_1 x_{21} + w_2 x_{22} + \cdots + w_d x_{2d} \le y_2 + \delta \\ \qquad\qquad\qquad \cdots \\ y_m - \delta \le w_1 x_{m1} + w_2 x_{m2} + \cdots + w_d x_{md} \le y_m + \delta \end{cases} \tag{10}$$

In this paper, the evolutionary classification optimization model is implemented by BSO algorithm, which is described as classification based on BSO (CBSO).

5 Evolutionary Classification Optimization Model with Feature Selection

Feature selection is an effective way of data preprocessing, and is often used to solve classification problems. In this paper, feature selection is applied to this model in two ways.

(1) Taking the feature subset and weight parameters as a whole, and BSO algorithm is used to search for the optimal feature subset and weight parameters simultaneously. We define 1 represents that the corresponding feature is selected, otherwise, 0 represents that the corresponding feature is not selected. For example: in the Fig. 1, the orange part represents the feature subset part, and the green part represents the weight parameter part. Four random numbers between 0 and 1 correspond to four features and compared with 0.6, it means that the first features are not selected and the other two feature are selected.

Figure 1: Representation of individuals

In this case, the feature subset and weight parameters are regarded as an individual and BSO is used to find the optimal individual. This method is defined as FS-CBSO.

(2) The feature subset and weight parameters are regarded as two parts. BSO algorithm is used to search the optimal feature subset on the one hand, and on the other hand, it is used to search the optimal weight parameter under the corresponding feature subset. Firstly, BSO algorithm is used to generate the population of feature subsets. Different feature subsets represent different structures of the model. Secondly, under different structures, BSO algorithm is used to search the optimal weight parameters of each structure. In addition, each structure learns from each other and retains the optimal feature subset and its corresponding weight parameters. In this case, feature subset and weight parameter are considered as two parts to search separately, and the search of weight parameter depends on feature subset. This method is defined as CBSO-FS.

The process of the two methods is shown in Fig. 2 and Fig. 3.

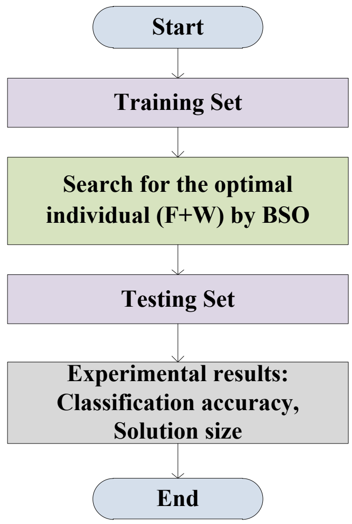

Figure 2: The process of FS-CBSO

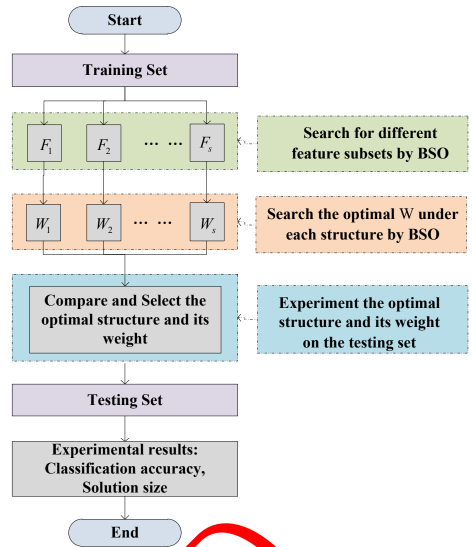

Figure 3: The process of CBSO-FS

6 Experimental Design

6.1 Datasets

This section selects 11 different datasets from the UCI Machine Learning Repository to test the classification performance of the improved BSO (IBSO). The number of instances, features and classes of the dataset are shown in Table 1 70% instances of these dataset are randomly selected as the training datasets to learn the classification model, and the other 30% are used to test the prediction accuracy of the classification model after modeling.

Table 1: Information of datasets

ID	Datasets	NoE	NoF	NoC
DS1	bcw	683	9	2
DS2	wbcd	569	30	2
DS3	ione2	351	33	2
DS4	biodeg	1055	41	2
DS5	spect	267	44	2
DS6	sonar	208	60	2
DS7	MicroMass	360	1300	2
DS8	madelon	600	500	2
DS9	Musk1	476	66	2
DS10	ConnectionistBenchData	208	60	2
DS11	hill	606	100	2

6.2 Parameter Settings

The objective function used in this paper is $\min(f(W)=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{k}(w_j * x_{ij} - y_i^2)})$, all methods run independently for 30 times, and the maximum number of function evaluations is 100,000. In addition, some other parameters are set to: the population size is \$100\$, the number of clusters is 2, $P_{cluster}=0.8$, $P_{one}=0.4$.

7 Experiment Results and Analysis

In order to prove the superiority of the new solution generation strategy, this section will compare with the original algorithm, and test the classification accuracy and solution size based on the evolutionary classification optimization model. The experimental results of CBSO, CBSO-FS, FS-CBSO, IBSO, IBSO-FS and FS-IBSO are shown in Table 2 and Table 3. "mean" and "std" represent the average and standard deviation of classification accuracy (solution size) after the algorithm runs independently on each dataset for 30 times, respectively. Among them, on the same dataset, the results with the most average and standard deviation are represented in bold.

Table 2: Classification accuracy of the six methods on eleven testing sets

Datasets	CBSO		CBSO-FS		FS-CBSO		IBSO		IBSO-FS		FS-IBSO	
	mean	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**
DS1	**0.9804**	0	0.8941	0.0402	0.8724	0.0366	**0.9804**	0	0.9279	0.0374	0.8601	0.0443
DS2	0.5314	0.0998	0.8435	0.0503	0.8341	0.0362	0.5424	0.1121	**0.8635**	0.1335	0.8443	0.0386
DS3	0.9343	0.0091	0.9689	0.0236	**0.9692**	0.0145	0.6295	0.1582	0.9508	0.0567	0.9552	0.0304
DS4	0.5242	0.0596	0.6263	0.1852	0.7572	0.0501	0.5450	0.0491	[illegible]	0.2017	**0.7662**	0.0101
DS5	0.4954	0.0680	0.4838	0.0501	0.4913	0.0490	**0.5250**	[illegible]	0.4967	0.0568	0.5096	0.0533
DS6	0.3995	0.1306	0.6758	0.1857	0.7231	0.1404	[illegible]	0.2390	0.6928	0.2590	**0.7651**	0.2439
DS7	0.5474	0.0411	0.6877	0.0077	0.6832	0.0107	0.4863	0.0530	**0.6899**	0.0161	0.6767	0.0113
DS8	0.4799	0.0576	0.4827	0.0358	0.4932	[illegible]	0.4972	0.0660	**0.5146**	0.0398	0.5002	0.0340
DS9	0.5272	0.0823	0.5580	0.0781	**0.5645**	0.0769	0.4920	0.0816	0.5385	0.0909	0.5290	0.0619
DS10	0.5013	0.0660	0.4517	0.4271	[illegible]	0.3827	0.5361	0.1239	0.4976	0.4031	**0.6133**	0.4232
DS11	0.4854	0.0649	0.4697	0.1561	[illegible]	0.1431	0.5115	0.1068	0.5120	0.1105	**0.5463**	0.1529

Table 3: Solution size of the six methods on eleven testing sets

Datasets	CBSO		CBSO-FS		FS-CBSO		IBSO		IBSO-FS		FS-IBSO	
	mean	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**	**mean**	**std**
DS1	9	NA	4.47	0.90	1.90	0.48	9	NA	4.30	1.02	**1.50**	0.51
DS2	30	NA	[illegible]	2.06	7.03	1.90	30	NA	8.67	2.15	**4.60**	1.30
DS3	33	NA	15.13	1.93	8.20	2.16	33	NA	12.07	2.38	**3.10**	0.92
DS4	41	NA	12.53	2.36	4.57	1.76	41	NA	11.07	2.03	**1.77**	0.682
DS5	44	NA	13.27	1.89	7.40	3.07	44	NA	12.80	2.17	**5.90**	3.29
DS6	60	NA	22.57	3.17	16.17	4.09	60	NA	17.93	2.48	**9.10**	2.77
DS7	24	NA	7.77	1.43	3.40	0.93	24	NA	7.57	1.36	**2.33**	0.84
DS8	100	NA	37.97	4.52	36.73	4.27	100	NA	35.47	5.01	**34.33**	4.21
DS9	1300	NA	483.63	40.53	**320.03**	40.16	1300	NA	533.27	29.59	544.03	16.51
DS10	500	NA	192.37	18.50	160	23.04	500	NA	**96.97**	10.32	196.53	12.69
DS11	166	NA	59.40	5.80	**57.83**	5.93	166	NA	61.90	4.55	59.83	7.97

Table 2 shows the classification accuracy of each algorithm on the test set, and Table 3 shows the solution size of each algorithm on the test set. We will analyze the experimental results from classification accuracy and solution size.

Classification accuracy: As can be seen from Table 2, IBSO achieves the highest classification accuracy on DS1 and DS5. IBSO-FS achieves the highest classification accuracy on DS2, DS7 and DS8. FS-IBSO achieves the highest classification accuracy on four datasets: DS4, DS6, DS10 and DS11. Both DS3 and DS9 achieved the highest classification accuracy on FS-CBSO. In general, the algorithm with new solution generation strategy achieves the best classification accuracy on eight datasets, and is inferior to the original algorithm only in two datasets.

Solution size: Because CBSO and IBSO do not use feature selection technology, it is not applicable (NA) in this section. It can be seen from Tab.3 that FS-IBSO has the minimum solution size on eight datasets, IBSO-FS has the minimum solution size on one dataset, and FS-CBSO has the minimum solution size on two datasets. In other words, the method with new solution generation strategy has better solution size on most datasets.

On the whole, the method with new solution generation strategy outperforms the original algorithm in classification accuracy and solution size. Therefore, it can be concluded that the new solution generation strategy to improve BSO algorithm for classification problem is feasible.

8 Conclusions and Future Work

In this paper, we propose the new solution generation strategy method to improve BSO algorithm. It introduces three new solution generation strategies into BSO to improve the classification performance of evolutionary classification optimization model. Experiments on 11 different datasets show that the method with new solution generation strategy has better classification performance and solution size. This means that the new solution generation strategy to improve the BSO algorithm for solving classification problems is effective.

In this paper, we have done a lot of research and optimization on the modeling of classification problem and the improvement of BSO algorithm. However, there are still some problems and need to be further explored. The experimental datasets selected in this paper have only two class labels, but the research on the problem of multiple class labels has not been carried out. In addition, the classification performance has not reached our ideal result; we need to further explore better algorithm improvement methods.

Funding Statement: This work was partially supported by the National Natural Science Foundation of China (61876089, 61403206, 61876185, 61902281), the opening Project of Jiangsu Key Laboratory of Data Science and Smart Software (No. 2019DS302), the Natural Science Foundation of Jiangsu Province (BK20141005), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (14KJB520025), the Engineering Research Center of Digital Forensics, Ministry of Education, and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. Soumaya, B. D. Taoufiq, N. Benayad, K. Yunus and A. Abdelkrim, "The detection of Parkinson disease using the genetic algorithm and SVM classifier," *Applied Acoustics*, vol. 171, 107528, 2021.

[2] M. J. Huang, Y. L. Tsou and S. C. Lee, "Integrating fuzzy data mining and fuzzy artificial neural networks for discovering implicit knowledge," *Knowledge Based Systems*, vol. 19, pp. 396–403, 2006.

[3] J. E. Goin, Y. L. Tsou and S. C. Lee, "Classification bias of the k-nearest neighbor algorithm," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 6, pp. 379–381, 2009.

[4] R. C. Barros, D. D. Ruiz and M. P. Basgalupp, "Evolutionary model trees for handling continuous classes in machine learning," *Information Sciences*, vol. 181, pp. 954–971, 2011.

[5] X. Z. Wang, Y. L. He and D. D. Wang, "Non-naive bayesian classifiers for classification problems with continuous attributes," *IEEE Transactions on Cybernetics*, vol. 44, pp. 21–39, 2013.

[6] T. T. Yan, L. Zhang, N. S. Chin and L. C. Peng, "Intelligent skin cancer detection using enhanced particle swarm optimization," *Knowledge Based Systems*, vol. 158, pp. 118–135, 2018.

[7] Y. W. C. Chien and Y. L. Chen, "Mining associative classification rules with stock trading data–A GA-based method," *Knowledge Based Systems*, vol. 23, pp. 605–614, 2010.

[8] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, pp. 459–471, 2007.

[9] H. Zhu and Y. Shi, "Brain storm optimization algorithms with k-medians clustering algorithms," in *7th Int. Conf. on Advanced Computational Intelligence*, vol. 11, pp. 107–110, 2015.

[10] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation,* vol. 1, pp. 53–66, 1997.

[11] N. Holden and A. A. Freitas, "A hybrid PSO/ACO algorithm for discovering classification rules in data mining," *Journal of Artificial Evolution & Applications*, vol. 2, pp. 1–11, 2008.

[12] J. Cervantes, F. G. Lamont, L. C. Asdrubal, L. Rodriguez, J. S. R. Castilla *et al.*, "PSO-based method for SVM classification on skewed data-sets," in *Int. Conf. on Intelligent Computing*, vol. 9227, pp. 79–86, 2015.

[13] S. Dalal and V. P. Vishwakarma, "GA based KELM optimization for ECG classification," *Procardia Computer*, vol. 167, pp. 580–588, 2020.

[14] Y. Xue, T. Tang and T. Ma, "Classification based on brain storm optimization algorithm," *Bio-Inspired Computing-Theories and Applications,* vol. 681, pp. 371–376, 2016.

[15] Y. Xue, B. Xue and M. Zhang, "Self-adaptive particle swarm optimization for large-scale feature selection in classification," *ACM Transactions on Knowledge Discovery from Data,* vol. 13, pp. 1–50, 2019.

[16] P. Saravanan and S. Subramanian, "A framework for detecting phishing websites using GA based feature selection and ARTMAP based website classification," *Procardia Computer,* vol. 171, pp. 1083–1092, 2020.

[17] Y. Shi, "Brain storm optimization algorithm," in *IEEE Cong. on Evolutionary Computation*, vol. 15, pp. 1227–1234, 2011.

[18] C. Li, D. Hu, Z. Song, F. Yang, Z. Luo *et al.,* "A vector grouping learning brain storm optimization algorithm for global optimization problems," *IEEE Access*, vol. 6, pp. 78193–78213, 2018.

[19] C. Li, Z. Song, J. Fan, Q. Cheng and P. X. Liu, "A brain storm optimization with multi-information interactions for global optimization problems," *IEEE Access*, vol. 6, pp. 19304–19323, 2018.

[20] Y. Yu, S. Gao, S. Cheng, Y. Wang, S. Song *et al.,* "CBSO: A mimetic brain storm optimization with chaotic local search," *Mimetic Computing*, vol. 10, pp. 353–367, 2017.

[21] Y. Shi, "An optimization algorithm based on brainstorming process," *International Journal of Swarm Intelligence Research*, vol. 2, pp. 35–62, 2011.