Knowledge-Based Systems 96 (2016) 61–75

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Evolving support vector machines using fruit fly optimization for medical data classification

Liming Shen[a], Huiling Chen[a,c,*], Zhe Yu[a], Wenchang Kang[a], Bingyu Zhang[a], Huaizhong Li[a], Bo Yang[b,c], Dayou Liu[b,c]

[a] College of Physics and Electronic Information Engineering, Wenzhou University, 325035 Wenzhou, China
[b] College of Computer Science and Technology, Jilin University, Changchun 130012, China
[c] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

ARTICLE INFO

Article history:
Received 7 March 2015
Revised 31 December 2015
Accepted 2 January 2016
Available online 11 January 2016

Keywords:
Support vector machine
Parameter optimization
Fruit fly optimization
Medical diagnosis

ABSTRACT

In this paper, a new support vector machines (SVM) parameter tuning scheme that uses the fruit fly optimization algorithm (FOA) is proposed. Termed as FOA-SVM, the scheme is successfully applied to medical diagnosis. In the proposed FOA-SVM, the FOA technique effectively and efficiently addresses the parameter set in SVM. Additionally, the effectiveness and efficiency of FOA-SVM is rigorously evaluated against four well-known medical datasets, including the Wisconsin breast cancer dataset, the Pima Indians diabetes dataset, the Parkinson dataset, and the thyroid disease dataset, in terms of classification accuracy, sensitivity, specificity, AUC (the area under the receiver operating characteristic (ROC) curve) criterion, and processing time. Four competitive counterparts are employed for comparison purposes, including the particle swarm optimization algorithm-based SVM (PSO-SVM), genetic algorithm-based SVM (GA-SVM), bacterial forging optimization-based SVM (BFO-SVM), and grid search technique-based SVM (Grid-SVM). The empirical results demonstrate that the proposed FOA-SVM method can obtain much more appropriate model parameters as well as significantly reduce the computational time, which generates a high classification accuracy. Promisingly, the proposed method can be regarded as a useful clinical tool for medical decision making.

1. Introduction

As a primary machine learning paradigm, support vector machines (SVM) [1,2] are rooted in the Vapnik–Chervonenkis theory and the structural risk minimization principle. The SVM attempts to determine a tradeoff between minimizing the training set error and maximizing the margin in order to achieve the best generalization ability and remain resistant to over fitting. Additionally, one major advantage of the SVM is the use of convex quadratic programing, which provides only global minima; thus, it avoids being trapped in local minima. Due to its advantageous nature, SVM has been applied to a wide range of classification tasks [3–6]. In particular, SVM has been shown to perform very well on many medical diagnosis tasks [5–11]. However, there is still a need for improving the SVM classifier's performance.

It has been demonstrated that the SVM classification accuracy can be substantially improved by establishing the proper model parameter settings [12]. Thus, key parameters, such as the penalty parameter and the kernel bandwidth of the kernel function, should be properly determined prior to its application to practical problems. The first parameter, the penalty parameter C, determines the trade-off between the fitting error minimization and the model complexity. The second parameter, the kernel bandwidth γ, defines the non-linear mapping from the input space to some high-dimensional feature space. Traditionally, these parameters were handled by the grid-search method [13] and the gradient descent method [14–16]. However, these methods are vulnerable to local optimum. Recently, biologically-inspired metaheuristics (such as the genetic algorithm [17], particle swarm optimization (PSO) [18], and bacterial foraging optimization (BFO) [19]) have been shown to be more likely to determine the global optimum solution than the traditional aforementioned methods.

Not only are the traditional optimization algorithms, such as GA, BFO, and PSO, complex to implement, but they are also difficult to understand. In addition, determining the global optimal solution is time-consuming. As a new member of the swarm-intelligence algorithms, the fruit fly optimization algorithm (FOA) [20] is inspired by the foraging behavior of real fruit flies. The FOA has certain outstanding merits, such as a simple computational process,

* Corresponding author at: College of Physics and Electronic Information Engineering, Wenzhou University, 325035 Wenzhou, China. Tel.: +86057786689125.
E-mail address: chenhuiling.jlu@gmail.com (H. Chen).

http://dx.doi.org/10.1016/j.knosys.2016.01.002
0950-7051/

simple implementation, and easy understanding with only a few parameters for tuning. Due to its good properties, FOA has become a useful tool for many real-world problems. In 2012, Li et al. [21] proposed the FOA to optimize the parameters in LSSVM (LSSVM-FOA) and applied it to forecast the annual electric load. The computational result showed that the proposed method outperformed the other alternative methods. In 2013, Chen et al. [22] proposed using the FOA to tune the parameters in the gray model neural network and applied the resultant model, FOAGMNN, to construct service satisfaction detection. Based on the experimental analysis results, the FOAGMNN model achieved the fastest error convergence and the best classification capability. In 2013, Li et al. [23] developed a hybrid annual power load forecasting model that combined FOA and generalized regression neural network (GRNN). In the proposed method, FOA was used to determine the appropriate spread parameter in GRNN, and the effectiveness of this proposed hybrid model was demonstrated in two experiment simulations. Both experiments revealed that the proposed hybrid model outperformed the GRNN model with the default parameter and others. In 2013, Shan et al. [24] presented an improved FOA (LGMS-FOA) that enhanced the FOA's performance. LGMS-FOA's superiority was demonstrated by simulation results and by comparing LGMS-FOA to FOA and other metaheuristics. Both the searching efficiency and the searching quality were greatly improved. In 2013, Wang et al. [25] proposed a novel binary fruit fly optimization algorithm (bFOA) to solve the multidimensional knapsack problem (MKP). In the bFOA, several techniques were introduced, including the binary string, local and global vision-based search process, a group generating probability vector, a global vision mechanism, two repair operators, and the Taguchi method. A comparative study demonstrated that the proposed bFOA effectively solved the MKP, especially for large-scale problems. Recently, Pan [26] proposed a modified FOA (MFOA), in which an escape parameter is added to the distance function. The findings showed that the forecasting model that combines MFOA and GRNN has the best ability for forecasting the closing price of both oil and gold. Pan and colleagues [27] presented an improved fruit fly optimization (IFFO) algorithm that introduced a new control parameter to adaptively tune the search scope around its swarm location for solving continuous function optimization problems. Yuan et al. [28] proposed a variation on the original FOA technique, named the multi-swarm fruit fly optimization algorithm (MFOA), by employing multi-swarm behavior to significantly improve the performance. By applying the proposed MFOA approach to several benchmark functions and the parameter identification of a synchronous generator, it was demonstrated that the proposed approach is superior to the original FOA technique. Li et al. [29] presented an improved fruit fly optimization algorithm (FOA) via a well-designed smell search procedure to solve the steelmaking casting problem, and the simulation results indicate that the proposed FOA is more effective than the four presented algorithms. Zheng et al. [30] proposed a novel fruit fly optimization algorithm (nFOA) based on multiple fruit fly groups to solve the semiconductor final testing scheduling problem, and the experimental results demonstrated that the nFOA effectively and efficiently solved the SFTSP. More recently, Wang et al. [31] presented an effective and improved FOA (IFOA) for optimizing numerical functions and solving joint replenishment problems. The improvements include a new method for maintaining the population diversity in order to enhance the exploration ability, a new parameter to avoid local optimization, and a random perturbation to the updated initial location to jump out of the local optimum. Furthermore, a comparative study reveals that the proposed IFOA can obtain better solutions than the current best algorithm. Wang et al. [32] proposed an adaptive mutation fruit fly optimization algorithm (AM-FOA) to optimize the parameters in the least squares support vector machine (LSSVM), and they successfully applied the resultant model, AM-FOA-LSSVM, to the practical melt index prediction problem. Xiao et al. [33] presented an improved FOA based on the cell communication mechanism (CFOA) that takes into consideration the global worst, mean, and best solutions into the search strategy to improve the exploitation. The results from a set of numerical benchmark functions show that the CFOA outperforms the FOA and the PSO in most experiments. Furthermore, the CFOA is applied to optimize the controller of peroxidation furnaces in carbon fibers production, and simulation results have validated the CFOA's effectiveness.

From the above FOA-related works, FOA has been proven to be effective for parameter optimization for machine learning algorithms, including the GRNN and LSSVM. Additionally, it should be noted that the related LSSVM parameter optimization based on FOA is only suitable for the regression problem; it cannot be used for classification problems. Therefore, this study will be the first to explore the FOA technique's ability to address SVM's model selection problem for classification. Furthermore, the resultant model, FOA-SVM, successfully and effectively detected the medical data classification problem. In the proposed FOA-SVM method, the cross validation classification accuracy is considered for designing the objective function to explore SVM's maximum generalization ability. The proposed method's effectiveness and efficiency was examined in terms of the classification accuracy, sensitivity, specificity, AUC, and CPU time on the medical datasets taken from the UCI machine learning repository. As the experimental results will show, our proposed method can obtain more appropriate model parameters and obtain a high predictive accuracy with much less processing time as compared to the grid search-based and other metaheuristic-based methods.

The remaining of this paper is organized as follows. Section 2 gives some brief background knowledge of SVM and FOA. The detailed implementation of the FOA-SVM methodology will be explained in Section 3. Section 4 describes the experimental design in detail. The experimental results and discussions of the proposed approach are presented in Section 5. Finally, conclusions are summarized in Section 6.

2. Background materials

2.1. Support vector machines (SVM)

This section gives a brief description of SVM. For more details, one can refer [2,34], which provides a complete description of the SVM theory.

Let us consider a binary classification task: $\{xi, yi\}$, $i = 1, \ldots, l$, $yi \in \{-1, 1\}$, $xi \in R^d$, where xi are data points, and yi are the corresponding labels. They are separated with a hyper plane given by $w^T x + b = 0$, where w is a d-dimensional coefficient vector that is normal to the hyper plane, and b is the offset from the origin. The linear SVM obtains an optimal separating margin by solving the following optimization task:

$$\begin{aligned} &\mathrm{Ming}(\mathrm{w},\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ &s.t.\ ,\ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \end{aligned} \tag{1}$$

By introducing Lagrangian multipliers $\alpha_i (i = 1, 2, \ldots, n)$, the primal problem can be reduced to a Lagarangian dual problem:

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \\ &s.t.\ ,\ \ \alpha_i \geq 0, i = 1, \ldots, n,\ \sum_{i=1}^{n}\alpha_i y_i = 0 \end{aligned} \tag{2}$$

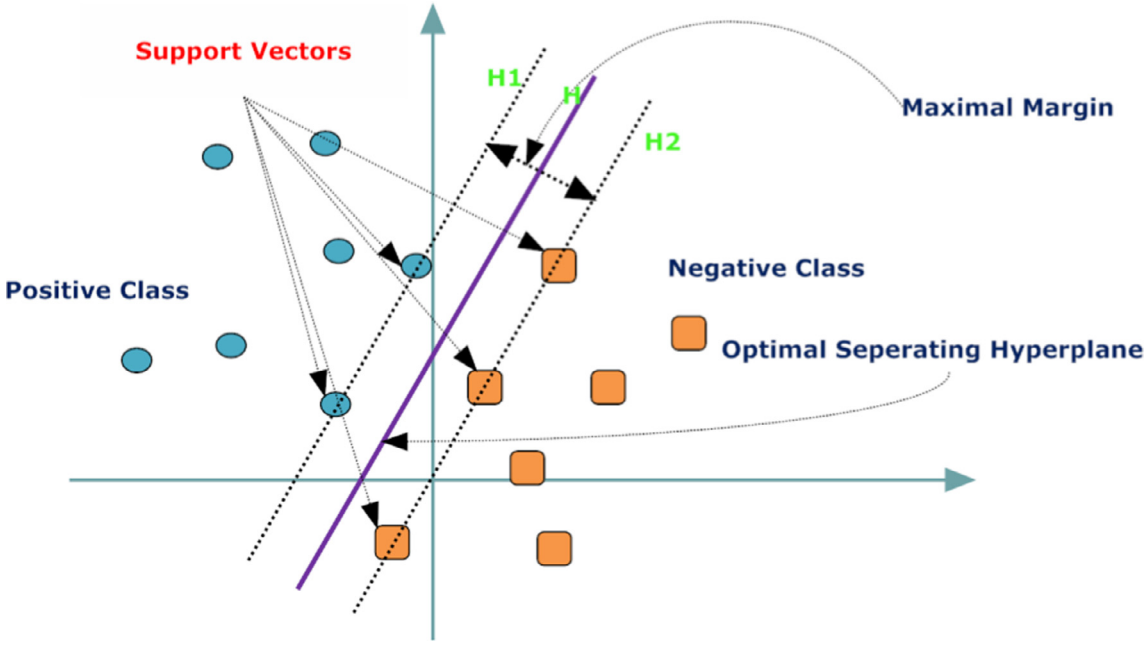

Fig. 1. The schematic model of a linear SVM.

It is clearly a quadratic optimization problem with linear constraints. From the Karush–Kuhn–Tucker (KKT) condition, we know: $\alpha_i(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1) = 0$. If $\alpha_i > 0$. The corresponding data points are called SVs. Hence, the solution takes the following form: $w = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$, where n is the number of SVs. Now, b can be obtained from $y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 = 0$, where $\mathbf{x}_i$ are SVs. After w and b are determined, the linear discriminant function can be given by Eq. (3).

$$g(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b\right) \tag{3}$$

Fig. 1 illustrates an operating model of a linear SVM, where the solid line H in the figure is the final SVM solution. According to the core concept of SVM, the classification of patients with disease can be viewed as a simple SVM application; that is, whether a certain prediction result belongs to a positive class.

In most cases, the two classes cannot be linearly separated. In order for the linear learning machine to work well in non-linear cases, a general idea is introduced. That is, the original input space can be mapped into some higher-dimensional feature space where the training set is linearly separable. With this mapping, the decision function can be expressed as:

$$g(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) + b\right) \tag{4}$$

where $\mathbf{x}_i^T\mathbf{x}$ in the input space is represented as the form $\phi(\mathbf{x}_i)^T\phi(\mathbf{x})$ in the feature space. It is not necessary to know the functional form of the mapping $\phi(\mathbf{x}_i)$ since it is implicitly defined by one selected kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. Thus, the decision function can be expressed as follows:

$$g(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{5}$$

In general, any positive semi-definite functions that satisfy Mercer's condition can be kernel functions [35]. As shown in Table 1, there are many kernel functions that can be used in the SVM. From among these kernels, the two most widely used kernels in SVM are the polynomial kernel and the Gaussian kernel (or Radial-Basis function, RBF). Their functions are given in Table 1, where d is the polynomial order, and γ is the predefined parameter controlling the width of the Gaussian kernel.

Table 1
The common kernel functions.

Kernel types	Kernel functions
Linear kernel	$K(x, x_i) = (x^T x_i)$
Polynomial kernel	$K(x, x_i) = ((x^T x_i) + 1)^d$
Radial based kernel (RBF)	$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2)$
Sigmoid kernel	$K(x, x_i) = \tanh((x^T x_i) + b)$

2.2. Fruit-fly optimization algorithms

The fruit fly optimization algorithm (FOA) [20] was developed based on the foraging behavior of *Drosophila*. The fruit fly is superior to other species in olfactory capabilities and visual sense; thus, it is capable of fully utilizing its instinct to locate food. More specifically, even at a distance of 40 km from the food source, the nose of a fruit fly can pick up various food scents that are dispersed throughout the air. Upon nearing the food source, the fruit flies will locate the food and the company's flocking location with the aid of their sensitive visual organs, and then, they will fly in that direction. The best fruit fly's information will be shared with the whole swarm during the iteration, and the next iteration will be based only on the information of the previous best fruit fly. The food searching iterative process of a fruit fly swarm is shown in Fig. 2.

According to the food searching characteristics of a fruit fly swarm, the FOA can be divided into several steps as follows:

Step 1. Parameters initialization.

Initialize the parameters of the FOA, such as the maximum iteration number, the population size, the initial fruit fly

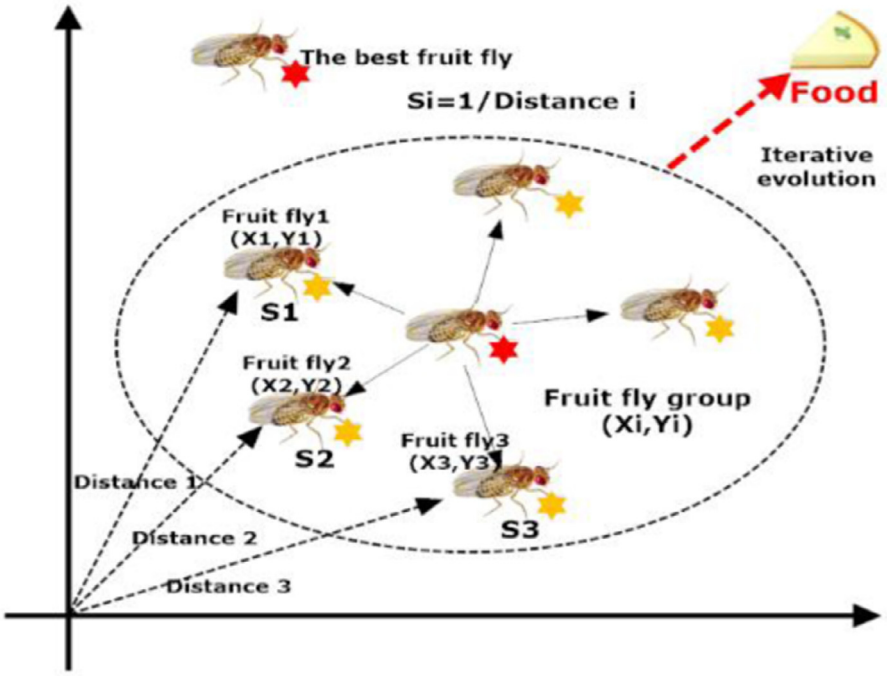

Fig. 2. The food searching iterative process of a fruit fly swarm.

swarm location (*X*_axis, *Y*_axis), and the random flight distance range.

$$X\_\text{axis} = \text{rands}(1,\ 2) \tag{6}$$

$$Y\_\text{axis} = \text{rands}(1,\ 2) \tag{7}$$

Step 2. Population initialization.

Give the random location (X_i, Y_i) and distance for the food search of an individual fruit fly, where i represents the population size.

$$X_i = X\_axis + RandomValue \tag{8}$$

$$Y_i = Y\_axis + RandomValue \tag{9}$$

Step 3. Population evaluation.

Firstly, calculate the distance of the food location to the origin (D). Then, compute the smell concentration judgment value (S), which is the reciprocal of the distance of the food location to the origin.

$$D_i = \sqrt{X_i^2 + Y_i^2} \tag{10}$$

$$S_i = 1/D_i \tag{11}$$

Step 4. Replacement.

Replace the smell concentration judgment value (S) with the smell concentration judgment function (also called the Fitness function) so as to find the smell concentration ($Smell_i$) of the individual location of the fruit fly.

$$Smell_i = \text{Function}(S_i) \tag{12}$$

Step 5. Find the maximal smell concentration.

Determine the fruit fly with the maximal smell concentration and the corresponding location among the fruit fly swarm.

$$[bestSmell bestIndex] = max(Smell) \tag{13}$$

Step 6. Keep the maximal smell concentration.

Retain the maximal smell concentration value and coordinates x and y. Then, the fruit fly swarm flies towards the location with the maximal smell concentration value.

$$Smellbest = bestSmell \tag{14}$$

$$X\_axis = X(bestIndex) \tag{15}$$

$$Y\_axis = Y(bestIndex) \tag{16}$$

Step 7. Iterative optimization.

Enter the iterative optimization to repeat the implementation of step 2–5. The circulation stops when the smell concentration is no longer superior to the previous iterative smell concentration or when the iterative number reaches the maximal iterative number.

3. The proposed FOA-SVM methodology

This study proposes a novel evolutionary SVM that employs the FOA strategy, and the resultant FOA-SVM model can adaptively determine the two key hyper-parameters for SVM. The general framework of the proposed method is demonstrated in Fig. 3. The proposed model is primarily comprised of two procedures: the inner parameter optimization and the outer classification performance evaluation. During the inner parameter optimization procedure, the SVM parameters are dynamically adjusted by the FOA technique via the 5-fold cross validation (CV) analysis. Then, the obtained optimal parameters are fed to the SVM prediction model to perform the classification task for PD diagnosis in the outer loop using the 10-fold CV analysis.

The classification accuracy is taken into account in designing the fitness:

$$f = avgACC = (\Sigma_{i=1}^{K} testACC_i)/k \tag{17}$$

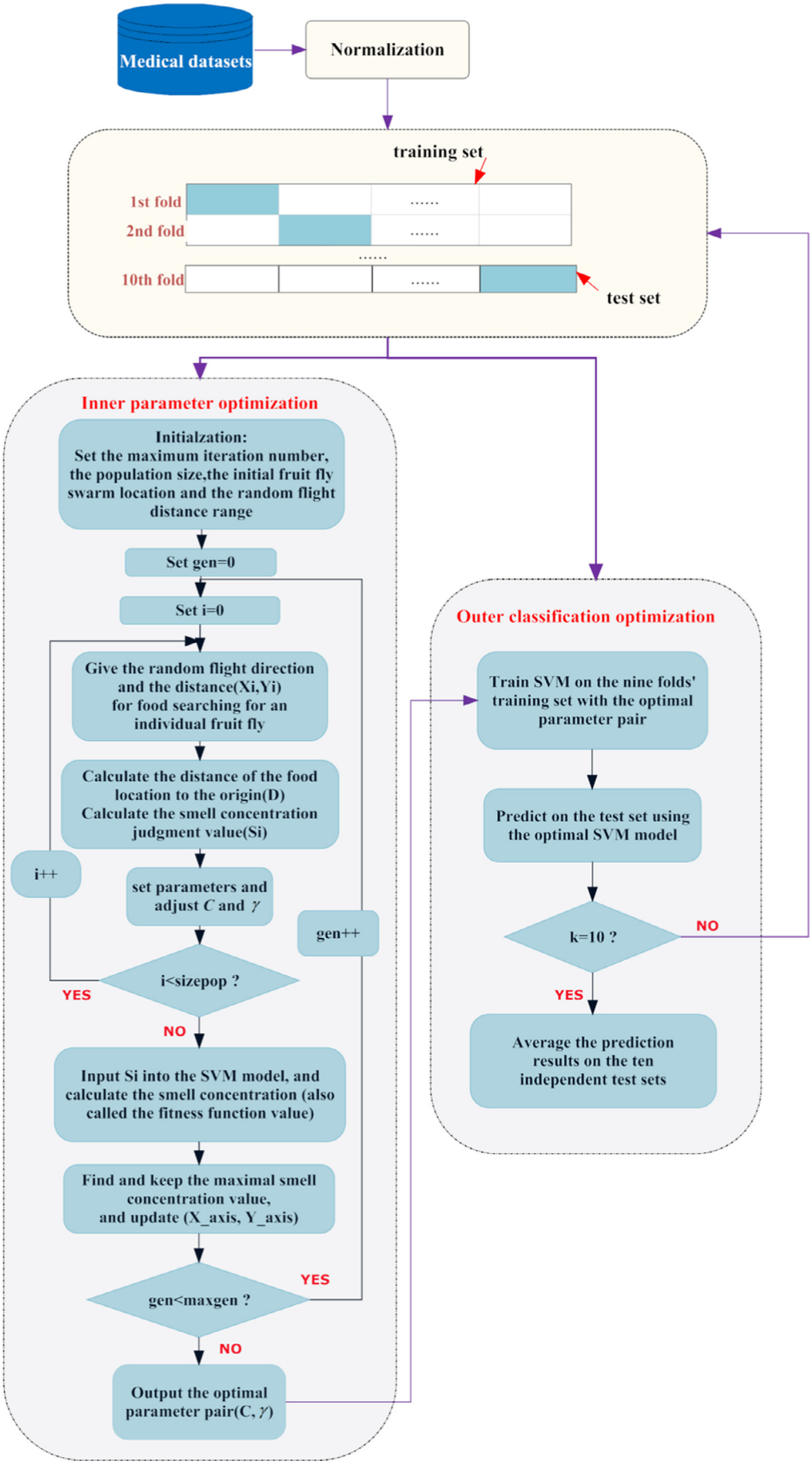

Fig. 3. Flowchart of the proposed FOA-SVM method.

Table 2
Description of datasets used in the experiments.

No.	Datasets	# of classes	# of instances	# of features	Miss.
1	Wisconsin breast cancer (Wisconsin)	2	699	9	Yes
2	Pima Indians diabetes (Pima)	2	768	8	No
3	Parkinson	2	195	22	No
4	Thyroid	3	215	5	No

'Miss.' means 'Missing value'.

where *avgACC* in the function *f* represents the average test accuracy achieved by the SVM classifier via the 5-fold CV. The pseudo-code of the parameter adjustment is as follows:

```
Begin
  For i = 1 to sizepop
    Set the SVM parameters with the initialized distance reciprocal S(i,1)
and S(i,2);
    Calculate the initial fitness;
    Train the SVM model with the distance reciprocal, and record test
results into the Smell array;
  End
    [bestSmell, bestindex] = max(Smell);
    past position = current position;
    bestCV = bestSmell;
    bestC = S(bestIndex,1);
    bestg = S(bestIndex,2);
  For j = 1: maxgen
    For i = 1 to sizepop
      X(i,:) = X_axis + ax*rand() − bx;
      Y(i,:) = Y_axis + ax*rand() − bx;
      D(i,1) = sqrt(X(i,1)^2 + Y(i,1)^2);
      D(i,2) = sqrt(X(i,2)^2 + Y(i,2)^2);
      S(i,1) = 1/D(i,1);
      S(i,2) = 1/D(i,2);
      Set the SVM parameters with S(i,1) and S(i,2);
      Calculate the initial fitness;
      Train the SVM model with the distance reciprocal, and record test
results into the Smell array;
  End
    [bestSmell, bestIndex] = max(Smell);
  If (bestSmell > bestCV)
      past position = current position;
      bestC = S(bestIndex,1);
      bestg = S(bestIndex,2);
      bestCV = bestSmell;
    End If
  End
  Return bestC, bestg;
End
```

The computational complexity of the FOA-SVM method depends upon the number of training samples (L), the population number (p), the number of generations (g), and the scale of the problem (D). Therefore, the overall computational complexity is O(SVM, FOA) = O(Initialization) + g*(O(SVM) + O(FOA position updating) + O(FOA global best calculation) + O(Output)).

SVM's computational complexity on L training samples is equal to $O(L^3)$. The computational complexity of initialization is O (p*D*$O(L^3)$)+$O(p)$, which represents the summation of the fitness calculation and the global best calculation. Therefore, the final computational complexity of the FOA-SVM method is O(SVM, FOA) = $O(p^*D^*O(L^3))+O(p)+g^*(O(p^*D^*O(L^3))+O(p^*D)+$ $O(p)+O(1)) \approx O(p^*D^*O(L^3))+g^*(O(p^*D^*O(L^3)))$.

4. Experimental studies

4.1. Data description

In order to evaluate the proposed FOA-SVM method, four biomedical datasets from the UCI machine learning data repository were used, including the Wisconsin breast cancer dataset, the Pima Indians diabetes dataset, the Parkinson dataset, and the thyroid disease dataset. The four medical datasets represent breast cancer, diabetes, Parkinson's disease, and thyroid disease diagnosis problems, respectively. Table 2 contains detailed descriptions of the datasets. Only the Wisconsin dataset contains missing values, and the mode of the attributes was used to replace the missing categorical attributes, and the mean of the attributes was used to replace the missing continuous ones.

Before constructing the SVM models, the data was scaled to the range [−1, 1] so that the feature values in the greater numerical ranges did not dominate those in the smaller numerical ranges and to avoid numerical difficulties during calculation. In order to guarantee valid results, the 10-fold CV was employed to evaluate the classification performance. In this study, we designed the experiment using a two-loop scheme, which was also used in [36]. However, it should be noted that a single repetition of the 10-fold CV would not generate enough classification accuracies for comparison. Therefore, to accurately evaluate the performance of all of the related methods, the 10-fold CV procedure was repeated 10 times, and the reported results were the average of all the runs.

4.2. Experimental setup

The FOA-SVM, PSO-SVM, Grid-SVM, GA-SVM, and BFO-SVM classification models were implemented using the MATLAB platform. For SVM, the LIBSVM implementation was utilized, which was originally developed by Chang and Lin [37]. We implemented the FOA, GA, PSO, BFO, and grid search algorithms from scratch. The computational analysis was conducted on a Windows Server 2008 operating system with Intel Xeon CPU E5-2650 v3(2.30 GHz) and 16GB of RAM.

In order to conduct an accurate comparison, the same number of generations and the same population swarm size were used for FOA, PSO, and GA. According to the preliminary experiment, when the number of generations and the swarm size are set to 250 and 8, respectively, the involved methods produce a satisfactory classification performance. For the Grid-SVM, the searching range of the parameters $C \in \{2^{-5}, 2^{-3},\ldots,2^{15}\}$ and $\gamma \in \{2^{-15}, 2^{-13},\ldots,2\}$ was adopted. For the metaheuristic methods, the same searching range of the parameters $C \in [2^{-5}, 2^{15}]$ and $\gamma \in [2^{-15}, 2]$ was used.

Each method's specific parameters are detailed in the following. For FOA-SVM, the main parameters are in the distance function as defined by $X(i,:) = X\_axis + ax^*rand() - bx$ and $Y(i,:) = Y\_axis + ax^*rand() - bx$, where X_axis and Y_axis are used to initialize the coordinate position of the fruit fly. The parameters of *ax*, *bx*, *ay*, and *by* in the above equations were set as 20, 10, 20, and 10, respectively. For PSO-SVM, the maximum velocity was set as approximately 60% of the dynamic range of the variable on each dimension. The acceleration coefficients c_1 and c_2 were set as follows: $c_1 = 2.05$, $c_2 = 2.05$, and the inertia weight w was set to 1. For GA-SVM, the crossover probability and mutation probability were set as 0.8 and 0.05, respectively. For BFO-SVM, the population swarm size was set as 8, the number of chemotactic steps was set as 50, the swimming length was set as 2, the number of reproduction steps was set as 2, the number of elimination-dispersal events was set as 2, the elimination-dispersal probability was set

Table 3
The nine attributes of the Wisconsin dataset.

Attribute	Description	Domain
F_1	Clump thickness	1–10
F_2	Uniformity of cell size	1–10
F_3	Uniformity of cell shape	1–10
F_4	Marginal adhesion	1–10
F_5	Single epithelial cell size	1–10
F_6	Bare nuclei	1–10
F_7	Bland chromatin	1–10
F_8	Normal nucleoli	1–10
F_9	Mitoses	1–10

Table 5
Paired *t*-test results of FOA-SVM and the other four methods in terms of ACC, AUC, sensitivity, and specificity on the Wisconsin dataset.

Metrics	*t*-value (significance)			
	PSO-SVM	Grid-SVM	GA-SVM	BFO-SVM
ACC	**5.038(0.001)**	**7.916(0.000)**	**7.416(0.000)**	**7.990(0.000)**
AUC	**4.486(0.002)**	**7.617(0.000)**	**8.078(0.000)**	**5.717(0.000)**
Sensitivity	**2.732(0.023)**	**3.736(0.005)**	**3.664(0.005)**	**9.993(0.000)**
Specificity	1.430(0.187)	**6.786(0.000)**	**8.228(0.000)**	**−3.469(0.007)**

Positive values indicate that the *i*th classifier has achieved a higher performance than the *j*th one. Significant values less than 0.05 are shown in bold.

as 0.25, and the size of step taken in the random direction specified by the tumble was set as 0.1. For the Grid-SVM, the nonlinear Gaussian (RBF) kernel was used, and the grid-search technique was employed using the 10-fold CV to determine the optimal parameter values of the RBF kernel function.

4.3. Measure for performance evaluation

The performance of the proposed model was evaluated using the following: the classification accuracy (ACC), the area under the receiver operating characteristic curve (AUC) criterion, the sensitivity, and the specificity. The ACC, sensitivity, and specificity are defined as follows:

$$Accuracy = TP + TN/(TP + FP + FN + TN) \times 100\% \quad (18)$$

$$Sensitivity = TP/(TP + FN) \times 100\% \quad (19)$$

$$Specificity = TN/(FP + TN) \times 100\% \quad (20)$$

where TP is the number of true positives, which represents cases that are correctly categorized in the 'positive' class; FN is the number of false negatives, which represents 'positive' class cases that are classified as negative; TN is the number of true negatives, which represents cases that are correctly categorized in the 'negative' class; and FP is the number of false positives, which represents 'negative' class cases that are classified as positive. The AUC is the area under the ROC curve, and it is one of the best methods for comparing classifiers in two-class problems. The method proposed in [38] was implemented to compute the AUC in this study.

5. Experimental results and discussions

Comparative experiments were performed between FOA-SVM and the other four competitive methods, including PSO-SVM, Grid-SVM, GA-SVM, and BFO-SVM, in order to evaluate the effectiveness of the proposed method for the four disease diagnosis problems.

5.1. Breast cancer diagnosis problem

The Wisconsin dataset consists of 699 instances and nine attributes, which were taken from needle aspirates from patients' breasts. The goal is to discriminate between the benign and malignant samples. The nine attributes in each sample were found to differ significantly between the benign and malignant samples. The nine attributes are detailed in Table 3. From the table, it can be seen that each attribute is graded from 1 to 10, with 10 being the most abnormal state. The class attribute was 2 for benign and 4 for malignant cases.

Table 4 illustrates the detailed classification results of the various methods in terms of the ACC, AUC, sensitivity, and specificity on the Wisconsin dataset. From the table, it can be seen that the FOA-SVM method achieves the highest performance among the five methods with average results of 96.90% ACC, 96.87% AUC, 96.86% sensitivity, and 96.89% specificity. It can be seen from the results that the standard deviation produced by the FOA-SVM is much smaller than that of the other four counterparts in terms of the ACC, AUC, sensitivity, and specificity. This indicates that the proposed FOA-SVM method offers considerably more stable and consistent results than the other four competitors. The detailed comparison results of 10 runs of the 10-fold CV for the five methods in terms of ACC are displayed in Fig. 4. It can be observed from the figure that FOA-SVM's performance is superior to those of the other four methods over the whole 10 runs.

In order to identify any differences among the five methods in terms of their classification performances, a paired *t*-test was conducted among them on the Wisconsin dataset. Furthermore, p-values of less than 0.05 were considered to be statistically significant in the experiment. From Table 5, it can be seen that FOA-SVM almost achieves significantly better results than the other four competitors in terms of the ACC, AUC, sensitivity, and specificity at the prescribed statistical significance level of 5%.

In order to investigate the computational efficiency of the proposed method, FOA-SVM was compared to the other four methods in terms of the CPU time. Fig. 5 illustrates the computational time in seconds of the five methods during one time run of the 10-fold CV on the Wisconsin dataset. As shown in Fig. 5, FOA-SVM requires an average of almost 71 s to complete the training and prediction procedure in each fold for the breast cancer diagnosis. Given the same generations and swarm size, FOA-SVM requires much less CPU time than either the PSO-SVM or BFO-SVM method. From among the five methods, BFO-SVM is the most time consuming (6.6 times more than that of FOA-SVM), whereas Grid-SVM requires the least amount of time.

Table 4
The detailed classification results of the various methods on the Wisconsin dataset.

Metrics	Methods				
	PSO-SVM	Grid-SVM	GA-SVM	BFO-SVM	FOA-SVM
ACC	0.9627 ± 0.0038	0.9624 ± 0.0025	0.9564 ± 0.0054	0.9557 ± 0.0056	0.9690 ± 0.0010
AUC	0.9626 ± 0.0042	0.9602 ± 0.0032	0.9529 ± 0.0063	0.9609 ± 0.0044	0.9687 ± 0.0009
Sensitivity	0.9624 ± 0.0070	0.9662 ± 0.0024	0.9627 ± 0.0050	0.9457 ± 0.0082	0.9686 ± 0.0014
Specificity	0.9659 ± 0.0068	0.9545 ± 0.0060	0.9432 ± 0.0099	0.9761 ± 0.0059	0.9689 ± 0.0018

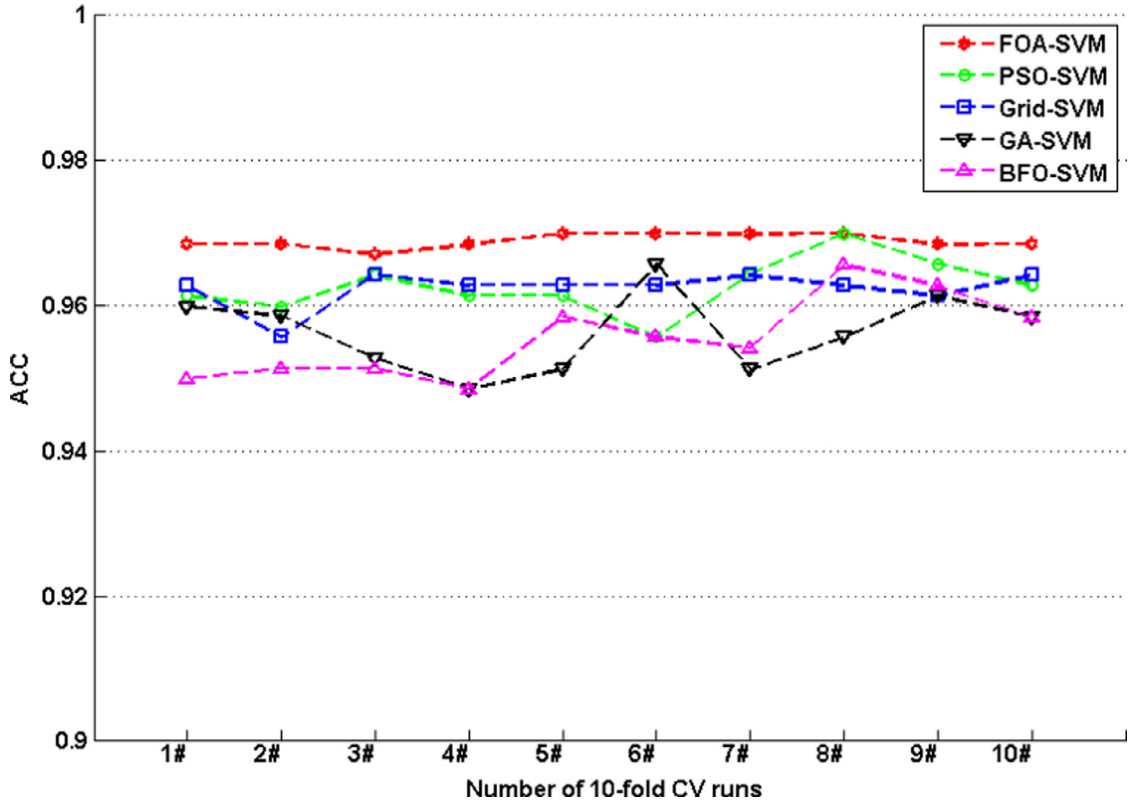

Fig. 4. The ACC obtained for each run by the five methods on the Wisconsin dataset.

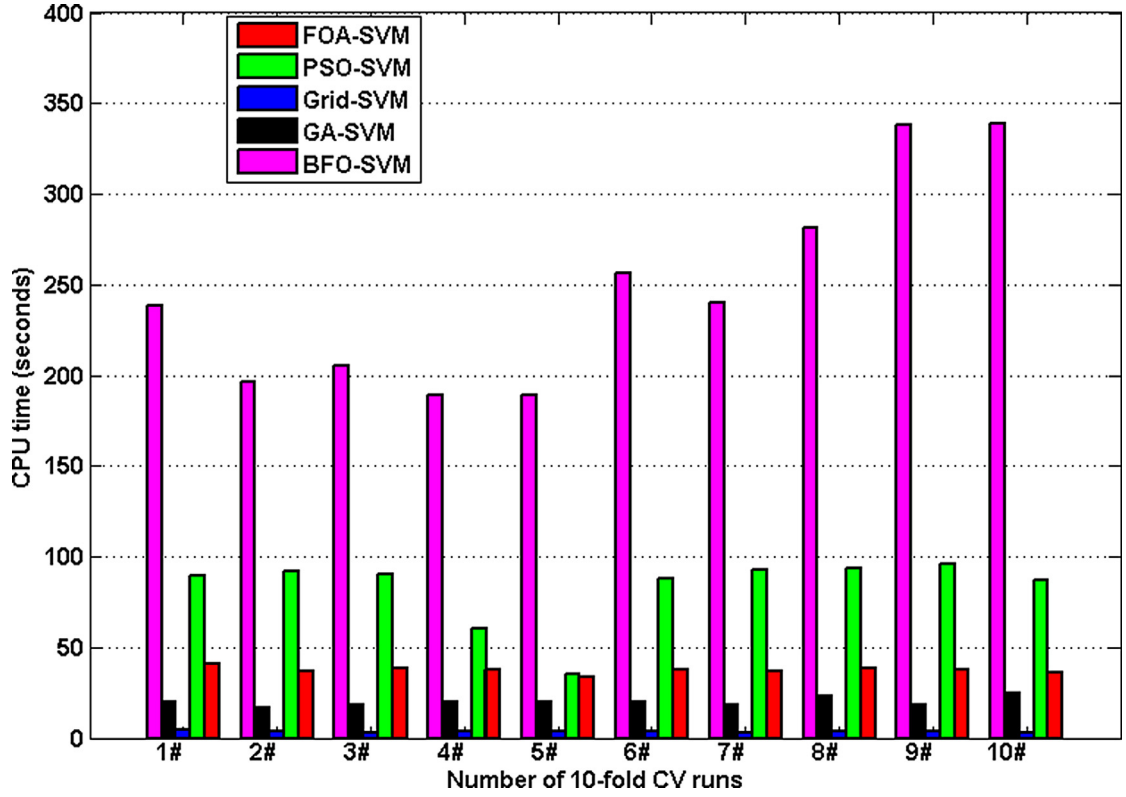

Fig. 5. Comparison results of the five methods in terms of CPU time on the Wisconsin dataset.

Table 6
The detailed classification results of the various methods on the Pima dataset.

Metrics	Methods				
	PSO-SVM	Grid-SVM	GA-SVM	BFO-SVM	FOA-SVM
ACC	0.7650 ± 0.0039	0.7648 ± 0.0029	0.7626 ± 0.0053	0.7647 ± 0.0049	0.7746 ± 0.0026
AUC	0.7146 ± 0.0046	0.7119 ± 0.0032	0.7114 ± 0.0062	0.7121 ± 0.0074	0.7234 ± 0.0045
Sensitivity	0.5418 ± 0.0135	0.5359 ± 0.0082	0.5412 ± 0.0113	0.5382 ± 0.0131	0.5507 ± 0.0121
Specificity	0.8874 ± 0.0071	0.8880 ± 0.0058	0.8816 ± 0.0074	0.8861 ± 0.0067	0.8962 ± 0.0040

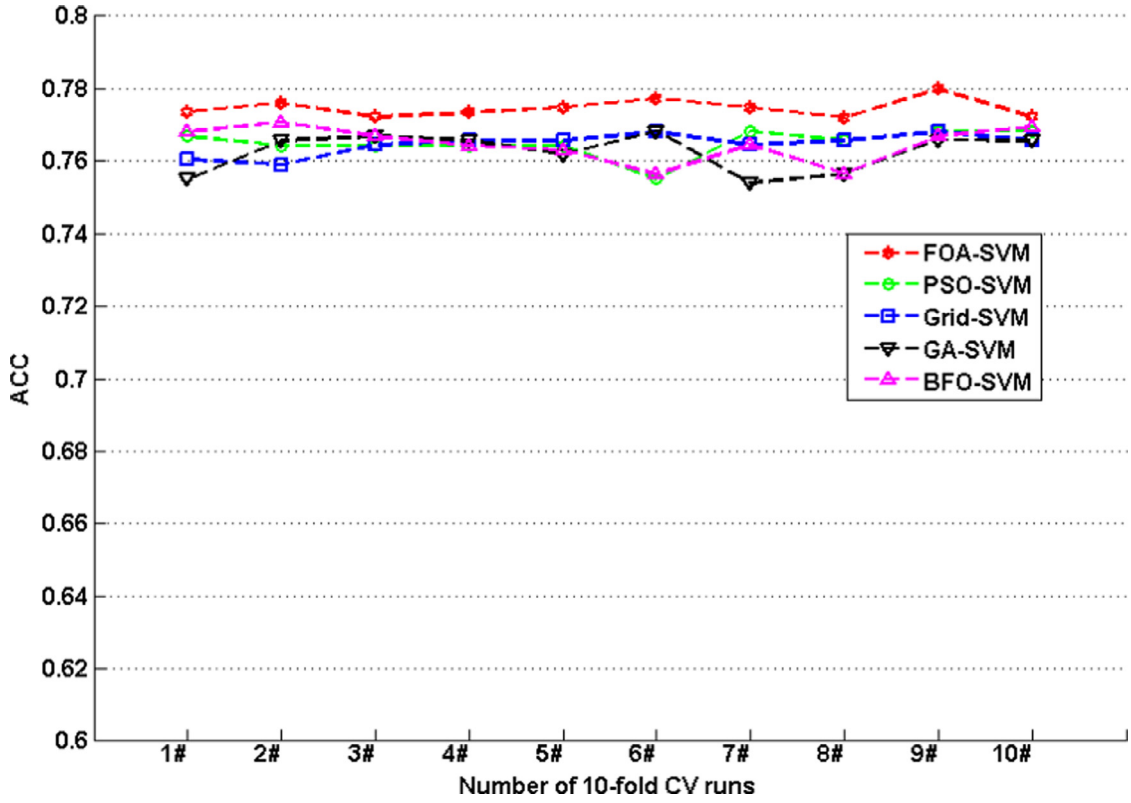

Fig. 6. The ACC obtained for each run by the five methods on the Pima dataset.

5.2. Diabetes disease diagnosis problem

The Pima dataset consists of 768 instances (500 normal, 268 diabetes), which were taken from female patients of Pima Indian heritage. The data's main focus is to predict the presence or absence of diabetes among Pima-Indian women. Each sample has eight features, including the (1) number of times pregnant, (2) plasma glucose concentration a 2 h in an oral glucose tolerance test, (3) diastolic blood pressure, (4) triceps skin fold thickness, (5) 2 h serum insulin, (6) body mass index, (7) diabetes pedigree function, and (8) age.

Table 6 illustrates the detailed classification results of various methods in terms of the ACC, AUC, sensitivity, and specificity on the Pima dataset. It can be seen in Table 6 that, from among the five methods, the FOA-SVM method performs the best with a relatively small standard deviation with average results of 77.46% ACC, 72.34% AUC, 55.07% sensitivity, and 89.62% specificity. The detailed comparison results of 10 runs of the 10-fold CV for the five methods in terms of ACC are displayed in Fig. 6. The same phenomenon can be observed as in the Wisconsin dataset; FOA-SVM beats the other four competitors over the whole 10 runs.

As was conducted for the Wisconsin dataset, a paired *t*-test was performed among the five methods on the Pima dataset in order to verify the significance of the proposed method. A p-value of less than 0.05 was considered to be statistically significant in the experiment. From Table 7, it can be seen that FOA-SVM almost achieves significantly better results than the other four methods in terms of the ACC, AUC, sensitivity, and specificity at the prescribed statistical significance level of 5%.

Table 7
Paired *t*-test results of FOA-SVM and the other four methods in terms of ACC, AUC, sensitivity, and specificity on the Pima dataset.

Metrics	*t*-value (significance)			
	PSO-SVM	Grid-SVM	GA-SVM	BFO-SVM
ACC	**5.985(0.000)**	**9.425(0.000)**	**7.372(0.000)**	**5.596(0.000)**
AUC	**5.566(0.000)**	**5.928(0.000)**	**6.898(0.000)**	**6.167(0.000)**
Sensitivity	1.965(0.081)	**2.707(0.024)**	**2.934(0.017)**	**3.476(0.007)**
Specificity	**3.971(0.003)**	**3.580(0.006)**	**5.690(0.000)**	**3.285(0.009)**

Positive values indicate that the *i*th classifier has achieved a higher performance than the *j*th one. Significant values less than 0.05 are shown in bold.

The five methods' CPU times on the Pima dataset were also recorded during the experiment. Fig. 7 displays the five methods' computational times in seconds during one run of the10-fold CV on the Pima dataset. As shown in Fig. 7, FOA-SVM requires an average of almost 170 s to complete the training and prediction procedure in each fold for the diabetes disease diagnosis. From among the four metaheuristic methods, FOA-SVM requires the least

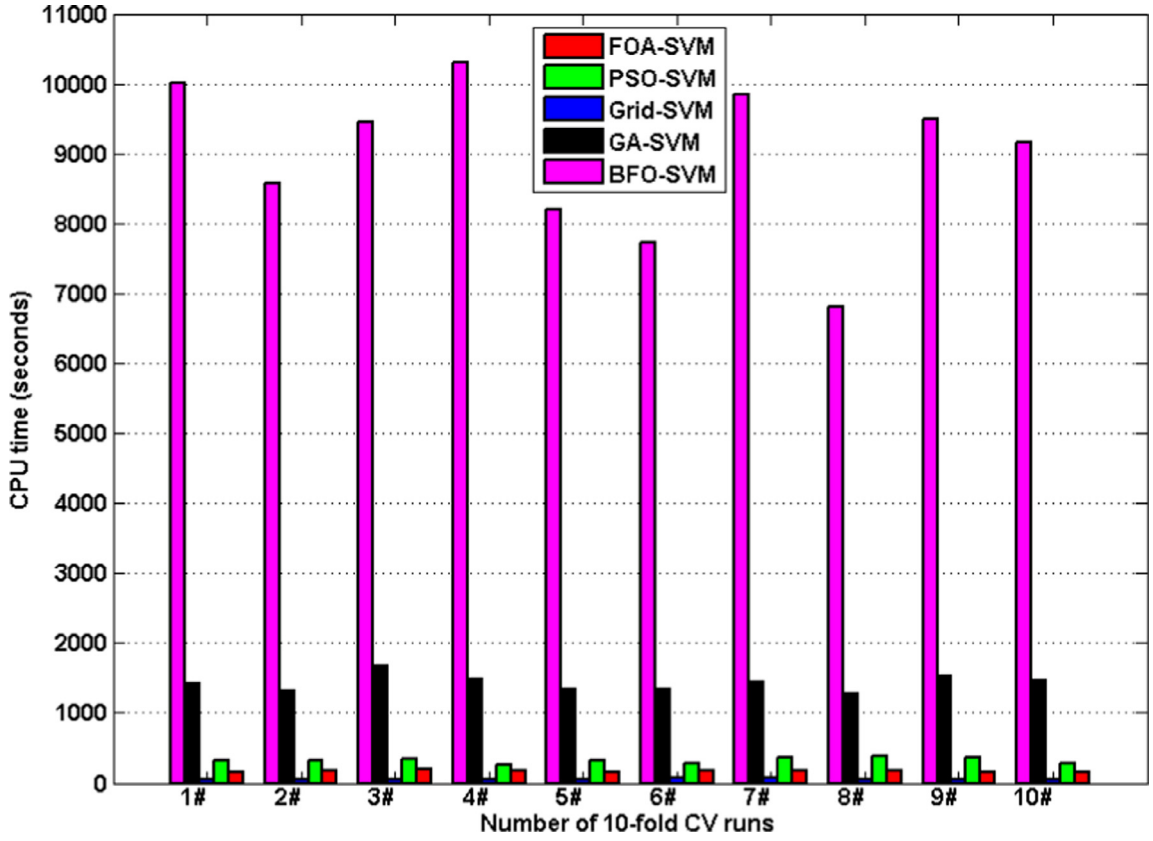

Fig. 7. Comparison results of the five methods in terms of CPU time on the Pima dataset.

Table 8
The 22 attributes of the Parkinson dataset.

Label	Attribute	Description
F1	MDVP:Fo (Hz)	Average vocal fundamental frequency
F2	MDVP:Fhi (Hz)	Maximum vocal fundamental frequency
F3	MDVP:Flo (Hz)	Minimum vocal fundamental frequency
F4	MDVP:Jitter (%)	Several measures of variation in fundamental frequency
F5	MDVP:Jitter (Abs)	
F6	MDVP:RAP	
F7	MDVP:PPQ	
F8	Jitter:DDP	
F9	MDVP:Shimmer	Several measures of variation in amplitude
F10	MDVP:Shimmer (dB)	
F11	Shimmer:APQ3	
F12	Shimmer:APQ5	
F13	MDVP:APQ	
F14	Shimmer:DDA	
F15	NHR	Two measures of ratio of noise to tonal components in the voice
F16	HNR	
F17	RPDE	Two nonlinear dynamical complexity measures
F18	D2	
F19	DFA	Signal fractal scaling exponent
F20	Spread1	Three nonlinear measures of fundamental frequency variation
F21	Spread2	
F22	PPE	

CPU time to complete the diabetes diagnosis task. Again, BFO-SVM is the most time consuming as it requires a CPU load of nearly 53 times that of FOA-SVM. Compared to the Wisconsin dataset, all five methods consume considerably more computational resources. From among the five methods, Grid-SVM is again superior to the other four metaheuristic-based methods in this case.

5.3. Parkinson's disease diagnosis problem

The Parkinson dataset has 195 samples and 22 attributes. The objective of this dataset is to discriminate healthy people from those with Parkinson's disease. In the medical experiment, various biomedical voice measurements were recorded for 23 patients with Parkinson's disease and eight healthy controls. The time since diagnoses ranged from 0 to 28 years, and the ages of the subjects ranged from 46 to 85 years, with a mean age of 65.8. Each subject provided an average of six phonations of the vowel (yielding 195 samples in total), and each is 36 s in length. All 22 features are presented in Table 8 along with descriptions.

Table 9 illustrates the detailed classification results of the various methods in terms of the ACC, AUC, sensitivity, and specificity

Table 9
The detailed classification results of the various methods on the Parkinson dataset.

Metrics	Methods				
	PSO-SVM	Grid-SVM	GA-SVM	BFO-SVM	FOA-SVM
ACC	0.9627 ± 0.0038	0.9624 ± 0.0025	0.9564 ± 0.0054	0.9557 ± 0.0056	0.9690 ± 0.0010
AUC	0.9626 ± 0.0042	0.9602 ± 0.0032	0.9529 ± 0.0063	0.9609 ± 0.0044	0.9687 ± 0.0009
Sensitivity	0.9624 ± 0.0070	0.9662 ± 0.0024	0.9627 ± 0.0050	0.9457 ± 0.0082	0.9686 ± 0.0014
Specificity	0.9659 ± 0.0068	0.9545 ± 0.0060	0.9432 ± 0.0099	0.9761 ± 0.0059	0.9689 ± 0.0018

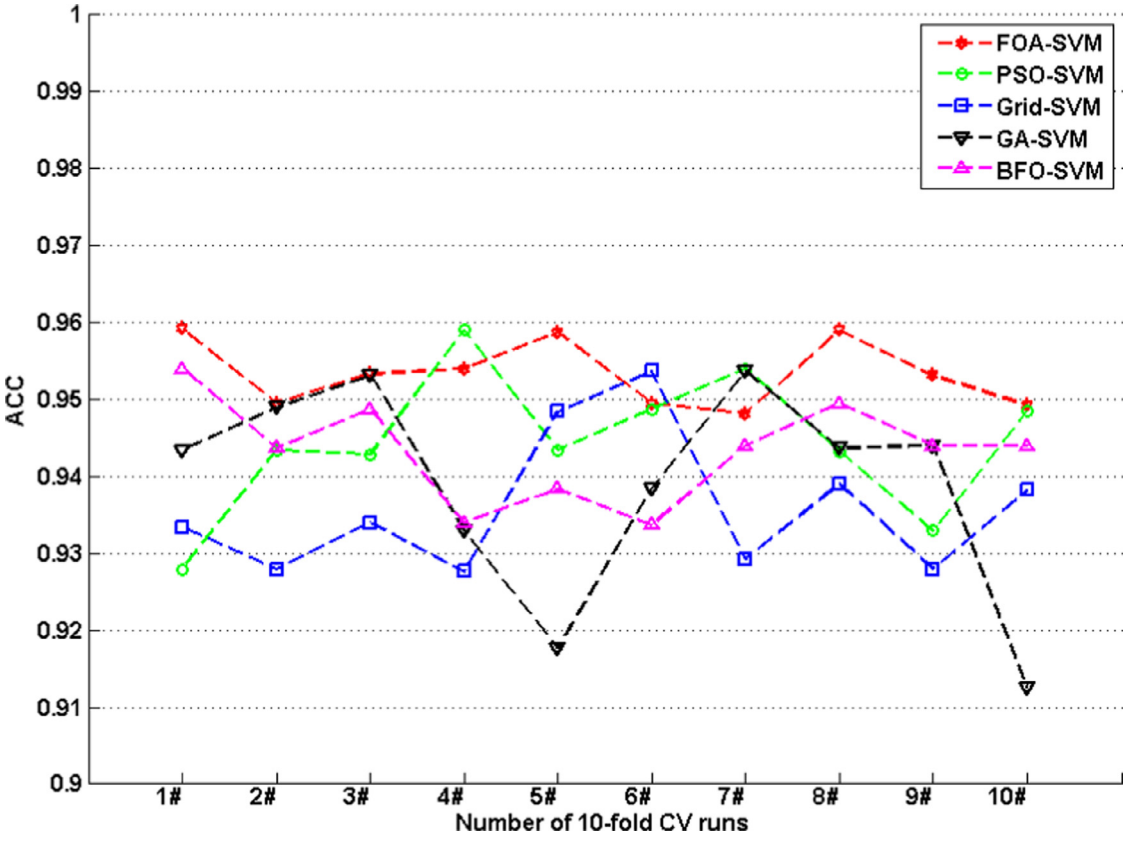

Fig. 8. The ACC obtained for each run by the five methods on the Parkinson dataset.

on the Parkinson dataset. It can be seen in Table 9 that, from among the five methods, the FOA-SVM method performs the best with the smallest standard deviation and with average results of 77.46% ACC, 72.34% AUC, 55.07% sensitivity, and 89.62% specificity. The detailed comparison results of 10 runs of the 10-fold CV for the five methods in terms of ACC are displayed in Fig. 8. It can be observed that FOA-SVM performs better than the other four competitors in seven runs of the 10-fold CV with the most consistent and stable results.

Again, a paired *t*-test was performed among the five methods on the Parkinson dataset so as to validate the significance of the proposed method. A p-value of less than 0.05 was considered statistically significant in the experiment. From Table 10, it can be seen that FOA-SVM achieves significantly better results than the other four methods in terms of ACC and sensitivity. Regarding the AUC results, FOA-SVM achieves a better performance than the other four methods; however, it is not a significantly better performance. Moreover, it should be noted that FOA-SVM performs slightly worse than PSO-SVM, GA-SVM, and BFO-SVM on the specificity metric.

The five methods' CPU times on the Parkinson dataset were also recorded during the experiment. Fig. 9 displays the five methods' computational times in seconds during one run of the 10-fold CV on the Parkinson dataset. As shown in Fig. 9, FOA-SVM requires an average of almost 17 s to complete the training and prediction procedure in each fold for the Parkinson's disease diagnosis. From among the four metaheuristic methods, FOA-SVM requires less computational load to complete the Parkinson's disease diagnosis task than both PSO-SVM and BFO-SVM. The same phenomenon has been observed for the BFO-SVM method, and it is again the most time consuming method; it requires nearly twice the CPU load of FOA-SVM. However, its CPU load is nearly the same as that of PSO-SVM, which was not observed in either the Wisconsin

Table 10
Paired *t*-test results of FOA-SVM and the other four methods in terms of ACC, AUC, sensitivity, and specificity on the Parkinson dataset.

Metrics	*t*-value (significance)			
	PSO-SVM	Grid-SVM	GA-SVM	BFO-SVM
ACC	**2.397(0.040)**	**5.846(0.000)**	3.023(0.014)	**4.953(0.001)**
AUC	0.703(0.500)	**3.902(0.004)**	1.204(0.259)	0.886(0.399)
Sensitivity	**6.047(0.000)**	**8.009(0.000)**	**7.023(0.000)**	**6.873(0.000)**
Specificity	−.741(0.478)	2.192(0.056)	−.317(0.759)	−1.069(0.313)

Positive values indicate that the *i*th classifier has achieved a higher performance than the *j*th one. Significant values less than 0.05 are shown in bold.

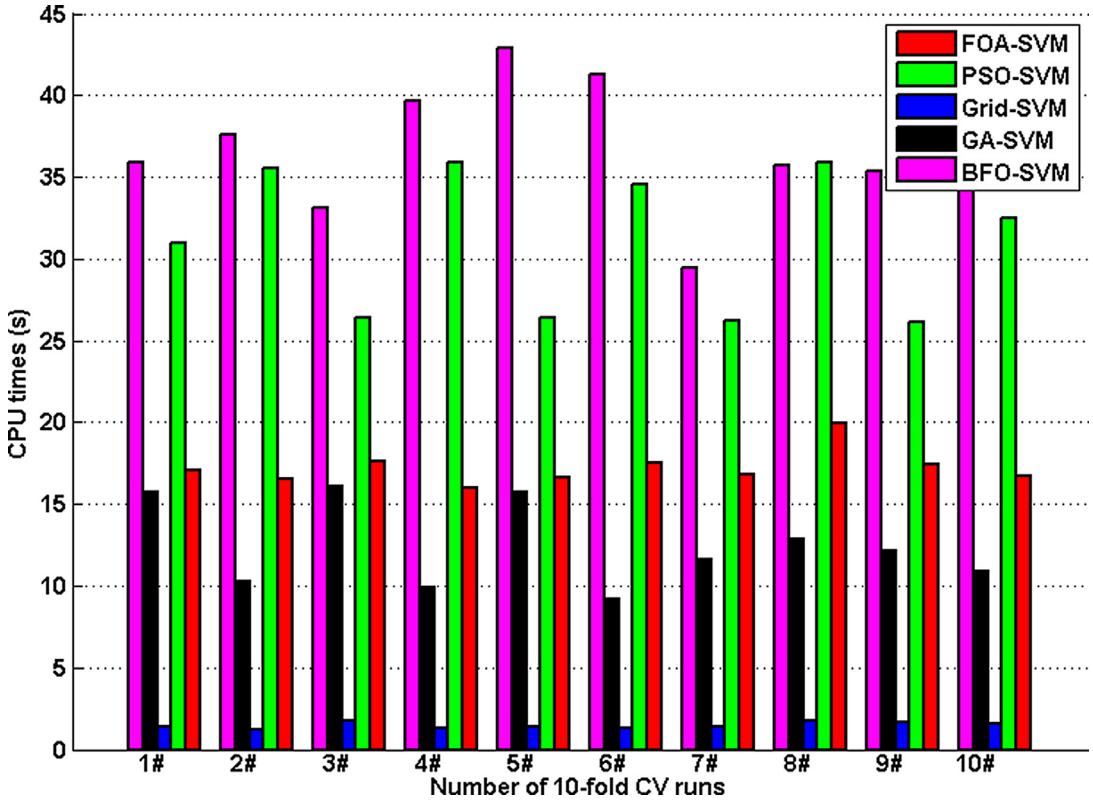

Fig. 9. Comparison results of the five methods in terms of CPU time on the Parkinson dataset.

Table 11
The five attributes of the thyroid dataset.

Attributes	Description
F_1	T3-resin uptake test (a percentage).
F_2	Total serum thyroxin as measured by the isotopic displacement method.
F_3	Total serum triiodothyronine as measured by radioimmunoassay.
F_4	Basal thyroid-stimulating hormone (TSH) as measured by radioimmunoassay.
F_5	Maximal absolute difference of TSH value after injection of 200 mg of thyrotropin-releasing hormone as compared to the basal value.

Table 12
The detailed classification results of the various methods on the thyroid dataset.

Metrics	Methods				
	PSO-SVM	Grid-SVM	GA-SVM	BFO-SVM	FOA-SVM
ACC	0.9526 ± 0.0080	0.9499 ± 0.0092	0.9594 ± 0.0106	0.9440 ± 0.0082	0.9638 ± 0.0062

dataset or the diabetes dataset. Again, as expected, Grid-SVM beats the other four metaheuristic-based methods in this case.

5.4. Thyroid disease diagnosis problem

The thyroid dataset consists of thyroid disease measurements and contains three classes and 215 samples. Five tests are used to attempt to predict whether a patient's thyroid can be classified as euthyroidism, hypothyroidism, or hyperthyroidism. The diagnosis (the class label) was based on a complete medical record, including anamnesis, scans, etc. The class distribution is as follows:

- Class 1: euthyroidism (150)
- Class 2: hyper (35)
- Class 3: hypo (30)

All samples have five features, and all are continuous. The entire dataset is detailed in Table 11.

Table 12 illustrates the detailed classification results of the various methods in terms of ACC on the Parkinson dataset. It can be seen from Table 12 that, from among the five methods, the FOA-SVM method performs better with the smallest standard deviation and with average results of 96.38% ACC. The detailed comparison results of 10 runs of the 10-fold CV for the five methods in terms of ACC are displayed in Fig. 10. It can be observed that FOA-SVM performs better than PSO-SVM, Grid-SVM, and BFO-SVM in all ten runs of the 10-fold CV, and it achieves comparable results to those of GA-SVM.

Again, a paired *t*-test was performed among the five methods on the thyroid dataset in order to validate the significance of the proposed method. A p-value of less than 0.05 was considered statistically significant in the experiment. From Table 13, it can be seen that FOA-SVM achieves significantly better results than PSO-SVM, Grid-SVM, and BFO-SVM. Furthermore, FOA-SVM achieves better ACC results than GA-SVM; however, it is not a significant difference.

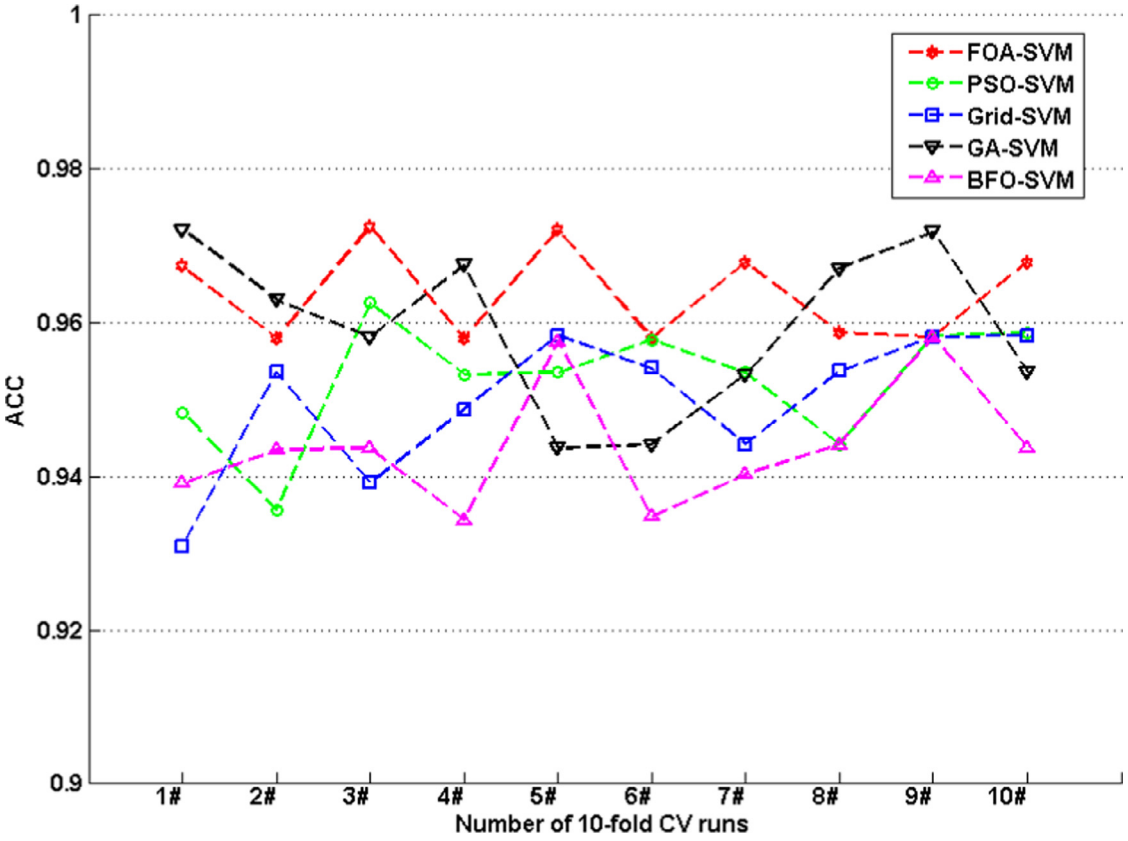

Fig. 10. The ACC obtained for each run by the five methods on the thyroid dataset.

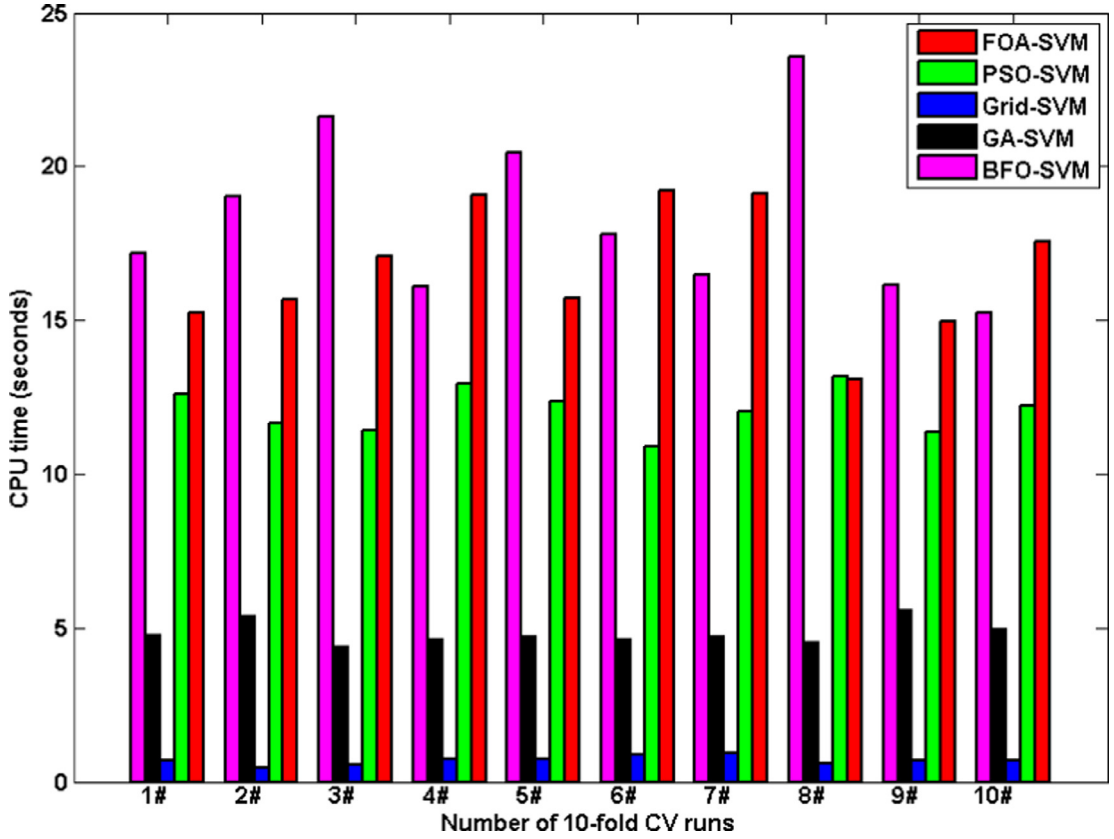

Fig. 11. Comparison results of the five methods in terms of CPU time on the thyroid dataset.

Table 13
Paired *t*-test results of FOA-SVM and the other four methods in terms of ACC on the thyroid dataset.

Metrics	*t*-value (significance)			
	PSO-SVM	Grid-SVM	GA-SVM	BFO-SVM
ACC	**4.512(0.001)**	**3.434(0.007)**	0.978(0.354)	**6.981(0.000)**

Positive values indicate that the *i*th classifier has achieved a higher performance than the *j*th one. Significant values less than 0.05 are shown in bold.

The five methods' CPU times on the thyroid dataset were also recorded during the experiment. Fig. 11 displays the five methods' computational times in seconds during one run of the 10-fold CV on the thyroid dataset. As shown in Fig. 11, FOA-SVM requires an average of almost 17 s to complete the training and prediction procedure in each fold for the thyroid disease diagnosis. It is noteworthy that FOA-SVM, PSO-SVM, and BFO-SVM all require very similar computational loads to conduct the thyroid disease diagnosis problem. Furthermore, BFO-SVM requires almost the same CPU time as the other metaheuristic-based methods. This may be because the data is relatively small as compared with the other datasets. In this setting, GA-SVM requires the least amount of time to complete the task. Again, as expected, Grid-SVM beats the other four metaheuristic-based methods in this case.

From the above analysis, it can be seen that FOA-SVM is the most suitable method; it produces an excellent performance and only requires a moderate computational cost for solving medical data classification problems. FOA-SVM's superior performance reveals that the FOA has the ability to obtain much more appropriate parameter settings with high generalization capability for the SVM classifier on a specific disease diagnosis problem. FOA-SVM's advantageous performance may be due to the FOA's unique olfactory and visual senses. In this study, there were only four medical datasets involved in the experiment; thus, more data sets are needed to further verify the effectiveness of the proposed methodology. Moreover, both the parameter setting and feature selection are crucial for improving the SVM performance since they influence one another. Therefore, in the future, we plan to simultaneously implement feature selection and parameter optimization. Moreover, as future work, we also intend to explore other promising metaheuristics to optimize SVM parameters, such as Grey Wolf Optimizer [39], Ant Lion Optimizer [40] and Multi-Verse Optimizer [41]. Furthermore, we plan to utilize the high performance technique to further improve the performance of the proposed method.

6. Conclusions

FOA-SVM is an effective and computationally efficient approach for solving medical data classification problems. This paper's novelty lies in the proposed FOA-based approach, which aims at maximizing the generalization capability of the SVM classifier by exploring the new swarm intelligence technique for optimal parameter tuning for medical data classification. The empirical experiments demonstrated the evident superiority of the proposed FOA-SVM over four other competitive counterparts in various evaluation metrics and especially the CPU time cost. This indicates that the proposed FOA-SVM method can be utilized as a valuable alternative clinical solution for medical decision support.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (NSFC) under Grant nos. of 61303113, 61133011, 61572226, 61373053, 61373166 and 61402336. This research is also funded by the Science and Technology Plan Project of Wenzhou of China under Grant no. G20140048, Jilin Province Natural Science Foundation under Grant no. 20150101052JC, the open project program of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University under Grant no. 93K172013K01, the open project program of Wenzhou University Laboratory under Grant no. 15SK26A.

References

[1] B.E. Boser, I.M. Guyon, V.N. Vapnik, A Training Algorithm for Optimal Margin Classifiers, ACM, New York, NY, USA, 1992.
[2] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[3] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[4] K.S. Shin, T.S. Lee, H. Kim, An application of support vector machines in bankruptcy prediction model, Expert Syst. Appl. 28 (1) (2005) 127–135.
[5] H.L. Chen, et al., A new hybrid method based on local fisher discriminant analysis and support vector machines for hepatitis disease diagnosis, Expert Syst. Appl. 38 (9) (2011) 11796–11803.
[6] H.L. Chen, et al., A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis, Expert Syst. Appl. 38 (7) (2011) 9014–9022.
[7] E.D. Übeyli, Comparison of different classification algorithms in clinical decision-making, Expert Syst. 24 (1) (2007) 17–31.
[8] H.L. Chen, et al., A three-stage expert system based on support vector machines for thyroid disease diagnosis, J. Med. Syst. 36 (3) (2012) 1953–1963.
[9] H.-L. Chen, et al., Support Vector Machine Based Diagnostic System for Breast Cancer Using Swarm Intelligence, J. Med. Syst. 36 (4) (2012) 2505–2519.
[10] M.F. Akay, Support vector machines combined with feature selection for breast cancer diagnosis, Expert Syst. Appl. 36 (2) (2009) 3240–3247.
[11] I. Maglogiannis, E. Zafiropoulos, I. Anagnostopoulos, An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers, Appl. Intell. 30 (1) (2007) 24–36.
[12] S. Keerthi, C. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural Comput. 15 (7) (2003) 1667–1689.
[13] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines. 2001, Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm (accessed 14.12.15).
[14] C. Gold, P. Sollich, Model selection for support vector machine classification, Neurocomputing 55 (1-2) (2003) 221–249.
[15] S. Keerthi, Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms, IEEE Trans. Neural Netw. 13 (5) (2002) 1225–1229.
[16] O. Chapelle, et al., Choosing multiple parameters for support vector machines, Mach. Learn. 46 (1) (2002) 131–159.
[17] C.-L. Huang, C.-J. Wang, A GA-based feature selection and parameters optimization for support vector machines, Expert Syst. Appl. 31 (2) (2006) 231–240.
[18] H.L. Chen, et al., Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy, Appl. Math. Comput. 239 (2014) 180–197.
[19] Y.-W. Fu, et al., A new evolutionary support vector machine with application to parkinson's disease diagnosis, in: Y. Tan, Y. Shi, C.C. Coello (Eds.), Advances in Swarm Intelligence, Springer International Publishing, 2014, pp. 42–49.
[20] W.-T. Pan, A new fruit fly optimization algorithm: Taking the financial distress model as an example, Knowl. Based Syst. 26 (2012) 69–74.
[21] H. Li, et al., Annual electric load forecasting by a least squares support vector machine with a fruit fly optimization algorithm, Energies 5 (11) (2012) 4430–4445.
[22] P.-W. Chen, et al., Using fruit fly optimization algorithm optimized grey model neural network to perform satisfaction analysis for e-business service, Appl. Math. Inf. Sci. 7 (2) (2013) 459–465.
[23] H.-Z. Li, et al., A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm, Knowl. Based Syst. 37 (2013) 378–387.
[24] D. Shan, G. Cao, H. Dong, LGMS-FOA: An improved fruit fly optimization algorithm for solving optimization problems, Math. Probl. Eng. 2013 (2013). Article ID 108768, 9 pages.
[25] L. Wang, X.-l. Zheng, S.-Y. Wang, A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem, Knowl. Based Syst. 48 (2013) 17–23.
[26] W.-T. Pan, Mixed modified fruit fly optimization algorithm with general regression neural network to build oil and gold prices forecasting model, Kybernetes 43 (7) (2014) 1053–1063.
[27] Q.-K. Pan, et al., An improved fruit fly optimization algorithm for continuous function optimization problems, Knowl. Based Syst. 62 (2014) 69–83.
[28] X. Yuan, et al., On a novel multi-swarm fruit fly optimization algorithm and its application, Appl. Math. Comput. 233 (2014) 260–271.
[29] J.-Q. Li, et al., Solving the steelmaking casting problem using an effective fruit fly optimisation algorithm, Knowl. Based Syst. 72 (2014) 28–36.
[30] X.-L. Zheng, L. Wang, S.-Y. Wang, A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem, Knowl. Based Syst. 57 (2014) 95–103.

[31] L. Wang, Y. Shi, S. Liu, An improved fruit fly optimization algorithm and its application to joint replenishment problems, Expert Syst. Appl. 42 (9) (2015) 4310–4323.
[32] W. Wang, X. Liu, Melt index prediction by least squares support vector machines with an adaptive mutation fruit fly optimization algorithm, Chemom. Intell. Lab. Syst. 141 (2015) 79–87.
[33] C. Xiao, K. Hao, Y. Ding, An improved fruit fly optimization algorithm inspired from cell communication mechanism, Math. Probl. Eng. 2015 (2015). Article ID 492195, 15 pages
[34] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines: and Other Kernel-Based Learning Methods, Cambridge Univ Press, 2000.
[35] B. Schölkopf, C.J.C. Burges, A.J. Smola, Advances in Kernel Methods: Support Vector Learning, The MIT Press, 1998.
[36] H.L. Chen, et al., A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method, Knowl. Based Syst. 24 (8) (2011) 1348–1359.
[37] C.C. Chang, C.J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.
[38] T. Fawcett, ROC graphs: Notes and practical considerations for researchers, Mach. Learn. 31 (2004) 1–38.
[39] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[40] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[41] S. Mirjalili, S. Mirjalili, A. Hatamlou, Multi-verse optimizer: A nature-inspired algorithm for global optimization, Neural Comput. Appl. (2015) 1–19, doi:10.1007/s00521-015-1870-7.