Support vector machine based diagnostic system for breast cancer using swarm intelligence

Hui-Ling Chen[a,b] Bo Yang[a,b] Gang Wang[a,b] Su-Jing Wang[a,b] Jie Liu[a,b] Da-You Liu[a,b]*

[a](College of Computer Science and Technology, Jilin University, Changchun 130012, China)
[b](Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)

Abstract:

Breast cancer is becoming a leading cause of death among women in the whole world, meanwhile, it is confirmed that the early detection and accurate diagnosis of this disease can ensure a long survival of the patients. In this paper, a swarm intelligence technique based support vector machine classifier (PSO_SVM) is proposed for breast cancer diagnosis. In the proposed PSO-SVM, the issue of model selection and feature selection in SVM is simultaneously solved under particle swarm (PSO optimization) framework. A weighted function is adopted to design the objective function of PSO, which takes into account the average accuracy rates of SVM (ACC), the number of support vectors (SVs) and the selected features simultaneously. Furthermore, time varying acceleration coefficients (TVAC) and inertia weight (TVIW) are employed to efficiently control the local and global search in PSO algorithm.

The effectiveness of PSO-SVM has been rigorously evaluated against the Wisconsin Breast Cancer Dataset (WBCD), which is commonly used among researchers who use machine learning methods for breast cancer diagnosis. The proposed system is compared with the grid search method with feature selection by *F*-score. The experimental results demonstrate that the proposed approach not only obtains much more appropriate model parameters and discriminative feature subset, but also needs smaller set of SVs for training, giving high predictive accuracy. In addition, Compared to the existing methods in previous studies, the proposed system can also be regarded as a promising success with the excellent classification accuracy of 99.3% via 10-fold cross validation (CV) analysis. Moreover, a combination of five informative features is identified, which might provide important insights to the nature of the breast cancer disease and give an important clue for the physicians to take a closer attention. We believe the promising result can ensure that the physicians make very accurate diagnostic decision in clinical breast cancer diagnosis.

Keywords: Support vector machines; Particle swarm optimization; breast cancer diagnosis; Feature selection; swarm intelligence

1 Introduction

Worldwide, breast cancer is the second most common type of cancer after lung cancer (10.4% of all cancer incidence, both sexes counted) and the fifth most common cause of cancer death. Moreover, it is now by far the most common cancer amongst women, with an incidence rate more than twice that of colorectal cancer and cervical cancer and about three times that of lung cancer. However breast cancer mortality worldwide is just 25% greater than that of lung cancer in women (http://www.wnba.com/silverstars/community/breast_health_awareness.html, last accessed February 2011). Research is under way to learn more and scientists are making great progress in detecting the disease at an early stage. Early diagnosis requires an accurate and reliable diagnosis procedure that allows physicians to distinguish benign breast tumors from malignant ones [1]. Thus, expert systems and artificial intelligent techniques are increasingly introduced to help improve the diagnostic capability. With the help of these automatic diagnostic systems, the possible errors experts made in the course of diagnosis can be avoided, and the medical data can be examined in shorter time and more detailed as well.

A great deal of artificial intelligent techniques has been investigated to diagnose the disease of breast cancer with high classification accuracies. Among these, in [2] presented by Quinlan, C4.5 decision tree method was used and the obtained classification accuracy was 94.74%. In [3], Hamiton etal.obtained 94.99% accuracy with the RIAC method using 10-fold CV. In [4], Ster and Dobnikar obtained 96.8% with linear discreet analysis (LDA) method. The accuracy obtained by Bennett and Blue [5] who used SVM was 97.2%, by Nauck and Kruse [6] was 95.06% with neuro-fuzzy techniques and by Pena-Rayes and Sipper [7] was 97.36% using fuzzy-GA method. In [8] presented by Setiono, the classification was based on a feed forward neural network rule extraction algorithm, the reported accuracy was 98.10%. In. [9] presented by Goodman et al, three different methods, optimized learning vector quantization (LVQ), big LVQ, and artificial immune recognition system (AIRS), were applied and the obtained accuracies were 96.7%, 96.8%, and 97.2%, respectively. In [10] presented by Abonyi et al., an accuracy of 95.57% was obtained with the application of supervised fuzzy clustering technique. In [11], Ubeyli presented the mixture experts (ME) network structure for breast cancer diagnosis, the obtained total classification accuracy was 98.85%. In [12] presented by Sahan et al., a new hybrid method based on fuzzy-artificial immune system and k-nn algorithm (Fuzzy-AIS-knn) was used and the obtained accuracy was 99.14%. In [13] presented by Ubeyli, multilayer perceptron neural network, four different methods, combined neural network, probabilistic neural network, recurrent neural network and SVM were used respectively, highest classification accuracy of 97.36% was achieved by SVM. In [14] presented by Polat and Gunes, least square SVM (LS-SVM) was used and 98.53% accuracy was obtained. Akay [15] reached 99.51% classification accuracy using a SVM-based method combined with F-score method. In [16], Ubeyli developed adaptive neuro-fuzzy inference system (ANFIS) for breast cancer detection, and the total accuracy of 99.08% was obtained. In [17] presented by Karabatak and Cevdet, the method combined with association rules and neural networks (AR+NN) were used and classification accuracy of 97.4% was obtained. In [18], Huang et al. reached 98.83%, 97.51% classification accuracy using sequential backward selection (SBS) algorithm integrating with BPNN and LM (SBS-BPLM), BPNN and PSO (SBS-BPPSO), respectively. In [19], Marcano-Cedeño et al. used the Artificial Metaplasticity Multilayer Perceptron (AMMLP) algorithm and the classification accuracy of 99.26% was obtained. In [20], Fan et al. reached 98.90% classification accuracy using case-based reasoning approach combined with fuzzy decision tree (CBFDT). In [21], Chen et al proposed a rough set based support vector machine classifier (RS_SVM) for breast cancer diagnosis, the highest and average classification accuracy of 100% and 96.87% were achieved respectively.

As can be seen from theses works, SVM has been used to diagnose the breast cancer and achieved the highest classification accuracy among the available artificial intelligent methods in literature. However, in our opinion despite its great potential, the SVM approach has not received the attention it deserves in the breast cancer diagnosis literature as compared to other research fields. SVM as a relatively new machine learning technique was first introduced by Vapnik [22]. It seeks to minimize the upper bound of the generalization error based on the structural risk minimization (SRM) principal that is known to have high generalization performance. Another key feature of SVM is that training SVM is equivalent to solving a linear constrained quadratic programming problem. Thus it is unlikely to be trapped in the local optimum [23-24]. Thanks to its good properties, it has found its application in a wide variety of fields including handwritten digit recognition [25] , face detection in images [26], text categorization [27], and so forth. When using SVM for tackling practical problems, there are two issues have to be handled. On the one hand, the appropriate kernel parameter setting plays a significant role in designing an effective SVM model. The first parameter, penalty parameter *C*, determines the trade-off between the fitting error minimization and model complexity. The second parameter, gamma (γ or d) of the kernel function, defines the non-linear mapping from the input space to some high-dimensional feature space. On the other hand, choosing the optimal input feature subset also influence the performance of the SVM model in great part. Feature selection is an important issue in building classification systems, which refers to choosing subset of attributes from the set of original attributes. Its key purpose is to identify the significant features, eliminate the irrelevant of dispensable features and build a good learning model. The benefits of feature selection are twofold: it considerably decreases the computation time of the induction algorithm and increases the accuracy of the resulting model as well [28]. Both of them are crucial because the feature selection influences the appropriate kernel parameters and vice versa [29], this suggested that they should be dealt with simultaneously.

Grid search [30] is one of the most common methods to determine appropriate values for *C* and γ , which can lead to the highest classification accuracy rate in an interval through setting appropriate values for the upper and lower bounds and the jumping interval in the search. However, this approach is a local search method which is vulnerable to local optimum. Additionally, setting an appropriate search interval is not an easy job. It will be costly in time and computational resources if the search interval is set to too large, otherwise, if the search interval is set too small will render the unsatisfactory outcome. Apart from grid search, the gradient descent method [31] is also used to obtain the optimal parameters of SVM. Nevertheless, one disadvantage of gradient descent algorithm is that this algorithm is sensitive to initial parameters. When initial parameters are far from the optimal solution, it will be easily converged to local optimum. In this study, we attempted to tackle the parameter optimization and feature selection problem for SVM simultaneously using a new learning scheme based on swarm intelligence. As a new swarm intelligence technique, Particle swarm optimization (PSO), has been found to be a promising technique for real world optimization problems [32] due to its strong global search capability. Compared to genetic algorithms (GA), PSO takes less time for each function evaluation as it does not use many of GA operators like mutation, crossover and selection operator, and most important of all PSO is very easy to implement. In this study, both continuous and discrete PSO algorithms are employed to construct the efficient SVM classifier. The continuous PSO algorithm is employed to evolve the optimal parameters, while the discrete PSO algorithm is used as a feature selection vehicle to identify the most discriminant features.

The main objective of this study is to exploit the maximum generalization capability of SVM and apply it to the breast cancer diagnosis to distinguish benign breast tumor from malignant one. The proposed breast cancer diagnostic system consists of two stages. In the first stage, the continuous PSO algorithm is employed to evolve the optimal kernel parameters, and the discrete PSO algorithm is utilized as a feature selection tool to obtain a compact and discriminative feature subset, which improves the accuracy and robustness of the subsequent classifiers. In the second stage, tumor classification is performed based on the optimal SVM prediction model. In the proposed PSO-SVM system, we take into account the ACC of SVM, the number of SVs and the number of features simultaneously in designing the objective function to exploit the maximum generalization capability of SVM. The three sub-objectives are summed into one single objective function by linearly weighting. In order to further balance the local and global search in PSO, the adaptive control parameters (including TVAC and TVIW) are introduced. The effectiveness of the proposed PSO-SVM diagnostic system is examined in terms of classification accuracy on the WBCD database taken from UCI machine learning repository. Compared with the grid search based method, our proposed PSO-SVM can not only obtain much more appropriate model parameters and discriminative feature subset, but also generate fewer numbers of SVs, giving high predictive accuracy. If it is compared with classification results of other methods in literature, our result can be regarded as a promising success.

The remainder of this paper is organized as follows. Section 2 offers brief background knowledge on SVM. The description of the PSO is presented in Section 3. In section 4 the detailed implementation of the PSO-SVM diagnostic system is presented. Section 5 describes the experimental design. The experimental results and discussion of the proposed approach are presented in Section 6. Finally, Conclusions and recommendations for future work are summarized in Section 7.

2 Support vector machines for classification

Support vector machine (SVM), originally developed by Vapnik [22, 33], is based on the Vapnik-Chervonenkis (VC) theory and structural risk minimization (SRM) principle [22, 34]. It tries to find the tradeoff between minimizing the training set error and maximizing the margin, in order to achieve the best generalization ability and remains resistant to over fitting. Additionally, one major advantage of the SVM is the use of convex quadratic programming, which provides only global minima hence avoid being trapped in local minima. For more details, one can refer to [22, 24], which give a complete description of the SVM theory.

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

$$\text{Minimize } \mathrm{g}(\mathrm{w},\xi) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \ , \tag{1}$$

$$\text{Subject to: } \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \ . \tag{2}$$

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

$$\text{Maximize } \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \ , \tag{3}$$

$$\text{Subject to: } \ 0 \leq \alpha_i \leq C \ , \sum_{i=1}^{n}\alpha_i y_i = 0 \ . \tag{4}$$

Obviously, it is a quadratic optimization problem (QP) with linear constraints. From Karush Kuhn–Tucker (KKT) condition, we know:

$$\alpha_i\left(y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1\right) = 0 \ . \tag{5}$$

If $\alpha_i > 0$, the corresponding data points are called SVs. Hence the solution takes the form as follow:

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \ , \tag{6}$$

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

$$g(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i^T\mathbf{x} + b)\right) . \tag{7}$$

In most cases, the two classes can not be linearly separated. In order to make the linear learning machine 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:

$$g(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i \phi(\mathbf{x}_i)^T\phi(\mathbf{x}) + b)\right) \tag{8}$$

where $\mathbf{x}_i^T\mathbf{x}$ in the input space is represented as the form of $\phi(\mathbf{x}_i)^T\phi(\mathbf{x})$ in the feature space. The functional form of the mapping $\phi(\mathbf{x}_i)$ does not need to be known 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:

$$g(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{9}$$

In general, any positive semi-definite functions that satisfy the Mercer's condition can be as kernel functions [35]. There exit many kernel functions that could be used by the SVM. For example, the linear kernel is defined as the dot product of two feature vectors in some expanded feature space. Furthermore, two most widely used kernels in SVM are the polynomial kernel and the Gaussian kernel (or Radial-Basis function, RBF), which are respectively defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p \tag{10}$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2) \tag{11}$$

where *p* is the polynomial order, and γ is the predefined parameter controlling the width of the Gaussian kernel.

It has been proved that proper model parameters setting can improve the SVM classification accuracy [36]. Values of parameters in SVM have to be carefully chosen in advance. These parameters include the followings: (1) regularization parameter C, which determines the tradeoff cost between minimizing the training error and the complexity of the model; (2) parameter gamma (γ or p) of the kernel function which defines the non-linear mapping from the input space to some high-dimensional feature space; (3) a kernel function used in SVM, which constructs a non-linear decision hyperplane in an input space. This investigation is going to consider the Gaussian kernel to find out the optimal parameter values of RBF kernel function (i.e., C and γ). Other kernel parameters can also be tackled in the same way by using our developed method.

3 Particle swarm optimization (PSO)

Particle swarm optimization (PSO) is inspired by the social behavior of organisms such as bird flocking and fish schooling, which was first developed by Kennedy and Eberhart [37] [38]. The algorithm seeks to explore the search space by a population of individuals or particles. Each particle represents a single solution with a velocity which is dynamically adjusted according to its own experience and that of its neighboring companions. And the population of particles is updated based on each particle's previous best performance and the best particle in the population. In this way, PSO combines local search with global search for balancing the exploration and exploitation. Considering a d-dimensional search space, the *i*th particle is represented as $\vec{X}_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,d})$, and its according velocity is represented as $\vec{V}_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,d})$ .The best previous position of the *i*th particle that gives the best fitness value is represented as $\vec{P}_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,d})$. The best particle among all the particles in the population is represented as $\vec{P}_g = (p_{g,1}, p_{g,2}, \cdots, p_{g,d})$. In every iteration, each particle updates its position and velocity according to the two best values.

3.1 PSO with inertia weight

In order to reduce the dependence of the search process on the hard bounds of the velocity, the concept of an inertia weight *w* was introduced in the PSO algorithm [39]. The velocity and position are updated as follows:

$$v_{i,j}^{n+1} = w \times v_{i,j}^n + c_1 \times r_1 \; (p_{i,j}^n - x_{i,j}^n) + c_2 \times r_2 \; (p_{g,j}^n - x_{i,j}^n), \tag{12}$$

$$x_{i,j}^{n+1} = x_{i,j}^n + v_{i,j}^{n+1}, \; j = 1, 2, \cdots, d. \tag{13}$$

where c_1 and c_2 are acceleration coefficients, which define the magnitude of the influences on the particles velocity in the directions of the personal and the global optima, respectively. To better balance the search space between the global exploration and local exploitation, Time-Varying Acceleration Coefficients (TVAC) have been introduced in [40]. This concept will be adopted in this study to ensure the better search for the solutions. The core idea of TVAC is that c_1 decreases from its initial value of c_{1i} to c_{1f}, while c_2 increases from c_{2i} to c_{2f} using the following equations as in [40]. TVAC can be mathematically represented as follows:

$$c_1 = (c_{1f} - c_{1i}) \frac{t}{t_{\max}} + c_{1i}, \tag{14}$$

$$c_2 = (c_{2f} - c_{2i}) \frac{t}{t_{\max}} + c_{2i}. \tag{15}$$

where c_{1f}, c_{1i}, c_{2f} and c_{2i} are constants, t is the current iteration of the algorithm and $t_{\max}$ is the maximum number of iterations.

In addition, r_1 and r_2 in Eq. (12) are random numbers, generated uniformly in the range [0, 1]. The velocity $v_{i,j}$ is restricted to the range $[-v_{\max}, v_{\max}]$, in order to prevent the particles from flying out of the solution space. generally, $v_{\max}$ is suggested to set to be 10-20% of the dynamic range of the variable in each dimension [41]. *w* is the inertia weight, which is used to balance the global exploration and local exploitation, a large inertia weight facilitates the global search, while a small inertia weight facilitates the local search. In order to reduce the weight over the iterations allowing the algorithm to exploit some specific areas, the inertia weight *w* is updated according to the following equation:

$$w = w_{\min} + (w_{\max} - w_{\min}) \frac{(t_{\max} - t)}{t_{\max}} \tag{16}$$

where $w_{\max}$, $w_{\min}$ are the predefined maximum and minimum values of the inertia weight *w,* t is the current iteration of the algorithm and $t_{\max}$ is the maximum number of iterations. Usually the value of w is varied between 0.9 and 0.4. Eq. (16) is also known as the time varying inertia weight (TVIW) [39], which has been shown to significantly improve the performance of PSO [42], since TVIW makes PSO have more global search ability at the beginning of the run and have more local search ability near the end of the run.

3.2 Discrete Binary PSO

PSO was originally introduced as an optimization technique for continuous space. in order to extend the application to discrete spaces, Kennedy and Eberhart [43] proposed a discrete binary version of PSO where a particle moves in a state space restricted to zero and one on each dimension, in terms of the changes in probabilities that a bit will be in one state or the other. If the velocity is high it is more likely to choose 1, and lower values favor choosing 0. A sigmoid function is applied to transform the velocity from continuous space to probability space:

$$sig(v_{i,j}) = \frac{1}{1+\exp(-v_{i,j})}, \ j = 1, 2, \cdots, d. \tag{17}$$

The velocity update Eq. (12) keeps unchanged except that $x_{i,j}, p_{i,j}$ and $p_{g,j} \in \{0,1\}$, and in order to ensure that bit can transfer between 1 and 0 with a positive probability $v_{\max}$ was introduced to limit $v_{i,j}$. In practice, $v_{\max}$ is often set as 4. The new particle position is updated using the following rule:

$$x_{i,j}^{n+1} = \begin{cases} 1, & if \ rnd < sig(v_{i,j}) \\ 0, & if \ rnd \ge sig(v_{i,j}) \end{cases}, \ j = 1, 2, \cdots, d. \tag{18}$$

where $sig(v_{i,j})$ is calculated according to Eq. (17), and rnd is a uniform random number in the range [0, 1].

4 Proposed PSO-SVM classification system

We have proposed the PSO-SVM classification system for breast cancer diagnosis, which combines the parameter optimization with the feature selection together, in order to acquire the highest classification accuracy. The proposed system consists of two stages. In the first stage, both the SVM parameters optimization and the feature selection are dynamically conducted by implementing PSO algorithm, the pseudo-code of this stage, termed as Inner_Parameter_Optimization, is given bellow:

Pseudo-code for the Inner_Parameter_Optimization procedure

```
Begin
    Randomly initialize particle swarm;
    While(number of generations or the stopping criterion is not met)
        For i = 1 to number of particles
            Train SVM model with the randomly chosen features by using 5-fold CV;
            Evaluate fitness of particle swarm;
            /* save the global optimal fitness as gfit, personal optimal fitness as pfit,
          global optimal particle as gbest and personal optimal particle as pbest.*/
          /* Update the velocity of continuous and discrete dimensions*/
            $v_{i,j}^{n+1} = w \times v_{i,j}^{n} + c_1 \times r_1 \, (p_{i,j}^{n} - x_{i,j}^{n}) + c_2 \times r_2 \, (p_{g,j}^{n} - x_{i,j}^{n})$
          /* $c_1 = (c_{1f} - c_{1i})\frac{t}{t_{\max}} + c_{1i}$ , $c_2 = (c_{2f} - c_{2i})\frac{t}{t_{\max}} + c_{2i}$ , $w = w_{\min} + (w_{\max} - w_{\min})\frac{(t_{\max} - t)}{t_{\max}}$ */
            If $(v_{i,j}^{n+1} \notin [V_{\min}, V_{\max}])$ $v_{i,j}^{n+1} = \max(\min(V_{\max}, v_{i,j}^{n+1}), V_{\min})$ Endif;
          /*Update the position of continuous dimensions*/
            $x_{i,j}^{n+1} = x_{i,j}^{n} + v_{i,j}^{n+1}, \ j = 1, 2, \cdots, d.$
            If $(x_{i,j}^{n+1} \notin [X_{\min}, X_{\max}])$ $x_{i,j}^{n+1} = \max(\min(X_{\max}, x_{i,j}^{n+1}), X_{\min})$ Endif;
            /*Update the position of discrete dimensions*/
            $sig(v_{i,j}) = \frac{1}{1 + \exp(-v_{i,j})}, \ j = 1, 2, \cdots, d.$
            If $(rnd < sig(v_{i,j}))$ $x_{i,j}^{n+1} = 1$ Else $x_{i,j}^{n+1} = 0$ Endif;
          /* Update the personal optimal fitness (pfit) and personal optimal position (pbest) by
            comparing the current fitness value (cfit) with the pfit stored in the memory.*/
             If (cfit > pfit)
                pfit = cfit;
                pbest = current position;
             Endif;
          Endfor;
            /*Get the maximum value (maxlocal) and index from the swarm of local
            fitness(local_fit)and update the global optimal fitness (gfit) and global optimal
            particle (gbest) by comparing the gfit with the optimal pfit from the whole
            population */
             [maxlocal,index] = max(local_fit);
            If (maxlocal > gfit)
               gfit = maxlocal;
               gbest = local_fit(index);
            Endif;
             Next generation until stopping criterion;
        Endwhile
          /*Get the best values of parameters (bestc and bestg) and the optimal feature
            subset(optimal_fsset)from gbest*/
           bestc = gbest (1);
           bestg = gbest (2);
           optimal_fsset = gbest(3:n+2);
           Return bestc, bestg, optimal_fsset;
  End.
```

In the second stage, SVM model performs the classification tasks using these optimal values and selected features via 10-fold CV technique, the pseudo-code of this stage, termed as Outer_Performance_Estimation, is given bellow:

Pseudo-code for the Outer_Performance_Estimation procedure

```
/*performance estimation by using k-fold CV where k = 10*/
Begin
  For j = 1:k
     Training set = k-1 subsets;
     Testing set = remaining subset;
     Train the SVM classifier on the training set using the parameters and feature subsets obtained
     from Inner_Parameter_Optimization ();
     Test it on the testing set;
  Endfor;
```

```
    Return the average classification accuracy rates of SVM over $j$ testing set;
End.
```

PSO-SVM takes into consideration three fitness values for parameter optimization and feature selection. The first one is the ACC of SVM, the second one is the number of SVs and the last one is the number of selected features. In this way, the PSO-SVM can not only achieve the high classification accuracy, but also obtain the good capability of generalization. The PSO-SVM classification system for breast cancer diagnosis is constructed through the following main steps:

- Step 1: Encode the particle with n+2 dimensions. The first two dimensions are C and γ which are continuous values. The remaining n dimensions is Boolean features mask, which is represented by discrete value, 1 indicates the feature is selected, and 0 represents the feature is discarded.
- Step 2: Initialize the individuals of the population with random numbers. Meanwhile, specify the PSO parameters including the lower and upper bounds of the velocity, the size of particles, the number of iterations, etc.
- Step 3: Train the SVM model with the selected feature subset in Step 2.
- Step 4: The particle with high classification accuracy and the small number of selected features can produce a high fitness value. In addition, the particle with smaller number of SVs can achieve higher classification accuracy, since the number of SVs is proportional to the generalization error of the SVM classifier [22]. Thus in this study, we take all of them into account to design the fitness function. The fitness value is calculated according to the following multi-objective function:

$$\begin{cases} f_1 = avgacc = \dfrac{\Sigma_{i=1}^{K} Test\_Accuracy_i}{K} \\ f_2 = (1 - \dfrac{nsv}{m}) \\ f_3 = (1 - \dfrac{\sum_{j=1}^{n} ft_i}{n}) \\ f = \alpha \times f_1 + \beta \times f_2 + \lambda \times f_3 \end{cases} \tag{19}$$

where variable $avgacc$ in the first objective function f_1 represents the average testing accuracy achieved by the SVM classifier via 10-fold CV, where K=10. Noted that here the 10-fold CV is employed to do the model selection that is different from the outer loop of 10-fold CV, which is used to do the performance estimation. nsv and m in the second function f_2 indicate the number of SVs and training data, respectively. In the third objective function f_3, ft_i is the value of feature mask ('1' represents that feature is selected and '0' indicates that feature is discarded), n is the total number of features. The weighted summation of the three sub-objective functions is selected as the final objective function. In f, variable α is the weight for SVM classification accuracy, β indicates the weight for the number of SVs, and λ represents the weight for the selected features. The weight can be adjusted to a proper value depends on the importance of the sub-objective function. Eq. (19) means that average accuracy rates, the number of SVs and feature subset length have different significance for the classification performance. According to our preliminary experiments, the classification performance is more depend on the average accuracy rates and the number of SVs than the number of selected features, so the value α and β are selected as bigger than that of λ. Generally, the weight is set to be constant value. After many tests, we found that the linearly increasing/decreasing function can further improve the classification performance over most datasets. Thus, we define the value of weight as the linearly increasing/decreasing function varying along with the iterations. They are defined as $\alpha = (\alpha_1 - \alpha_2)\dfrac{t}{t_{\max}} + \alpha_2$, $\beta = (\beta_1 - \beta_2)\dfrac{t}{t_{\max}} + \beta_2$, $\lambda = (\lambda_1 - \lambda_2)\dfrac{t}{t_{\max}} + \lambda_2$, respectively.

Upon finishing the computation of the fitness value, we save the global optimal fitness as *gfit*, personal optimal fitness as *pfit*, global optimal particle as *gbest* and personal optimal particle as *pbest.*

- Step 5: Increase the number of iteration.
- Step 6: Increase the number of population. Update the position and velocity of C and γ in each particle according to Eqs. (12-13), and the features in each particle according to Eqs. (12,17-18).
- Step 7: Train the SVM model with the selected feature subset in Step 6 and calculate the fitness value of each particle according to Eq. (19).
- Step 8: Update the personal optimal fitness (*pfit*) and personal optimal position (*pbest*) by comparing the current fitness value with the *pfit* stored in the memory. If the current fitness is dominated by the *pfit* stored in the memory, then keep the *pfit* and *pbest* in the memory; otherwise, replace the *pfit* and *pbest* in the memory with the current fitness value and particle position.
- Step 9: If the size of the population is reached, then go to Step 10. Otherwise, go to Step 6.
- Step 10: Update the global optimal fitness (*gfit*) and global optimal particle (*gbest*) by comparing the *gfit* with the optimal *pfit* from the whole population, If the current optimal *pfit* is dominated by the *gfit* stored in the memory, then keep the g*fit* and *gbest* in the memory; otherwise, replace the *gfit* and *gbest* in the memory with the current optimal *pfit* and the optimal *pbest* from the whole population.
- Step 11: If the stopping criteria are satisfied, then go to Step 12. Otherwise, go to Step 5. The termination criteria are that the iteration number reaches the maximum number of iterations or the value of *gfit* does not improve after 100 consecutive iterations.
- Step 12: Get the optimal C , γ and the feature subset from the best particle (*gbest*).

5 Experimental design

5.1 Data Description

In this study, we have performed our conduction on the Wisconsin Breast Cancer Dataset (WBCD) taken from UCI machine learning repository (UCI Repository of Machine Learning Databases). The dataset contains 699 instances taken from needle aspirates from patients’ breasts, of which 458 cases belong to benign class and the remaining 241 cases belong to malignant class. It should be noted that there are 16 instances which have missing values, in this study all the missing values are replaced by the mean of the attributes. Each record in the database has nine attributes. These nine attributes were found to differ significantly between benign and malignant samples. The nine attributes listed in Table 1 are graded 1-10, with 10 being the most abnormal state. The class attribute was represented as 2 for benign and 4 for malignant cases.

Table 1 The detail of the nine attributes of breast cancer data

Label	**Attribute**	**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

5.2 Experimental setup

The proposed PSO-SVM diagnostic system was implemented using MATLAB platform. For SVM, LIBSVM implementation was utilized, which is originally developed by Chang and Lin

[44]. We implemented the PSO algorithm from scratch. The empirical experiment was conducted on Intel Quad-Core Xeon 5130 CPU (2.0 GHz) with 4GB of RAM.

Normalization is employed to avoid feature values in greater numerical ranges dominating those in smaller numerical ranges, as well as to avoid the numerical difficulties during the calculation [30]. Usually, the data could be normalized by scaling them into the interval of [-1, 1] according to the Eq. (20), where *x* is the original value, x' is the scaled value, max_a is the maximum value of feature *a*, and min_a is the minimum value of feature *a*.

$$x' = (\frac{x - min_a}{max_a - min_a})*2-1 \tag{20}$$

In order to guarantee the valid results, the *k*-fold CV presented by Salzberg [45] was used to evaluate the classification accuracy. This study set *k* as 10, i.e., the data was divided into ten subsets. Each time, one of the ten subsets is used as the test set and the other nine subsets are put together to form a training set. Then the average error across all ten trials is computed. The advantage of this method is that all of the test sets are independent and the reliability of the results could be improved. We attempted to design our experiment using two loops. The inner loop is used to determine the optimal parameters and best feature subset. The outer loop is used for estimating the performance of the SVM classifier. In order to keep the same proportion of benign and malignant cases of each set as that of the entire data set, here a stratified 10-fold CV is employed as the outer loop and a stratified 9-fold CV is used for the inner loop. It is referred to as the nested stratified 10-fold CV, which is also used in [46] for the microarray gene data analysis. It is worth noting that the test data used in the test stage is isolated from the training data used in the training stage, namely the best parameter pair (C ,γ) and feature subset are obtained from the training dataset, and then the test dataset is used to obtain the average CV accuracy in the testing stage, thus preventing it from obtaining the over-estimate the accuracy.

The detail parameter setting for PSO-SVM is set as follows. The number of the iterations and particles is set to 200 and 30, respectively. The searching ranges for C and γ are as follows: $C \in [2 \,\hat{}\, (-5), 2 \,\hat{}\, (15)]$ and $\gamma \in [2 \,\hat{}\, (-15), 2 \,\hat{}\, (5)]$. $v_{\max}$ are set about 10% of the dynamic range of the variable on each dimension for the continuous type of dimensions (as suggested in [41]). For the discrete type particle for feature selection, $[-v_{\max}, v_{\max}]$ is set as [−6, 6]. As suggested in [40], $c_{1i} = 2.5, c_{1f} = 0.5, c_{2i} = 0.5, c_{2f} = 2.5$ .According to our preliminary experiment, $w_{\max}$ and $w_{\min}$ are set to 0.9 and 0.4, and the parameters of α, β and λ are taken as $\alpha_1 = 0.3, \alpha_2 = 0.6$, $\beta_1 = 0.7$, $\beta_2 = 0.3$, $\lambda_1 = 0$, $\lambda_2 = 0.1$, respectively.

5.3 Measure for Performance Evaluation

Sensitivity, specificity, total classification accuracy (ACC) and the area under the Receiver Operating Characteristic curve (AUC) were used to test the performance of the proposed PSO-SVM model. Before defining these measures, we introduced the concept of confusion matrix, which is presented in Table 2. Where TP is the number of true positives, which means that some cases with 'positive' class is correctly classified as positive; FN, the number of false negatives, which means that some cases with the 'positive' class is classified as negative; TN, the number of true negatives, which means that some cases with the 'negative' class is correctly classified as negative; and FP, the number of false positives, which means that some cases with the 'negative' class is classified as positive.

Table 2 Confusion matrix for breast cancer diagnosis

	Predicted positive	Predicted negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

According to the confusion matrix, ACC, sensitivity and specificity are defined as follows:

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{21}$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \tag{22}$$

$$Specificity = \frac{TN}{FP + TN} \times 100\% \tag{23}$$

The receiver operating characteristic (ROC) curve is a graphical display that gives the measure of the predictive accuracy of a logistic model. The curve displays the true positive rate and false positive rate. AUC is the area under the ROC curve, which is one of the best methods for comparing classifiers in two-class problems.

5.4 Comparative study

In this study, we attempt to conduct a performance comparison between the proposed PSO-SVM system and the grid search method with feature selection by the *F*-score [47], termed FS-SVM. In FS-SVM method, the importance of each feature is measured by *F*-score, and the SVM parameters are optimized by grid search algorithm. As mentioned before, the grid search is a common method for searching for the best C and γ. Fig. 1 shows the procedure of the SVM training using grid search. The searching space of parameters C and γ are set to $C = \{2^{-5}, 2^{-3}, \cdots, 2^{15}\}$ and $\gamma = \{2^{-15}, 2^{-13}, \cdots, 2^{1}\}$, respectively. There will be $11 \times 9 = 99$ parameter combinations of (C, γ) are tried and the one with the best CV accuracy is chosen as the parameter values. Here, 5-fold CV is adopted to conduct the parameter optimization. Then the best parameter pair (C, γ) is used to create the model for training. After obtain the predictor model, the prediction is conducted on each testing set accordingly. *F*-score is a fundamental and simple method that measures the distinction between two classes with real values. Given the training vectors if the number of positive and negative instances are n^{+} and n^{-}, respectively, then the *F*-score of the i^{th} feature is explained as follows [47]:

$$F(i) = \frac{(\overline{x}_i^{(+)} - \overline{x}_i)^2 + (\overline{x}_i^{(-)} - \overline{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)} - \overline{x}_i^{(+)})^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}(x_{k,i}^{(-)} - \overline{x}_i^{(-)})^2} \tag{24}$$

where $\overline{x}_i$, $\overline{x}_i^{(+)}$, $\overline{x}_i^{(-)}$ are the averages of the i^{th} feature of the whole, positive, and negative datasets, respectively. $x_{k,i}^{(+)}$ is the i^{th} feature of the k^{th} positive instance, and $x_{k,i}^{(-)}$ is the i^{th} feature of the k^{th} negative instance. The numerator shows the discrimination between the positive and negative sets, and the denominator defines the one within each of the two sets. The larger the *F*-score is, the more likely this feature is more discriminative [47]. In this study, we select features with high *F*-score according to following procedure. Firstly, the data was divided into ten subsets via stratified *k*-fold CV. Each time, one of the *k* subsets is used as the testing set and the other *k*-1 subsets are used as the training set. Then the average classification accuracy across all *k* trials is computed. The specific procedure is as follows [47]:

Step 1. Calculate *F*-score of every feature.

Step 2. Sort *F*-score, and Set possible number of features according to the following formula:

$$f = [n / 2^{j}],\ j \in \{0, 1, 2, \cdots, l\}\ . \tag{25}$$

where l is an integer with $n / 2^{l} \geq 1$.

Step 3. For each f (threshold), do the following:

a) Drop features with *F*-score below this threshold.

b) Randomly split the training set into X_{train} and X_{test} using 5-fold CV. Do the following step for each fold:

c) Let X_{train} be the new training data. Perform SVM training procedure (as shown in Fig. 1) to obtain a predictor; use the predictor to predict X_{test}.

d) Calculate the average classification accuracy.

Step 4. Choose the threshold with the highest average classification accuracy.

Step 5. Drop features with *F*-score below the selected threshold. Rerun SVM training procedure (as shown in Fig. 1) on the remaining testing set and Measure the classification accuracy on each testing set.

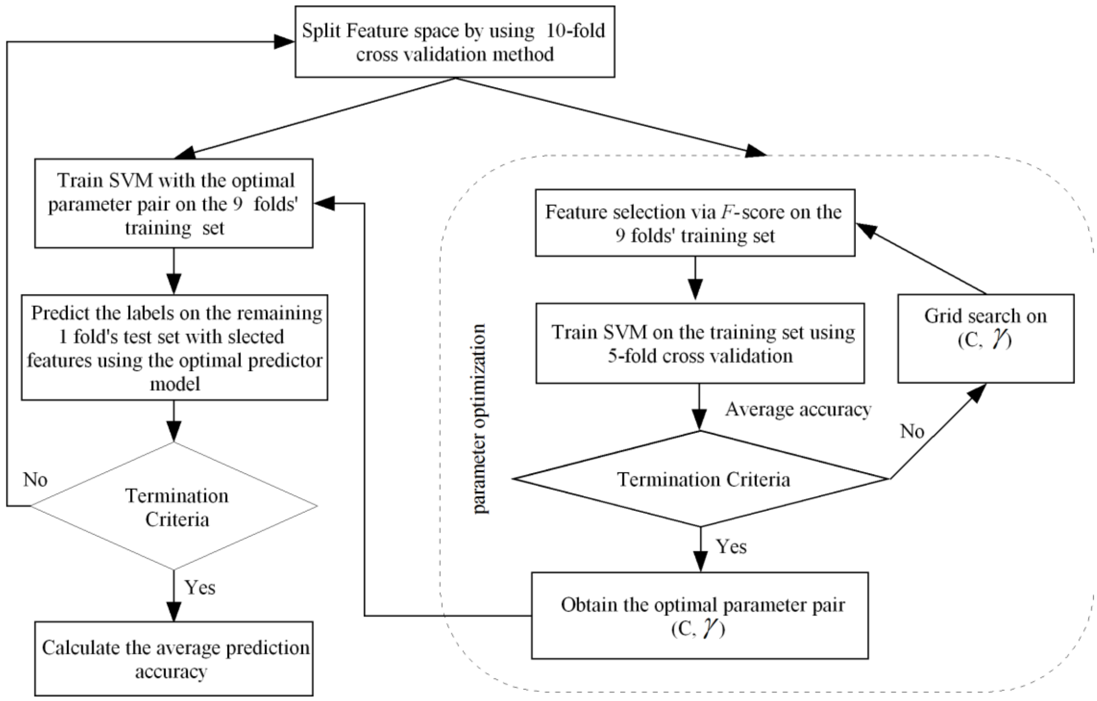
Split Feature space by using 10-fold cross validation method
Train SVM with the optimal parameter pair on the 9 folds' training set
Predict the labels on the remaining 1 fold's test set with slected features using the optimal predictor model
No
Termination Criteria
Yes
Calculate the average prediction accuracy
Feature selection via F-score on the 9 folds' training set
Train SVM on the training set using 5-fold cross validation
Grid search on (C, γ)
Average accuracy
No
Termination Criteria
Yes
Obtain the optimal parameter pair (C, γ)
parameter optimization

Fig.1 The procedure of SVM training using grid search method

6 Experimental results and discussions

To evaluate the effectiveness of the proposed PSO-SVM system for breast cancer, we conduct experiments on the WBCD database. Table 3 shows classification accuracy rate, number of selected features, and optimal pairs of (C ,γ) for each fold using PSO-SVM and FS-SVM. It can be observed that, the average accuracy rates achieved by the developed PSO-SVM system are much better than that of FS-SVM. The average classification accuracy rate of PSO-SVM is 99.28%, while the average classification accuracy rate of FS-SVM is 96.99%. Furthermore, the number of the optimal feature subset obtained by the PSO-SVM is much smaller than that of FS-SVM. For the PSO-SVM method, the average number of the selected features via 10-fold CV is about 5, while the average number of the selected features is about 9 for the FS-SVM method. The detailed results of the sensitivity, specificity and AUC for WBCD database are listed in Table 4. It can be clearly seen that the average values of sensitivity, specificity and AUC achieved by PSO-SVM are much better than those of FS-SVM. Moreover, it is interesting to see that the standard deviation for the acquired average classification rates, sensitivity and AUC by PSO-SVM is much smaller than that of FS-SVM, which indicates consistency and stability of the proposed system.

Table 3 Classification accuracy rate, number of selected features, and optimal parameter settings for WBCD database using PSO-SVM and FS-SVM

Fold	PMOPSO-SVM				FS-SVM			
	C	γ	ACC(%)	#features	C	γ	ACC(%)	#features
#1	23992.103	0.001	98.571	4	1.000	0.031	94.285	9
#2	29804.780	10.507	100.000	5	0.250	0.001	98.571	9
#3	25769.629	15.904	100.000	5	1.000	0.001	98.571	9
#4	30374.823	0.050	98.550	6	4.000	0.000	95.714	9
#5	10161.191	13.454	100.000	5	1.000	0.007	95.714	9
#6	10505.758	0.001	98.571	4	0.250	0.007	97.142	9
#7	15879.893	12.110	98.571	5	0.250	0.001	95.714	9
#8	8335.270	0.001	100.000	5	1.000	0.001	98.571	9
#9	10743.829	8.948	98.571	6	16.000	0.001	97.142	9
#10	2330.583	12.156	100.000	5	1.000	0.001	98.571	9

Avg.	16789.785	7.313	99.283	5.0	2.575	0.005	96.999	9.0
Dev.	9932.049	6.532	0.755	0.666	4.840	0.009	1.572	0.000

$^{\text{Avg.}}$ The average value over 10-fold cross validation.
$^{\text{Dev.}}$ The standard deviation over 10-fold cross validation.

Table 4 Sensitivity, specificity and AUC for WBCD database using PSO-SVM and FS-SVM

Fold	PMOPSO-SVM			FS-SVM		
	Sensitivity (%)	Specificity (%)	AUC (%)	Sensitivity (%)	Specificity (%)	AUC (%)
#1	100.00	97.73	98.86	86.36	97.92	92.14
#2	100.00	100.00	100.00	100.00	97.73	98.86
#3	100.00	100.00	100.00	100.00	97.73	98.86
#4	95.24	100.00	97.62	90.91	97.92	94.41
#5	100.00	100.00	100.00	90.91	97.92	94.41
#6	100.00	97.73	98.86	96.30	97.67	96.99
#7	100.00	97.73	98.86	90.91	97.92	94.41
#8	100.00	100.00	100.00	100.00	97.73	98.86
#9	100.00	97.73	98.86	96.30	97.67	96.99
#10	100.00	100.00	100.00	100.00	97.73	98.86
Avg.	99.52	99.09	99.31	95.17	97.79	96.48
Dev.	1.51	1.17	0.82	5.03	0.11	2.47

$^{\text{Avg.}}$ The average value over 10-fold cross validation.
$^{\text{Dev.}}$ The standard deviation over 10-fold cross validation.

In order to verify the effectiveness of the proposed method, a paired *t* test on the average classification accuracy rates, sensitivity, specificity and AUC is used. As shown in Table 5, the *p*-value is much smaller than the prescribed statistical significance level of 0.05. Therefore, it is evident that the proposed PSO-SVM system obtains more appropriate parameters and feature subset, performing significantly better than FS-SVM method. The better performance of the proposed method can be attributed to all features, i.e., adaptive control parameters (including TVIW and TVAC), and consideration of the three sub-objectives (ACC, number of SVs and selected features) in the objective function.

Table 5 Paired *t* test of PSO-SVM vs.FC-SVM on the WBCD database

Performance metric	PSO-SVM	FS-SVM	Paired t-test p-value
ACC (%)	99.3 ± 0.75	96.9 ± 1.57	0.0002
Sensitivity (%)	99.5 ± 1.51	95.2 ± 5.03	0.0185
Specificity (%)	99.1 ± 1.17	97.8 ± 0.11	0.0069
AUC (%)	99.3 ± 0.82	96.5 ± 2.47	0.0051

Confidence level $\alpha = 0.05$.

To explore how many features and what features were selected during the feature selection procedure, we further conducted an experiment on WBCD database to investigate the detail of the feature selection mechanism of the PSO algorithm. The selected feature in 10 folds for WBCD database is shown in Table 6. The original numbers of features of WBCD database is 9. As shown in Table 6, not all features are selected for classification after the feature selection. The average number of selected features by PSO-SVM is 5.0, and its most important features are Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Bare Nuclei and Mitoses, which can be found in the frequency of selected features of 10-fold CV as shown in Fig. 2.

Table 6 Features selected for WBCD database by PSO-SVM

Fold	Selected features

#1	Clump Thickness, Marginal Adhesion, Bland Chromatin, Normal Nucleoli
#2	Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Mitoses
#3	Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Bare Nuclei, Mitoses
#4	Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Mitoses
#5	Clump Thickness, Marginal Adhesion, Bare Nuclei, Bland Chromatin, Mitoses
#6	Clump Thickness, Uniformity of Cell Shape, Bare Nuclei, Mitoses
#7	Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Bare Nuclei, Bland Chromatin
#8	Clump Thickness, Uniformity of Cell Shape, Bare Nuclei, Normal Nucleoli, Mitoses
#9	Clump Thickness, Uniformity of Cell Shape, Marginal Adhesion, Bare Nuclei, Normal Nucleoli, Mitoses
#10	Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Bare Nuclei, Mitoses

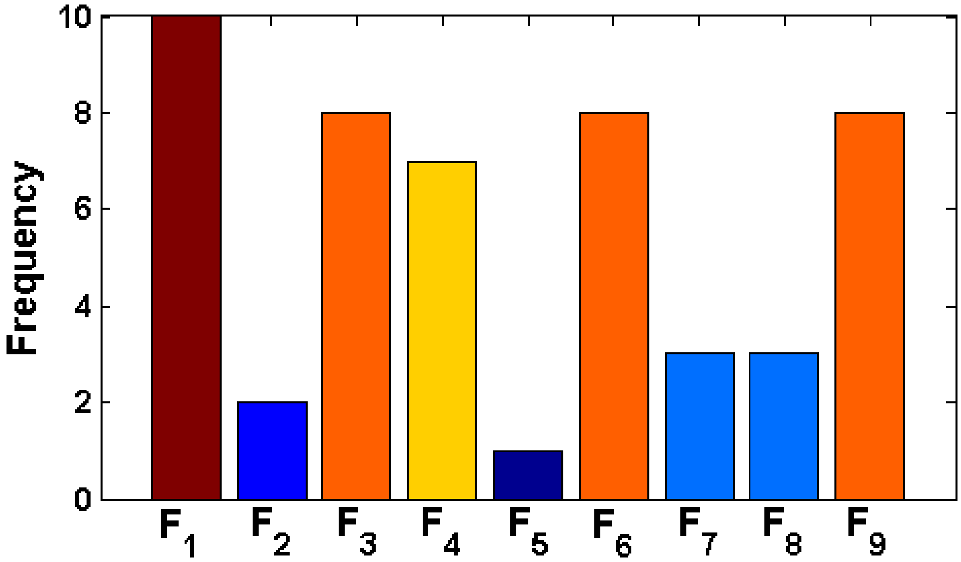

Fig.2 The frequency of selected features in 10-fold CV on the WBCD database

To give some idea about how the number of SVs influences the performance of PSO-SVM and FS-SVM, SVs were calculated in running the testing of 10-fold CV. Table 7 shows the number of SVs needed for training via the 10-fold CV by PSO-SVM and FS-SVM respectively. It can be seen that the FS-SVM method has generated much more SVs than those of PSO-SVM. Because it has been shown by Vapnik [22] that the number of SVs is proportional to the generalization error of the SVM classifier. That is the reason why our developed system has better generalization capability. Note that the number of SVs is not integers, because they are the average of the 10-fold CV.

Table 7 Number of SVs produced by PSO-SVM and FS-SVM

#SVs Fold	PSO-SVM	FS-SVM
1#	45	71
2#	41	68
3#	40	70
4#	43	71
5#	44	68
6#	43	67
7#	42	68
8#	45	68
9#	41	70
10#	39	72
Average	42.3	69.3

Analytical results reveal that the proposed PSO-SVM system has excellent generalization capability. Since grid search is a local search method which is vulnerable to local optimum, and *F*-score is just a simple way to determine important features, it does not reveal mutual information among features [47]. In addition, as shown in Table 8, the PSO-SVM is also compared with other approaches developed in the literature to show the effectiveness of our approach. From the table, it is evident that our developed PSO-SVM diagnostic system has comparable or even better classification accuracy than those achieved by other SVM classifiers on WBCD database, and obtain much better classification accuracy than those of non-SVM methods proposed in previous studies.

Table 8 Classification accuracies obtained with our method and other classifiers from literature

Study	Method	Accuracy (%)
Quinlan (1996) [2]	C4.5	94.74 (10×CV)
Hamilton et al. (1996) [3]	RIAC	94.99 (10×CV)
Ster and Dobnikar (1996) [4]	LDA	96.80 (10×CV)
Bennett and Blue (1998) [5]	SVM	97.20 (5×CV)
Nauck and Kruse (1999) [6]	NEFCLASS	95.06 (10×CV)
Penna-Reyes and Sipper(1999) [7]	Fuzzy-GA	97.36 (train: 75%-test: 25%)
Setiono (2000) [8]	Neuro-Rule	98.10 (train: 50%-test: 50%)
Goodman et al. (2002) [9]	Optimized-LVQ	96.70 (10×CV)
Goodman et al. (2002) [9]	Big-LVQ	96.80 (10×CV)
Goodman et al. (2002) [9]	AIRS	97.20 (10×CV)
Abonyi and Szeifert (2003) [10]	SFC	95.57 (10×CV)
Ubeyli (2005) [11]	ME	98.85(train: 37%-test: 63%)
Seral Şahan et al. (2007) [12]	Fuzzy-AIS-knn	99.14 (10×CV)
Ubeyli (2007) [13]	SVM	99.54 (train: 37%-test: 63%)
Polat and Günes (2007) [14]	LS-SVM	98.53 (10×CV)
Akay (2009) [15]	SVM + Fscore	99.51 (train: 80%-test: 20%)
Ubeyli (2009) [16]	ANFIS	99.08(train: 37%-test: 63%)
Karabatak and Cevdet (2009) [17]	AR + NN	97.40 (3×CV)
Huang et al. (2010) [18]	SBS-BPPSO	97.51 (10×CV)
	SBS-BPLM	98.83 (10×CV)
Marcano-Cedeno et al.(2011) [19]	AMMLP	99.26 (train: 60%-test: 40%)
Fan et al. (2011) [20]	CBFDT	98.90 (train: 75%-test: 25%)
Chen et al.(2011) [21]	RS_SVM	100.00(train:80%-test:20%) (highest)
		96.87(train: 80%-test: 20%) (average)
Our Study	PSO-SVM	99.3 (10×CV)

According to the above study, it make us be more convinced that the proposed diagnostic system can be very helpful in assisting the physicians to make the accurate diagnosis on the patients and will show great potential in the area of clinical breast cancer diagnosis. In addition, a combination of five features (i.e., ‘Clump Thickness’, ‘Uniformity of Cell Shape’, ‘Marginal Adhesion’, ‘Bare Nuclei’ and ‘Mitoses’) for classifying breast tumors is identified to be most informative. It implies that these five features are worthwhile to be taken close attention by the physicians when the final decision is made.

7 Conclusions and future work

This work has explored a new diagnostic system, PSO-SVM, for breast cancer diagnosis. The PSO-SVM diagnostic system is proposed underlying the swarm intelligent framework. The main novelty of this paper lies in the proposed PSO-based approach, which aims at maximizing the generalization capability of the SVM classifier by simultaneously tackling the kernel parameter setting and identifying the most discriminative feature subset for breast cancer diagnosis. In designing the objective function, classification accuracy, number of SVs and number of features are simultaneously taken into consideration. Moreover, the developed system is adaptive in nature attributed to adaptive control parameters (including TVIW and TVAC). There are two distinct strengths for the proposed PSO-SVM system: one is its ability to build an interpretable diagnostic model because smaller numbers of features are used. The other is its ability to build an optimal prediction model because all model parameters are optimized. Particularly, through a series of empirical experiments on the WBCD database, we show that the proposed PSO-SVM system not only maximizes the generalization performance but also selects most informative features. These indicate that the proposed PSO-SVM system can be used as a viable alternative solution to breast cancer diagnosis.

The future investigation will pay much attention to evaluating the proposed system in other medical diagnosis problems. In addition, we should note that when dealing with the practical diagnosis problems, the PSO-based system costs a lot of CPU time, thus improving the performance of our proposed system using high-performance computing techniques will be involved in our future work as well.

8 Acknowledgements

This research is supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 60873149, 60973088, 60773099 and the National High-Tech Research and Development Plan of China under Grant Nos. 2006AA10Z245, 2006AA10A309. This work is also supported by the Open Projects of Shanghai Key Laboratory of Intelligent Information Processing in Fudan University under the Grand No. IIPL-09-007, the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) and the basic scientific research fund of Chinese Ministry of Education.

References

1. Subashini, T., V. Ramalingam, and S. Palanivel, *Breast mass classification based on cytological patterns using RBFNN and SVM.* Expert Systems with Applications, 2009. **36**(3): p. 5284-5290.
2. Quinlan, J., *Improved use of continuous attributes in C4. 5.* Journal of Artificial Intelligence Research, 1996. **4**: p. 77-90
3. Hamilton, H.J., et al., *RIAC: a rule induction algorithm based on approximate classification*. 1996, University of Regina: International conference on engineering applications of neural networks.
4. Ster, B. and A. Dobnikar, *Neural networks in medical diagnosis: Comparison with other methods*. 1996: In Proceedings of the international conference on engineering applications of neural networks. p. 427-430.
5. Bennett, K. and J. Blue, *A support vector machine approach to decision trees*. 1998, in Neural Networks Proceedings,. p. 2396-2401.
6. Nauck, D. and R. Kruse, *Obtaining interpretable fuzzy classification rules from medical data.* Artificial Intelligence in Medicine, 1999. **16**(2): p. 149-169.
7. Pena-Reyes, C.A. and M. Sipper, *A fuzzy-genetic approach to breast cancer diagnosis.* Artificial Intelligence in Medicine, 1999. **17**(2): p. 131-155.
8. Setiono, R., *Generating concise and accurate classification rules for breast cancer diagnosis.* Artificial Intelligence in Medicine, 2000. **18**(3): p. 205-219.
9. Goodman, D., L. Boggess, and A. Watkins, *Artificial immune system classification of multiple-class problems.* Intelligent Engineering Systems Through Artificial Neural Networks, Fuzzy Logic, Evolutionary Programming Complex Systems and Artificial Life, 2002. **12**: p. 179–184.
10. Abonyi, J. and F. Szeifert, *Supervised fuzzy clustering for the identification of fuzzy classifiers.* Pattern Recognition Letters, 2003. **24**(14): p. 2195-2207.
11. Übeyli, E.D., *A mixture of experts network structure for breast cancer diagnosis.* Journal of Medical Systems, 2005. **29**(5): p. 569-579.
12. Sahan, S., et al., *A new hybrid method based on fuzzy-artificial immune system and k-nn algorithm for breast cancer diagnosis.* Computers in Biology and Medicine, 2007. **37**(3): p. 415-423.
13. Ubeyli, E.D., *Implementing automated diagnostic systems for breast cancer detection.* Expert Systems with Applications, 2007. **33**(4): p. 1054-1062.

14. Polat, K. and S. Gunes, *Breast cancer diagnosis using least square support vector machine.* Digital Signal Processing, 2007. **17**(4): p. 694-701.

15. Akay, M.F., *Support vector machines combined with feature selection for breast cancer diagnosis.* Expert Systems with Applications, 2009. **36**(2): p. 3240-3247.

16. Übeyli, E.D., *Adaptive neuro-fuzzy inference systems for automatic detection of breast cancer.* Journal of Medical Systems, 2009. **33**(5): p. 353-358.

17. Karabatak, M. and M.C. Ince, *An expert system for detection of breast cancer based on association rules and neural network.* Expert Systems with Applications, 2009. **36**(2, Part 2): p. 3465-3469.

18. Huang, M.-L., Y.-H. Hung, and W.-Y. Chen, *Neural Network Classifier with Entropy Based Feature Selection on Breast Cancer Diagnosis.* Journal of Medical Systems, 2010. **34**(5): p. 865-873.

19. Marcano-Cedeño, A., J. Quintanilla-Domíngueza, and D. Andina, *WBCD Breast Cancer Database Classification Applying Artificial Metaplasticity Neural Network.* Expert Systems with Applications, 2011. **http://dx.doi.org/10.1016/j.eswa.2011.01.167**.

20. Fan, C.-Y., et al., *A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification.* Applied Soft Computing, 2011. **11**(1): p. 632-644.

21. Chen, H.L., et al., *A support vector machine classifier with rough set based feature selection for breast cancer diagnosis.* Expert Systems with Applications, 2011. **38**(7): p. 9014-9022.

22. Vapnik, V.N., *The nature of statistical learning theory*. 1995: Springer, New York

23. Shawe-Taylor, J. and N. Cristianini, *Kernel methods for pattern analysis*. 2004: Cambridge Univ Pr.

24. Cristianini, N. and J. Shawe-Taylor, *An introduction to support Vector Machines: and other kernel-based learning methods*. 2000: Cambridge Univ Press.

25. Cortes, C. and V. Vapnik, *Support-vector networks.* Machine learning, 1995. **20**(3): p. 273-297.

26. Osuna, E., R. Freund, and F. Girosit. *Training support vector machines: an application to face detection*. 1997.

27. Joachims, T., C. Nedellec, and C. Rouveirol. *Text categorization with support vector machines: learning with many relevant*. 1998: Springer.

28. John, G.H., R. Kohavi, and K. Pfleger. *Irrelevant features and the subset selection problem*. 1994: In Proceedings of ICML-94, 11th International Conference on Machine Learning (New Brunswick, NJ, 1994), 121-129. .

29. Frohlich, H., O. Chapelle, and B. Scholkopf. *Feature selection for support vector machines by means of genetic algorithms*. 2003: IEEE Computer Society Washington, DC, USA.

30. Hsu, C.W., C.C. Chang, and C.J. Lin, *A practical guide to support vector classification*. 2003, Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 2003. available at http://www.csie.ntu.edu.tw/cjlin/libsvm/.

31. Keerthi, S., *Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms.* IEEE Transactions on neural networks, 2002. **13**(5): p. 1225-1229.

32. Clerc, M. and J. Kennedy, *The particle swarm - explosion, stability, and convergence in a multidimensional complex space.* IEEE transactions on Evolutionary Computation, 2002. **6**(1): p. 58-73.

33. Boser, B.E., I.M. Guyon, and V.N. Vapnik. *A training algorithm for optimal margin classifiers*. 1992: ACM New York, NY, USA.

34. Vapnik, V., *Statistical learning theory.* NY Wiley, 1998.

35. Sch lkopf, B., C.J.C. Burges, and A.J. Smola, *Advances in kernel methods: support vector learning*. 1998: The MIT press.

36. Keerthi, S. and C. Lin, *Asymptotic behaviors of support vector machines with Gaussian kernel.* Neural computation, 2003. **15**(7): p. 1667-1689.

37. Kennedy, J. and R.C. Eberhart. *Particle swarm optimization*. in*: Proceedings of the IEEE International Conference on Neural Network, vol. 4, 1995, pp. 1942–1948.* 1995.

38. Eberhart, R.C. and J. Kennedy. *A new optimizer using particle swarm theory*. in*: Sixth international symposium on micro machine and human science, Nagoya, pp 39–43*. 1995.

39. Shi, Y. and R. Eberhart. *A modified particle swarm optimizer*. in *Proceedings of the IEEE international conference on evolutionary computation, IEEE Press, Piscataway, NJ (1998) p. 69–73.* 1998.

40. Ratnaweera, A., S. Halgamuge, and H. Watson, *Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients.* IEEE transactions on Evolutionary Computation, 2004. **8**(3): p. 240-255.

41. Eberhart, R.C. and Y. Shi. *Particle swarm optimization: developments, applications and resources*. 2001: in: Proceedings of 2001 Congress on evolutionary computation,vol.1 2001,pp.81-86.

42. Shi, Y. and R.C. Eberhart. *Empirical study of particle swarm optimization*. 1999: Congress on evolutionary computation, Washington D.C., USA, pp 1945–1949.

43. Kennedy, J. and R.C. Eberhart. *A discrete binary version of the particle swarm algorithm*. in*:Proceedings of IEEE conference on systems, man and cybernetics, pp 4104–4108*. 1997.

44. Chang, C.C. and C.J. Lin, *LIBSVM: a library for support vector machines*. 2001, Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

45. Salzberg, S.L., *On comparing classifiers: Pitfalls to avoid and a recommended approach.* Data mining and knowledge discovery, 1997. **1**(3): p. 317-328.

46. Statnikov, A., et al., *GEMS: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data.* International Journal of Medical Informatics, 2005. **74**(7-8): p. 491-503.

47. Chen, Y.-W. and C.-J. Lin, *Combining SVMs with Various Feature Selection Strategies*, in *Feature Extraction*. 2006. p. 315-324.