Classification Based on Brain Storm Optimization With Feature Selection

YU XUE1,2, (Member, IEEE), YAN ZHAO1, AND ADAM SLOWIK3, (Senior Member, IEEE)

1School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
2Jiangsu Key Laboratory of Data Science and Smart Software, Jinling Institute of Technology, Nanjing 210044, China
3Department of Electronics and Computer Science, Koszalin University of Technology, 75-453 Koszalin, Poland

Corresponding author: Yu Xue (xueyu@nuist.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61876089, Grant 61403206, Grant 61876185, and Grant 61902281; in part by the opening Project of Jiangsu Key Laboratory of Data Science and Smart Software under Grant 2019DS301; in part by the Natural Science Foundation of Jiangsu Province under BK20141005; in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant 14KJB520025; in part by the Engineering Research Center of Digital Forensics, Ministry of Education; in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions; and in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant KYCX20_0981 and Grant KYCX20_0979.

ABSTRACT Classification is one of the most classic problems in machine learning. Due to the global optimization ability, evolutionary computation (EC) techniques have been successfully applied to solve many problems and the evolutionary classification model is one of the methods used to solve classification problems. Recently, some evolutionary algorithms (EAs) such as the fireworks algorithm (FWA) and brain storm optimization (BSO) algorithm have been employed to implement the evolutionary classification model and achieved the desired results. This means that it is feasible to use EC techniques to solve the classification problem directly. However, the existing evolutionary classification model still has some disadvantages. The limited datasets used in the experiment make the experimental results not convincing enough, and more importantly, the structure of the evolutionary classification model is closely related to the dimension of datasets, which may lead to poor classification performance, especially on large-scale datasets. Therefore, this paper aims at improving the structure of the evolutionary classification model to improve classification performance. Feature selection is an effective method to deal with large datasets, firstly, we introduce the concept of feature selection and use the different feature subsets to construct the structure of the evolutionary classification model. Then, the BSO algorithm is employed to implement the evolutionary classification model to search the optimal structure by search for the optimal feature subset. Moreover, the optimal weight parameters corresponding to the different structures are also searched by the BSO algorithm while searching the optimal feature subset. For verification of the classification effectiveness of the proposed method, 11 different datasets are selected for experiments. The results show that it is feasible to optimize the structure of the evolutionary classification model by introducing feature selection. Moreover, the new method has better classification performance than the original method, especially on large-scale or high-dimensional datasets.

INDEX TERMS Brain storm optimization (BSO) algorithm, classification, evolutionary computation (EC), feature selection.

I. INTRODUCTION

Classification problem has been widely studied in machine learning [1]. Many different classification methods have been proposed and been widely used in practical applications. For instance, k-nearest neighbor (KNN) [2], naive bayesian classification (NBC) [3], support vector machine (SVM) [4], decision tree (DT) [5], artificial neural network (ANN) [6], etc. However, many of these methods are structurally deterministic, which makes them often fall into a locally optimal solution.

Evolutionary computation (EC) techniques, such as ant colony optimization (ACO) [7], particle swarm optimization (PSO) [8], genetic algorithm (GA) [9], brain storm...
optimization (BSO) [10], differential evolution algorithm (DE) [11], fireworks algorithm (FWA) [12], artificial bee colony (ABC) [13], extremal optimization (EO) [14], etc., are well-known for their global optimization capabilities and have been concerned and improved by researchers. For example, according to the threshold of some parameters, El-Abd [15] introduced a new population initialization scheme and a global-best version, which combined with fitness-based grouping and per-variable updates to improve the BSO algorithm performance. The proposed GBSO method achieves ideal results on different problem scales and extensive classical functions. An optimization problem modeling for Loney’s solenoid problem was successfully solved by Duan and Li [16] proposed a quantum-behaved BSO (QBSO) algorithm. In addition, Liang et al. [17] combined the advantages of ACO and BSO, and proposed a new feature selection algorithm, which has achieved good results in accuracy, percent rate, and recall rate, etc. Thus, considering the advantages of EC techniques in solving various problems, it may also be a better way to solve the classification problem.

In recent years, many researchers have introduced EC techniques to solve the classification problem. Si and Dutta [18] proposed a method based on PSO algorithm and Feed-Forward Neural Network (FFNN) for classification task in medical machine learning. Nugraha et al. [19] proposed a PSO-SVM algorithm for journal rank classification, and the research shows that PSO can improve the classification performance of SVM. Besides, Rajathi and Radhamani [20] combined the KNN algorithm with ACO to identify the presence of heart disease, and their method has achieved good performance in accuracy and error rate. However, all of the above methods use EC techniques to optimize a classifier, and do not attempt to solve the classification problem directly with EC techniques. This greatly affects the global optimization ability of EC techniques, which may not be fully reflected.

Xue et al. proposed an evolutionary classification model that can solve the classification problem by the BSO algorithm [21] and FWA [22]. In the proposed model, linear equations are constructed based on the training set, and the objective function is constructed based on the equations. However, neither [21] nor [22] consider the relationship between the complexity of the model structure and the size of the dataset. Besides, there are not enough datasets were selected for the experiment in [21], [22], especially on large-scale datasets. Therefore, considering optimize the structure of the evolutionary classification model is a key point to improve its classification performance.

Feature selection is a classical data preprocessing technique, which is often used to solve the problem with high-dimensional or large-scale datasets. Selecting a good feature subset can not only improve the classification accuracy, but also simplify classification structure. Therefore, the concept of feature selection has been introduced into more and more classification problems in recent years. Due to good global optimization ability, EC techniques are often achieved good results in finding the optimal feature subset. For instance, Xue et al. [23] proposed a new initialization strategy and a new update mechanism of individual optimization and global optimization based on the traditional PSO algorithm to solve the feature selection problem. Their experiments are carried out from two aspects: 1) maximizing the classification performance; 2) considering the number of features and classification performance at the same time. Compared with the algorithm only considering the classification accuracy, the algorithm considering both the feature number and the classification accuracy shows better classification performance in the experiment. In addition, Tajik et al. [24] used GA in feature selection to improve the efficiency of computer-aided diagnosis. The research also shows that the combination of feature number and classification accuracy has good performance in classification. Inspired by this, we introduce the feature selection technique into the evolutionary classification model to optimize the structure of the model and improve the classification performance.

In this paper, feature selection is introduced to improve the structure of the evolutionary classification model, which is implemented by the BSO algorithm. Firstly, the structure of the model depends on the dimension of the dataset, so the BSO algorithm is used to initialize many different feature subsets to construct different structures of the model. In addition, the BSO algorithm is also used to initialize weight parameters corresponding to the different structures. Then, taking the error values as the objective function, the BSO algorithm is used to search for the optimal structure and its weight parameters. Finally, solution size (the number of selected features) and classification accuracy are used as the final evaluation criteria to evaluate the performance of the model. Different from the previous evaluation criteria, we not only compare the classification accuracy, but also evaluate the solution size. Therefore, the new evolutionary optimization classification model can significantly reduce the computational complexity. In addition, for verification of the effectiveness of the proposed method, 11 different datasets from the University of California Irvine (UCI) Machine Learning Repository [25] are used for experiments. The following is the arrangement of the rest of this article. Section II summarizes the BSO algorithm and the evolutionary classification model. Section III introduces our method and describes the evolutionary classification model with feature selection based on the BSO algorithm in detail. Section IV gives the information of the datasets used in the experimental and the parameter values used in the algorithm. Then, the experimental results are given and analyzed by Section V. Finally, Section VI concludes the paper and insight the future research directions.

II. RELATIVE WORK

A. EVOLUTIONARY CLASSIFICATION MODEL

In [21], [22], a basic evolutionary classification model was proposed as follows:
Given a training set $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where $x_i = x_{i1}, x_{i2}, \ldots, x_{id}$ represents the $i^{th}$ instance and the class label of the $i^{th}$ instance is described as $y_i$, $d$ is the dimension of the training set and $n$ represents the number of the instances, write it as follows:

$$
\begin{bmatrix}
  x_{i1}, & x_{i2}, & \cdots, & x_{id}, & y_i \\
  x_{21}, & x_{22}, & \cdots, & x_{2d}, & y_2 \\
  \vdots, & \vdots, & \cdots, & \vdots, & \vdots \\
  x_{n1}, & x_{n2}, & \cdots, & x_{nd}, & y_n \\
\end{bmatrix}
$$

(1)

First of all, the classification problem can be regarded as an optimization problem for solving linear equations by introducing weight vector $W = \{w_1, w_2, \ldots, w_d\}$, such as Eq.(2):

$$
\begin{align*}
  w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id} &= y_i \\
  w_1 x_{21} + w_2 x_{22} + \cdots + w_d x_{2d} &= y_2 \\
  \vdots \\
  w_1 x_{n1} + w_2 x_{n2} + \cdots + w_d x_{nd} &= y_n \\
\end{align*}
$$

(2)

where $W$ is the solution set that can satisfy the equation set.

Let the coefficient matrix of the linear equation be $X$ and the constant column vector be $Y$, as defined below:

$$
X = \begin{bmatrix}
  x_{i1}, & x_{i2}, & \cdots, & x_{id} \\
  x_{21}, & x_{22}, & \cdots, & x_{2d} \\
  \vdots, & \vdots, & \cdots, & \vdots \\
  x_{n1}, & x_{n2}, & \cdots, & x_{nd} \\
\end{bmatrix}	ext{ and } Y = \begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_n \\
\end{bmatrix}
$$

(3)

Therefore, Eq.(2) can also be written as $X \cdot W^T = Y$ [26].

The rank of matrix $X$ is $R(X)$ and the rank of a matrix $(X, Y)$ is $R(X, Y)$. In the majority of cases, the number of instances is far larger than $d$, and these equations are uncorrelated. Thus, it is greatly possible for these equations to be uncorrelated and have no solution which can satisfy all the equations. Fortunately, solving the classification problem, we only need to find the approximate solutions for the following equation:

$$
\begin{align*}
  w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id} &\approx y_i \\
  w_1 x_{21} + w_2 x_{22} + \cdots + w_d x_{2d} &\approx y_2 \\
  \vdots \\
  w_1 x_{n1} + w_2 x_{n2} + \cdots + w_d x_{nd} &\approx y_n \\
\end{align*}
$$

(4)

With this set of approximate solutions, we can calculate the class label of instance $x_i$, described as:

$$
\begin{align*}
  y_1 - \delta &\leq w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id} \leq y_1 + \delta \\
  y_2 - \delta &\leq w_1 x_{21} + w_2 x_{22} + \cdots + w_d x_{2d} \leq y_2 + \delta \\
  \vdots \\
  y_n - \delta &\leq w_1 x_{n1} + w_2 x_{n2} + \cdots + w_d x_{nd} \leq y_n + \delta \\
\end{align*}
$$

(5)

where $\delta$ is a threshold, which means that the error of the calculated label is allowed to be within this range. In addition, the objective functions used in the model are defined as follows:

$$
\begin{align*}
\text{min}(f(W)) = & \sum_{i=1}^{d} \sum_{j=1}^{n} (w_j \ast x_{ij} - y_j^2) \\
\end{align*}
$$

(6)

### B. BRAIN STORM OPTIMIZATION ALGORITHM

In recent years, many EAs have emerged and the BSO algorithm is one of the popular algorithms, which proposed by Professor Shi [27] in 2011. BSO algorithm uses the clustering idea to search the local optimal value, and obtains the global optimal value by comparing all the local optimal values, which is suitable for solving the multi-peak high-dimensional function problem, so the BSO algorithm is applied to the evolutionary classification model.

In BSO, the individuals in the algorithm represent the solutions of the potential problem, and the solutions are converged to several clusters. New individuals are updated through the evolution and fusion of individuals in the clusters. The new individuals are compared with the original individuals, and the individuals with better fitness values are stored and updated iteratively. The process of BSO is as follows:

- **Step 1 Initialization**: $N$ initial individuals (initial solutions) are generated randomly, and the individuals’ fitness values are calculated.
- **Step 2 Regeneration population**: If a good enough solution is not found or the preset maximum number of iterations is not reached, repeat the following steps:
  1. Solution clustering: $N$ individuals are clustered into $m$ clusters by $k$-means algorithm;
  2. Generating new solutions: New individuals are generated from randomly selected individuals from the clusters;
  3. New solution selection: Compare the fitness value of the new solution and the original solution, and store the solution with better fitness value in the next iteration.
- **Step 3 Optimal individual**: The global optimal individuals are obtained by comparing the central individuals in $m$ clusters.

Figure 1 depicts the execution flow chart of the BSO algorithm.

### C. FEATURE SELECTION

The number of instances and features determines the size of the dataset. In the era of big data, it has become an inevitable trend to deal with large-scale or high-dimensional datasets in solving classification problems. However, they often make the classifier more complex, which resulting in the performance degradation. In practice, many features of the dataset are redundant and irrelevant. They not only increase the training time and computational complexity, but also affect the classification performance, which is called the “curse of dimensionality”. Feature selection is an effective
way to solve such problems, which can reduce the exponential
increase of training time and computational complexity due
to the increase of dimension.

In a $d$-dimensional dataset, it means that there are
$2^d - 1$ feature subsets existing. This means that datasets with
smaller dimensions have fewer feature subsets. Therefore,
we can use the enumeration method to found the optimal
feature subset. However, if the $d$ is large, it is impossible to
enumerate all feature subsets. At this time, how to identify
the effective features from the original dataset is a difficult
problem. Therefore, it is necessary to find a method to obtain
an effective feature subset efficiently.

The basic feature selection process must determine the four
aspects shown in Figure 2:

1) Generating process: Generating candidate feature
subset;
2) Evaluation criteria: Evaluation of the characteristics of
the subset;
3) Stop criterion: Decide when to stop;
4) Verification results: Verify that the expected results are
achieved.

**III. PROPOSED METHOD**

In the evolutionary classification model with feature selec-
tion based on the BSO algorithm, the structure size of the
evolutionary optimization classification model depends on
the dimension of the training set, and the optimization of the
model structure needs to exclude the irrelevant and redundant
features of the training set. We use the evolutionary mecha-
nism of the BSO algorithm to generate a new structure and
its corresponding weight parameters, which mainly includes
the following two aspects and three steps: 1) feature subset
optimization and weight parameters optimization; 2) initial-
ization population, generation of new individuals, selection
of individuals.

Firstly, the feature subset population and weight parameter
population are initialized respectively.

The purpose of feature selection is to find out the most
effective features from all features. It means that all features
have two possibilities: selected and unselected. In order to
represent the feature subset conveniently, we define that 1
represents the feature selected, and 0 represents the feature is
not selected. In this paper, $N$ groups of $d$ random numbers
in the range of $[0, 1]$ are randomly generated by the BSO
algorithm, which corresponding to $d$ features. If the random
number is $\geq 0.6$, it means that its corresponding feature is
selected. Otherwise, if the random number is $< 0.6$, the cor-
responding feature is unselected. Thus, the feature subset
can be represented as binary coding, for example: in the
Figure 3, five random numbers between 0 and 1 correspond
to five features. Among them, 0.33, 0.42, and 0.17 were
all less than the threshold value 0.6, which represents that
the first feature, the third feature and the fourth feature are
not selected. In addition, 0.65 and 0.94 are greater than the
threshold value 0.6, which means that the second feature, and
the fifth feature are selected.

![FIGURE 3. Binary coding representation of feature subset.](image-url)
but how to determine the search space well also is our next research direction.

Then, we combine the feature subset population with the weighted parameters population. The fitness value of each individual was calculated. BSO algorithm takes the minimum error value \(\min(f(W))\) as the optimization objective function, and the fitness calculation method is Eq.(8).

\[
f(W) = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{d} (w_j * x_{ij} - y_i^2)}
\]

where \(n\) is the number of samples in the training set.

Secondly, \(N\) individuals are clustered into \(m\) by the \(k\)-means algorithm, and the individuals with the optimal fitness value in each cluster are selected as the central individuals. Finally, the optimal individuals are obtained by comparing the central individuals of each cluster. If the current optimal individuals are not satisfied, we generate new individuals from the existing population of individuals.

Through clustering operation, the search area can be improved. However, after a large number of iterations, all solutions converge to a small search area with high probability. In order to avoid being trapped in local optimum or premature convergence, a disrupt cluster operation is introduced into the BSO algorithm. The disrupt cluster operation is controlled by a probability value \(P\). Generate a random number in the range of \((0, 1)\), and if this random number is greater than \(P\), a random individual (including feature subset and weight parameters) is generated to replace a central individual randomly. Otherwise, new individuals will be generated directly.

The algorithm randomly selects clusters to generate new individuals, and new individuals can be generated based on one or more existing individuals. In this paper, if the probability value \(P_{\text{cluster}}\) is greater than 0.8, and the new individuals are generated based on one existing individual. Otherwise, the new individuals are generated based on a combination of more existing individuals. If new individuals are generated in one cluster, a search area can be optimized, and the algorithm focuses on development capability. Accordingly, if new individuals are generated based on more clusters, the new individuals generated may be far away from the original cluster center. In this case, the algorithm focuses more on the exploration ability.

The probability values \(P_{\text{one}}\) and \(P_{\text{two}}\) are used to control the selection of central individual or non-central random individuals in one or more clusters, respectively, to generate new individuals. For one cluster is selected, a random value is generated and if the random value is greater than the probability values \(P_{\text{one}}\), we choose the central individual in this cluster to generate new individuals. Otherwise, a non-central random individual in this cluster is selected to generate new individuals. At this time, the generation of new individuals are described by Eq.(9) and Eq.(10):

\[
x_{\text{new}}^i = x_{\text{selected}}^i + \xi(t) \cdot N(\mu, \sigma^2)
\]

\[
\xi(t) = \text{logsig}(0.5 \cdot \frac{\text{maxiteration} - t}{k}) \cdot \text{rand}()\]

where \(x_{\text{new}}^i\) represents the \(i^{\text{th}}\) new individual with feature subset and weight parameters; \(x_{\text{selected}}^i\) represents the \(i^{\text{th}}\) individual to be updated with feature subset and weight parameters; \(N(\mu, \sigma^2)\) are random numbers based on normal distribution; \(\text{rand}()\) is a function that generates random numbers in the range of \((0, 1)\); \(\text{maxiteration}\) represents the maximum number of iterations, while \(t\) represents the current number of iterations; \(k\) is the coefficient controlling the \(\text{logsig()}\) function, which is used to change the search step size of \(\xi(t)\), and then to balance the convergence speed of the algorithm. The transfer function \(\text{logsig()}\) is defined as Eq.(11):

\[
\text{logsig}(a) = \frac{1}{1 + \exp(-a)}
\]

For two clusters are selected, the new individuals generated are based on two individuals. A random value is generated, and compared with the probability value \(P_{\text{two}}\). We randomly select the central individual of two clusters to generate new individuals, if the random value is less than the probability value \(P_{\text{two}}\); otherwise, we randomly select the common individual of two clusters to generate new individuals. A new individual is generated by two existing individuals \(x_{\text{selected1}}^i\) and \(x_{\text{selected2}}^i\), and \(x_{\text{selected1}}^i\) represents the selected individual with feature subset and weight parameters in the first cluster and \(x_{\text{selected2}}^i\) represents the selected individual with feature subset and weight parameters in the second cluster. In this case, the selected individual \(x_{\text{selected}}^i\) can be written as:

\[
x_{\text{selected}}^i = x_{\text{selected1}}^i \cdot \text{tem} + x_{\text{selected2}}^i \cdot (1 - \text{tem})
\]

where \(\text{tem}\) is a random number in the range of \((0, 1)\), which is generated by \(\text{rand}()\).

Finally, the new individual selection is to keep the individuals with better fitness in the population. After generating new individuals, if the stop criteria are not met, the individuals with better functional fitness are saved to the next iteration by selection strategy. The strategy of clustering, generating, and selecting new individuals is used to introduce new individuals into the population to ensure the diversity of the population. The evolutionary classification model with feature selection based on the BSO algorithm (FS-CBSO) is described in Algorithm 1 in detail.

IV. EXPERIMENTAL DESIGN

A. DATASETS

Eleven datasets (DS1-DS11) used in the experiment are selected from the UCI Machine Learning Repository. In addition, in order to verify the performance of the algorithm on large-scale datasets, DS4, DS7 and DS8 are chosen in the experiment, where the data dimension of DS7 and DS8 is more than 500. Table 1 details all the datasets information, where “DSn” is the \(n^{\text{th}}\) dataset, “NoE” is the number of instances, “NoF” is the dimension of a dataset, and “NoC” is the number of classes. The datasets selected in this paper are classified into two classes.
Algorithm 1: The framework of the FS-CBSO algorithm

**Input:** Set parameter values including the number of population size ($N$), the number of clusters ($m$), the max number of iterations ($max_{iteration}$), the probability values ($P, P_{cluster}, P_{one}, P_{two}$);

**Output:** The optimal individual with feature subset and its corresponding weight parameters and classification accuracy.

1. **Initialization:** Randomly generate $N$ individuals with feature subset and parameters, and calculate the $N$ individuals’ fitness value;
2. while $current_{iteration} < max_{iteration}$ do
3. 3. **Cluster:** $N$ individuals are clustered into $m$ clusters and selected the central individuals of each cluster;
4. 4. if $rand() > P$ then
5. 5. A random individual is generated to replace a central individual randomly;
6. 6. end
7. 7. Generate new individuals: Randomly selects one or two individuals to generate new individuals;
8. 8. for $i < N$ do
9. 9. if $rand() < P_{cluster}$ then
10. 10. if $rand() > P_{one}$ then
11. 11. Randomly select a central individual;
12. 12. end
13. 13. else
14. 14. Randomly select a common individual;
15. 15. end
16. 16. end
17. 17. else
18. 18. if $rand() > P_{two}$ then
19. 19. Randomly select two central individuals and fused by Eq.(12);
20. 20. end
21. 21. else
22. 22. Randomly select two common individuals and fused by Eq.(12);
23. 23. end
24. 24. end
25. 25. Generate new individuals based on the selected individual by Eq.(9);
26. 26. **Selection:** The fitness values of the old and new individuals were compared, and the individuals with smaller fitness values were left in the population;
27. end

B. PARAMETER SETTINGS

In the experiment, BSO is used to implement the evolutionary classification model. The detailed parameter values setting of BSO is shown in Table 2 [10], [28]. Among them, the threshold value $\theta$ for determining whether to select features is set to 0.6, which has been proved in [28]. In the experiment, the maximum evaluation times ($max_{iteration}$) of the two algorithms are 100,000. Use the BSO algorithm to find $W$ in all datasets to minimize the objective function $\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{d}(w_j \cdot x_{ij} - y_i^2)}$. Then, $W$ is used to solve the classification label of each instance. Finally, the classification accuracy and the solution size on each dataset are calculated, and each dataset is repeated for 30 times to obtain more general results.

V. RESULTS AND ANALYSIS

In the experiment, the performance of the FS-CBSO method the evolutionary classification model based on the BSO (CBSO) method is compared. Table 3 and Table 4 are the two algorithms’ experimental results. Table 3 shows the classification accuracy of the two algorithms on 11 testing sets and Table 4 shows the solution size of the two algorithms on 11 testing sets. “Mean” is the average value of classifi-
FIGURE 4. Box plots of classification accuracy on test sets (DS1-DS11).

A. RESULT ANALYSIS OF CLASSIFICATION ACCURACY

As can be seen from Table 3, FS-CBSO can achieve the highest classification accuracy on most datasets. Specifically, the classification accuracy of FS-CBSO is significantly higher than that of CBSO on the eight datasets (DS2, DS3, DS4, DS6, DS7, DS9, DS10, and DS11). In addition, the classification accuracy of FS-CBSO is similar to
CBSO in DS5 and DS8, while FS-CBSO is inferior to CBSO only in DS1. Besides, on large-scale dataset DS4 and high-dimensional dataset DS9, FS-CBSO is superior to CBSO in classification performance.

**B. RESULT ANALYSIS OF SOLUTION SIZE**

Table 4 shows that FS-CBSO can reduce the features by 63% ~ 79% on eight datasets. For DS4, DS5 and DS7, more than 85% of features can be reduced. Compared with CBSO, FS-CBSO can effectively reduce the number of features and achieve the expected effect of optimizing the structure of the classification model.

**C. RESULT ANALYSIS OF BOX PLOTS**

Figure 4 intuitively shows the robustness and classification accuracy of the two algorithms on the 11 datasets. We can see that the stability of the FS-CBSO method is better than that of the CBSO method in 6 datasets and worse than the CBSO method in 5 datasets. However, the box plots show that the classification accuracy of the FS-CBSO method is better than the CBSO method in eight datasets and worse than the CBSO method in only one dataset. Generally speaking, the FS-CBSO method is superior to CBSO in classification accuracy and stability.

**TABLE 4. Solution size of the two algorithms on eleven testing sets.**

| Datasets | CBSO mean | CBSO std | FS-CBSO mean | FS-CBSO std |
|----------|-----------|----------|--------------|-------------|
| DS1      | 9         | 0        | 1.9          | 0.48        |
| DS2      | +         | 30       | 7.03         | 1           |
| DS3      | +         | 33       | 8.2          | 2.16        |
| DS4      | +         | 41       | 4.57         | 1.76        |
| DS5      | +         | 44       | 7.4          | 3.07        |
| DS6      | 60        | 60       | 16.17        | 4.09        |
| DS7      | +         | 24       | 3.4          | 0.93        |
| DS8      | 100       | 36.73    | 4.27         |             |
| DS9      | +         | 1300     | 320.03       | 40.16       |
| DS10     | +         | 500      | 160          | 23.04       |
| DS11     | +         | 166      | 57.83        | 5.93        |

To sum up, we can conclude that FS-CBSO has a smaller solution size than CBSO while ensuring the classification accuracy. The introduction of feature selection in the evolutionary optimization classification model can solve the problem of complex structure of the model, especially on large-scale and high-dimensional datasets.

**VI. CONCLUSION**

In this paper, based on the original evolutionary classification model, feature selection is introduced to optimize the structure of the model. Then, the BSO algorithm regards the optimization problem as a search problem and searches the optimal value in the solution space: searching the optimal structure and its corresponding weight parameters. In addition, in order to verify the effectiveness of the proposed method, 11 different datasets were selected to test the classification performance. The experimental results show that FS-CBSO has better classification accuracy and smaller solution size than CBSO on most of the 11 datasets, especially on large-scale or high-dimensional datasets, which strongly proves the feasibility of the proposed method.

However, the existing work has the defect of insufficient comparative experiments. For example, the problem solved is also limited to solving two classification label problems, and the classification performance has not reached the ideal effect. Therefore, in future work, we will compare the FS-CBSO algorithm with other algorithms. In addition, there are some limitations in using the minimum error value as the evaluation standard, and we will consider other more suitable objective functions. Besides, how to set the boundary value of solution space is also a subject worthy of study.

**REFERENCES**

[1] Y. Lei, F. Jia, J. Lin, S. Xing, and X. X. Ding, “An intelligent fault diagnosis method using unsupervised feature learning towards mechanical big data,” IEEE Trans. Ind. Electron., vol. 63, no. 5, pp. 3137–3147, May 2016.

[2] J. E. Guin, “Classification bias of the k-nearest neighbor algorithm,” IEEE Trans. Pattern Anal. Mach. Intell., vols. PAMI-6, no. 5, pp. 379–381, May 1984.

[3] X.-Z. Wang, Y.-L. He, and D. D. Wang, “Non-naive Bayesian classifiers for classification problems with continuous attributes,” IEEE Trans. Cybern., vol. 44, no. 1, pp. 21–39, Jan. 2014.

[4] B. Liu, Y. Xiao, and L. Cao, “SVM-based multi-state-mapping approach for multi-class classification,” Knowl.-Based Syst., vol. 129, pp. 79–96, Aug. 2017.

[5] L. Rokach and O. Maimon, “Decision trees,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 35, no. 4, pp. 476–487, May 2005.

[6] M.-J. Huang, Y.-L. Tsou, and S.-C. Lee, “Integrating fuzzy data mining and fuzzy artificial neural networks for discovering implicit knowledge,” Knowl.-Based Syst., vol. 19, no. 6, pp. 396–403, Oct. 2006.

[7] M. V. F. Veriz, M. Birattari, and T. Stutzle, “Ant colony optimization,” IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28–39, Nov. 2006.

[8] T. Y. Tan, L. Zhang, S. C. Neoh, and C. P. Lim, “Intelligent skin cancer detection using improved particle swarm optimization,” Knowl.-Based Syst., vol. 158, pp. 118–135, Oct. 2018.

[9] Y.-W. Chang Chien and Y.-L. Chen, “Mining associative classification rules with stock trading data—A GA-based method,” Knowl.-Based Syst., vol. 23, no. 6, pp. 605–614, Aug. 2010.

[10] H. Zhu and Y. Shi, “Brain storm optimization algorithms with k-medians clustering algorithms,” in Proc. 7th Int. Conf. Adv. Comput. Intell. (ICACI), Mar. 2015, pp. 107–110.

[11] F. Al-Obeidat, N. Belacel, J. A. Carretero, and P. Mahanti, “Differential evolution for learning the classification method PROAFTN,” Knowl.-Based Syst., vol. 23, no. 5, pp. 418–426, Jul. 2010.

[12] M. A. El Majdoubi, I. Ribouh, S. Bourjine, B. El Benani, and A. A. El Imrani, “Fireworks algorithm framework for big data optimization,” Memetic Comput., vol. 8, no. 4, pp. 333–347, Dec. 2016.

[13] V. Coletto-Alcudia and M. A. Vega-Rodríguez, “Artificial bee colony algorithm based on dominance (ABCD) for a hybrid gene selection method,” Knowl.-Based Syst., vol. 205, Oct. 2020, Art. no. 106323.

[14] A. Majumdar, T. Debath, K. S. Sood, and K. L. Baishnab, “Kyasunar forest disease classification framework using novel extremal optimization tuned neural network in fog computing environment,” J. Med. Syst., vol. 42, no. 10, pp. 1–16, Oct. 2018.

[15] M. El-Abd, “Global-best brain storm optimization algorithm,” Swarm Evol. Comput., vol. 37, pp. 27–44, Dec. 2017.

[16] H. Duan and C. Li, “Quantum-Behaved brain storm optimization approach to solving Loney’s solenoid problem,” IEEE Trans. Magn., vol. 51, no. 1, pp. 1–7, Jan. 2015.
[17] H. Liang, Z. Wang, and Y. Liu, “A new hybrid ant colony optimization based on brain storm optimization for feature selection,” IEICE Trans. Inf. Syst., vol. E102.D, no. 7, pp. 1396–1399, 2019.

[18] T. Si and R. Dutta, “Partial opposition-based particle swarm optimizer in artificial neural network training for medical data classification,” Int. J. Inf. Technol. Decis. Making, vol. 18, no. 05, pp. 1717–1750, Sep. 2019.

[19] Y. R. Nugraha, A. P. Wibawa, and I. A. E. Zaeni, “Particle swarm optimization—Support vector machine (PSO-SVM) algorithm for journal rank classification,” in Proc. 2nd Int. Conf. Comput. Informat. Eng. (IC2IE), Sep. 2019, pp. 69–73.

[20] S. Rajathi and G. Radhamani, “Prediction and analysis of rheumatic heart disease using kNN classification with ACO,” in Proc. Int. Conf. Data Mining Adv. Comput. (SAPIENCE), Mar. 2016, pp. 68–73.

[21] Y. Xue and T. Tang, “Classification based on brain storm optimization algorithm,” in Proc. Int. Conf. Bio-Inspired Comput., Theories Appl., 2016, pp. 371–376.

[22] Y. Xue and B. Zhao, “Classification based on fireworks algorithm,” in Proc. Int. Conf. Bio-Inspired Comput., Theories Appl. Berlin, Germany: Springer, 2016, pp. 35–40.

[23] B. Xue, M. Zhang, and W. N. Browne, “Novel initialisation and updating mechanisms in PSO for feature selection in classification,” in Proc. Eur. Conf. Appl. Evol. Comput. Berlin, Germany: Springer, 2013, pp. 428–438.

[24] M. N. Tajik, A. U. Rehman, W. Khan, and B. Khan, “Texture feature selection using ga for classification of human brain mri scans,” in Proc. Int. Conf. Swarm Intell., 2016, pp. 233–244.

[25] A. Frank and A. Asuncion. (2010). UCI Machine Learning Repository. Accessed: Apr. 25, 2018. [Online]. Available: http://archive.ics.uci.edu/ml

[26] Y. Xue, B. Zhao, T. Ma, and W. Pang, “A self-adaptive fireworks algorithm for classification problems,” IEEE Access, vol. 6, pp. 44406–44416, 2018.

[27] Y. Shi, “Brain storm optimization algorithm,” in Proc. Int. Conf. Swarm Intell. Berlin, Germany: Springer, 2011, pp. 303–309.

[28] B. Xue, M. Zhang, and W. N. Browne, “Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms,” Appl. Soft Comput., vol. 18, pp. 261–276, May 2014.

YU XUE (Member, IEEE) received the Ph.D. degree from the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China, in 2013. From 2016 to 2017, he was a Visiting Scholar with the Victoria University of Wellington, Wellington, New Zealand. He is currently an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology. He is also a Visiting Scholar of the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA. He has published more than 20 journal and conference papers. His research interests include evolutionary computation, feature selection, adaptive search, evolutionary machine learning. He is a member of ACM (2270255) and CCF (E200029023M).

YAN ZHAO

ADAM SLOWIK (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering and electronics, in 2001, the Ph.D. degree (Hons.) from the Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland, in 2007, and the Dr.Habil. degree in computer science (intelligent systems) from the Department of Mechanical Engineering and Computer Science, Czestochowa University of Technology, Czestochowa, Poland, in 2013. Since October 2013, he has been an Associate Professor with the Department of Electronics and Computer Science, Koszalin University of Technology. He is currently the Head of the Department of Computer Engineering, Koszalin University of Technology. His research interests include soft computing, computational intelligence, and, particularly, bio-inspired optimization algorithms and their engineering applications. He is the author or coauthor of over 100 refereed articles in international journals, two books, and conference proceedings, including one invited talk. He is a member of the program committees of several important international conferences in the area of artificial intelligence and evolutionary computation. He was a recipient of one Best Paper Award (IEEE Conference on Human System Interaction - HSI 2008). He is a Reviewer of many international scientific journals. He is an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.