Research Article

A New Hybrid Algorithm for Bankruptcy Prediction Using Switching Particle Swarm Optimization and Support Vector Machines

Yang Lu,[1] Nianyin Zeng,[2] Xiaohui Liu,[3,4] and Shujuan Yi[1]

[1]*College of Information Technology, Heilongjiang Bayi Agricultural University, Daqing 163319, China*
[2]*Department of Mechanical and Electrical Engineering, Xiamen University, Xiamen 361005, China*
[3]*Department of Computer Science, Brunel University, Uxbridge, Middlesex UB8 3PH, UK*
[4]*Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia*

Correspondence should be addressed to Yang Lu; luyanga@sina.com

Received 27 November 2014; Accepted 19 December 2014

Academic Editor: Zidong Wang

Bankruptcy prediction has been extensively investigated by data mining techniques since it is a critical issue in the accounting and finance field. In this paper, a new hybrid algorithm combining switching particle swarm optimization (SPSO) and support vector machine (SVM) is proposed to solve the bankruptcy prediction problem. In particular, a recently developed SPSO algorithm is exploited to search the optimal parameter values of radial basis function (RBF) kernel of the SVM. The new algorithm can largely improve the explanatory power and the stability of the SVM. The proposed algorithm is successfully applied in the bankruptcy prediction problem, where experiment data sets are originally from the UCI Machine Learning Repository. The simulation results show the superiority of proposed algorithm over the traditional SVM-based methods combined with genetic algorithm (GA) or the particle swarm optimization (PSO) algorithm alone.

1. Introduction

In the past few years, bankruptcy prediction has been extensively investigated since it is a critical issue in the accounting and finance field [1, 2]. Bankruptcy prediction is very important for banks, since they may lose billions of dollars each year due to the bankruptcy of companies. Therefore, banks need to make a decision on the amount and the interest rate of a loan which reflect the creditworthiness of the counter party. It is also of great importance to predict the risk for accounting and financial researchers, credit rating agencies, and other organizations when dealing with counter parties on the business transactions. Thus, there has been a growing research interest in applying statistical methods, expert systems, and artificial intelligence technologies in the bankruptcy early warning model so as to obtain higher prediction accuracy. The bankruptcy is a typical binary classification problem since there are only two results of prediction, bankruptcy and nonbankruptcy. Up to now, many researchers have proposed some classical bankruptcy prediction models based on statistical methods such as Beaver's univariate model and Altman's multivariate model (see, e.g., [1]). However, the validity of these traditional statistical methods mainly depends on the subjective judgments of the human financial experts when applied in the selection of some parameters which, in turn, inevitably makes feature selection bias.

With the development of data mining techniques, machine learning methods have been exploited by many researchers for the bankruptcy prediction problem since these methods can provide an unbiased feature selection and decision making mechanism. Let us now provide a brief review of the latest literature on the issue of the bankruptcy prediction problem based on the intelligent methods. In [3, 4], artificial neural networks (ANNs) have been used for the bankruptcy prediction problem. Genetic algorithm (GA) has been applied for the feature selection in [5]. On the other hand, a hybrid of ANNs and GA algorithm has been proposed in [6] for the bankruptcy prediction. Recently, data

mining methods [7] have been widely investigated for the bankruptcy prediction problem. As is well known, ANNs methods suffered from multiple local minima which significantly limits their applications. In the context of bankruptcy prediction, the issue to be investigated is a typical binary classification problem, and there is a great need to seek a more appropriate approach that is capable of handling the pattern classification. In search of such a candidate, SVM appears to be an ideal one because of its advantages such as the fact that the solution to the SVM is global and unique and it has a simple geometric interpretation and gives a sparse solution. SVM models, especially, have already been successfully applied to many financial problems including bankruptcy prediction [8]. However, there is a limitation of the SVM approach which lies in the selection of the parameter values of the kernel function. In order to cope with the limitation, intelligent optimization algorithms, such as GA and the PSO, have been exploited to optimize both the feature subset and the parameters of SVM; see, for example, [9].

Compared with GA, PSO does not have genetic operators such as crossover and mutation, so it is implemented easily with very few parameters to adjust. The PSO algorithm has been successfully applied in a variety of areas because of its advantages such as the effectiveness in performing difficult optimization tasks and the convenience for implementation with fast convergence to a reasonably good solution [10, 11]. PSO, especially, has been applied for optimizing the parameters of kernel functions in the SVM; see, for example, [9] and the references therein. Up to now, several variants of PSO algorithms have been proposed in order to improve the search ability of the PSO algorithm. However, the PSO algorithm has some disadvantages such as easy falling into local minima, slow convergence speed, and poor accuracy. In order to overcome these drawbacks, a SPSO algorithm has been proposed in [12]. The SPSO algorithm has been developed which introduces a mode-dependent velocity updating equation with Markovian switching parameters in order to overcome the contradiction between the local search and global search. The proposed SPSO algorithm can not only avoid the local search stagnating in a local area and wasting more time on an invalid search but also lead the swarm move to a more potential area quickly. Therefore, the SPSO algorithm can greatly improve the ability of SVM. Because of its advantages, the SPSO has been successfully applied in the constrained optimization problem and parameters estimation problem; see, for example, [11, 12]. Unfortunately, up to now, little research has been done on the integration of SPSO and the SVM method, also there has been a great potential application in bankruptcy prediction problem. Inspired by the above discussion, in this paper, we apply the SPSO algorithm to optimize the parameters of the kernel function in the SVM method for the bankruptcy prediction problem.

In this paper, we aim to develop a hybrid SPSO and SVM method for bankruptcy prediction problem. The proposed method integrates the merits of SPSO and the SVM, which can optimize the parameters of kernel functions in SVM and the penalty parameter c simultaneously. The experiment data sets are originally from the UCI Machine Learning Repository donated on February 9, 2014 [13]. The main contribution of this paper is mainly twofold. (1) A new hybrid approach called switching particle swarm optimization-support vector machine (SPSO-SVM) is proposed to the issue of the bankruptcy prediction. Note that the proposed algorithm can correctly and effectively predict the bankruptcy. (2) Experiment results have shown that the proposed algorithm can not only largely improve the explanatory power and the stability of the SVM, but also obtain a higher prediction accuracy than other SVM models that are optimized by (1) standard PSO algorithm; (2) genetic algorithm (GA); or (3) SVM that all the parameters are randomly initialized.

The rest of this paper is organized as follows: The relevant background is introduced in Section 2. In Section 3, the SPSO algorithm for parameters optimization of the SVM model is described. The results of bankruptcy prediction by the proposed method are discussed in Section 4 and the overall performance is also demonstrated. Finally, concluding remarks are given in Section 5.

2. Research Background

2.1. Support Vector Machine. SVM, firstly developed by Vapnik in 1995 [14], is a supervised learning model with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis [15]. It was constructed on the basis of VC-dimensional theory and the structural risk minimization principle of statistical learning theory. The main features of SVM are the use of kernels, the absence of local minima, the sparseness of the solution, and the capacity control obtained by optimizing the margin [16]. Compared with other algorithms, SVM has many unique advantages when applied in solving small sample, nonlinear, and high-dimensional pattern recognition problem. In order to gain the best generalization ability, SVM will find the best compromise between model complexity and the learning ability.

The SVM has two kinds of models when applied for classification, which are the linear SVM and the nonlinear SVM. First of all, let us provide a brief introduction of the linear SVM. Given a data set of n points $D = \{(x_i, y_i) \mid x_i \in R^p, y_i \in \{-1, 1\}\}_{i=1}^n$ where y_i is either 1 or -1, indicating the class to which the point x_i belongs, each x_i is a p dimensional real vector. We define a linear discriminated function, $g(x) = w \cdot x + b$, that was produced by the linear combination of x individual components. The decision rule for the two classes (C_1 and C_2) problem is as follows: If $g(x) >= 0$, then x belongs to C_1. If $g(x) < 0$, then x belongs to C_2. Especially, when the equation $g(x) = 0$, that is called as the hyper plane. This can be rewritten as

$$y_i \left(w \cdot x_i - b \right) \geq 1, \quad \forall 1 \leq i \leq n, \tag{1}$$

where b is offset in linear space.

Remark 1. w would be zero only when all points have the same class. When points are linearly separable, length of w is const margin between classes.

In the case that the sample is nonlinear separable essentially, we need to use the nonlinear SVM which is a hyperplane in the high-dimensional feature space. The classification function is represented by

$$f(x) = \operatorname{sgn}\left\{ \sum_{i=1}^{n} \alpha_i^* y_i k(x_i, x) + b^* \right\}, \tag{2}$$

where sgn stands for signal function and b^* represents optimal hyperplane offset in nonlinear space.

Remark 2. SVM maps the input vectors x into a high-dimensional feature space Z through some nonlinear mapping, chosen a priori. In this space, an optimal separating hyperplane is constructed. To be strict a so-called linear SVM is an optimal hyperplane.

The kernel function K is defined as follows:

$$K(x, y) = \phi(x) \cdot \phi(y) \quad \forall x, y \in X, \tag{3}$$

where $\phi : X \to F$ is mapping from the input space X to the feature space F (Hilbert space). In this paper, Gaussian RBF is selected as the kernel function of SVM due to the fact that only a single parameter needs to be tuned. The RBF can be expressed as follows:

$$k(x, x_i) = \exp\left(-\gamma \| x - x_i \|^2 \right), \tag{4}$$

where exp stands for exponential function and $\gamma = 1/2\sigma^2$. In this situation, the corresponding feature space is a Hilbert space of infinite dimensions.

However, the performance of SVM depends largely on the selection of parameters, so setting appropriate values for the kernel parameters is a key step in the SVM accurate prediction. How to find the best values is difficult and also a hot research interest of the SVM algorithm. In this paper, a novel switching particle swarm optimization algorithm is used for the parameter optimization of the SVM. The position of a particle x_i, especially, denotes an optimized parameter σ, and the fitness of a particle x_i is calculated by using a 10-fold cross validation. Therefore, the data set is divided into 10 groups in this paper. One of the 10 groups is used as the test sample and the other 9 groups are exploited to train the SVM. The fitness function fitness(x) is described as follows:

$$\text{fitness}(x) = \mu \Delta(\text{precision}) + \nu, \tag{5}$$

where μ and ν are weight values that are set as 0.8 and 0.2, respectively.

2.2. Particle Swarm Optimization. PSO is an effective global optimization algorithm developed by Kennedy and Eberhart in 1995 [10]. It is based on the movement of organisms in a bird flock or fish school to optimize a problem by having a population of candidate solutions represented by particles and moving these particles around in the search space according to some kinds of mathematical equation over the particle's position and velocity. Each particle's movement is influenced by its local best known position and also guided toward the best known position in the search space, which are updated as better positions are found by other particles. Therefore, the algorithm is expected to move the swarm toward the best solutions. Compared with other optimization algorithms, the PSO algorithm retains the global search strategy and avoids the complex genetic manipulation and the optimal solution can be found within a few iterations.

Assuming the particle swarm search in an n-dimensional space, the population $Z = \{Z_1, Z_2, \ldots, Z_m\}$ is composed of m particles; the position of each particle in population $Z_i = \{Z_{i1}, Z_{i2}, \ldots, Z_{in}\}$ represents a solution to the problem. Particles constantly adjust their positions Z_i to search for a new solution. Each particle can remember their optimal solution, denoted by P_{id}, and the swarm experienced the best position, that is, the currently optimal solution, denoted by P_{gd}. The velocity of the ith particle is represented by $V_i = \{v_{i1}, v_{i2}, \ldots, v_{in}\}$. Therefore, the velocity and position of the particles at $(t + 1)$th iteration are updated according to the following equations:

$$v_{id}(t+1) = w v_{id}(t) + \eta_1 \text{rand}() \left(p_{id} - z_{id}(t) \right)$$
$$+ \eta_2 \text{rand}() \left(p_{gd} - z_{id}(t) \right) \tag{6}$$
$$z_{id}(t+1) = z_{id}(t) + v_{id}(t+1),$$

where $v_{id}(t+1)$ represents the ith particle velocity on the d-dimension at $t + 1$ iteration, w is the inertia weight, η_1 and η_2 are constant acceleration coefficients, and rand() is a random number between 0 and 1.

The original form of PSO algorithm described above has been successfully applied in a variety of fields because of its advantages such as the effectiveness in performing difficult optimization tasks and the convenience for implementation with fast convergence to a reasonably good solution. However, PSO algorithm suffered from the premature convergence problem without consistently converging to the global optimum. Nevertheless, various methods have been exploited to overcome this limitation; see, for example, [11, 12, 17] and the references therein.

3. Switching Particle Swarm Optimization for the Parameter Optimization of SVM

In this section, a modified version of the traditional PSO algorithm, namely, SPSO algorithm [11, 12], is employed to optimize the parameters of SVM for the bankruptcy prediction problem.

In the SPSO algorithm, a mode-dependent velocity updating equation with Markovian switching parameters is introduced to overcome the contradiction between the local search and global search. Generally speaking, in the early search stage, the particle in the swarm should keep its independence and the swarms diversity, which helps to enlarge the search scope and avoid premature convergence to a local optimal. In the latter stage of the search process, all the swarms may converge to the best particle for getting a more

```
(1)  Parameter ← SwarmParaInitialization()
(2)  While not satisfying the terminal condition do
(3)      for i = 1 to swarm numbers S do
(4)          evaluate Fitness(x_i)
(5)          update Particle Best()
(6)          update Swarm Best()
(7)          if Div(k) < threshold then
(8)              increase the x of Π^(k) = ()
(9)          end if
(10)         if Div(k) < threshold then
(11)             increase the χ of Π^(k) = ( χ  1 − χ )   (χ ∈ [0.4, 0.8])
                                          ( χ  1 − χ )
(12)         end if
(13)         if P_best < threshold then
(14)             Π^(k) = ( ν  1 − ν )   (ν ∈ [0.04, 0.3])
                          ( ν  1 − ν )
(15)         end if
(16)         Calculate new Velocity() of the Particle by (7)
(17)         Calculate new Position() of the Particle by (7)
(18)     end for
(19) end while
```

Algorithm 1: Pseudocode of the SPSO.

accurate solution. The velocity and position of the particle are updated according to the following equations:

$$
\begin{aligned}
v_i(k+1) &= w(\xi(k)) v_i(k) \\
&\quad + c_1(\xi(k)) \gamma_1 (p_1(k) - x_i(k)) \\
&\quad + c_2(\xi(k)) \gamma_2 (p_g(k) - x_i(k)) \\
x_i(k+1) &= x_i(k) + v_i(k+1),
\end{aligned}
\tag{7}
$$

where $w(\xi(k))$, $c_1(\xi(k))$, and $c_2(\xi(k))$ are the inertia weight and acceleration coefficients. All of them are mode-dependent on a Markov chain. Let $\xi(k)$ be a Markov chain taking values in a finite state space $S = \{1, 2, \ldots, N\}$ with probability transition matrix $\Pi^k = (\pi_{ij}^{(k)})_{N \times N}$. $\Pi_{ij}^k \geq 0$ $(i, j \in S)$ is the transition rate from i to j and $\sum_{j=1}^{N} \Pi_{ij}^k = 1$. Π^k can be adjusted by the current search information to balance the global and local search abilities as the following equation:

$$
\text{Div}(k) = \frac{1}{S \cdot |L|} \sum_{i=1}^{S} \sqrt{\sum_{d=1}^{D} (x_{id}(k) - \overline{x}_d(k))^2},
\tag{8}
$$

where S is the swarm size, $|L|$ is the length of the longest diagonal in the search space, D is the dimension of the object problem, x_{id} is the dth value of the ith particle, and $\overline{x}_d$ is the dth value of the average point $\overline{x}$ in the whole swarm that can be computed by

$$
\overline{x}_d = \frac{1}{S} \sum_{i=1}^{S} x_{id}(k).
\tag{9}
$$

Based on the above discussion, the pseudocode of the SPSO algorithm is illustrated in Algorithm 1.

Finally, the flowchart of using the SPSO algorithm to optimize the parameters of SVM is presented in Figure 1.

4. An Illustrative Example

In this section, we apply the hybrid SPSO and SVM algorithm in the bankruptcy prediction problem. The experiment data sets are originally from the UCI Machine Learning Repository donated on February 9, 2014, with URL http://archive.ics.uci.edu/ml/datasets.html. The data sets consist of 143 non-bankruptcy samples and 107 bankruptcy samples; that is, we totally have 250 sample data sets. Each sample data set, especially, includes 6 attributes corresponding to the qualitative parameters in the bankruptcy problem. Each attribute contains three parameters, which are P = Positive, A = Average, and N = Negative, respectively. There are only two prediction results for the bankruptcy problem, which are B = Bankruptcy and NB = Nonbankruptcy.

The SPSO algorithm is implemented in Matlab environment, and the LIBSVM [18] library is exploited for the SVM classifier. Firstly, the parameters "P," "A," and "N" are denoted by 1, 0, and −1, respectively. Secondly, the parameters "NB" and "B" are denoted by 1 and 2, respectively, which represent the prediction results of SVM belonging to the first class and the second class, respectively. At the same time, a uniformly distributed random noise was added to the sample data sets in order to represent the different weights when experts are predicting the bankruptcy. Finally, from 1 to 71 sample data sets of nonbankruptcy and from 144 to 196 sample data sets of bankruptcy are selected as training data sets for the SVM.
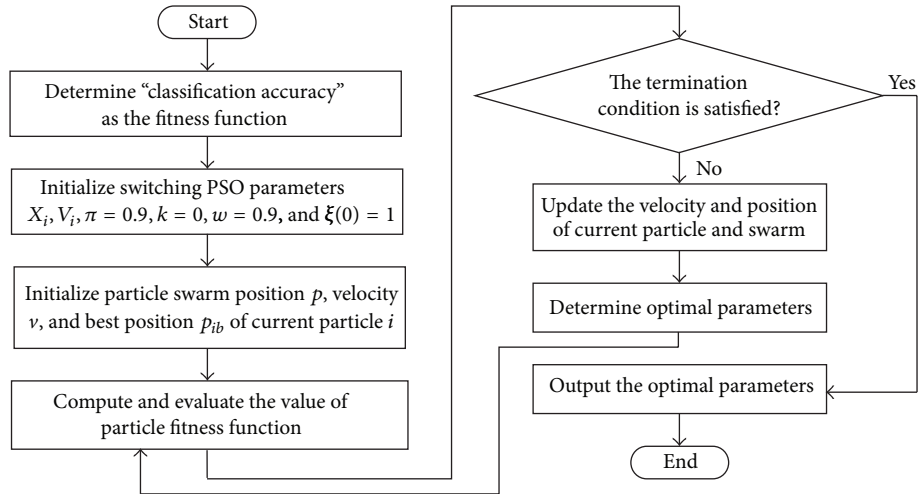
FIGURE 1: The flowchart of optimizing the parameters of SVM by the SPSO algorithm.
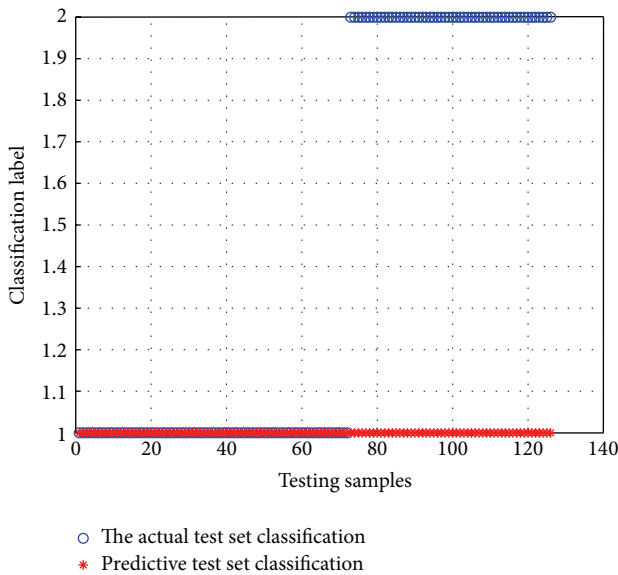

o The actual test set classification
* Predictive test set classification

FIGURE 2: The classification result of the SVM model.

—*— Best fitness
—o— Average fitness

FIGURE 3: The fitness curve of searching for the best parameters by the GA method.

We use the 10-fold cross validation, especially, to find out the optimal parameter values of the kernel function.

The classification results of the SVM model and the proposed hybrid SPSO and SVM model are shown in Figures 2 and 6, respectively. Therefore, for the 126 testing samples, we can see from Figure 2 that the basic SVM model where the parameters are set according to experience can only predict 72 testing samples; the average prediction accuracy is 57.1429%. However, it can be seen from Figure 6 that the optimized SVM model can predict 125 testing samples; the average prediction accuracy reached 99.2063%. On the other hand, the fitness curves of SVM models optimized by GA, PSO, and the SPSO algorithm are shown in Figures 3–5, respectively. It is observed from Figures 3–5 that the GA-SVM algorithm has better prediction accuracy than the PSO-SVM method; however, the GA-SVM algorithm takes more time. In particular, we can also see from these figures that the
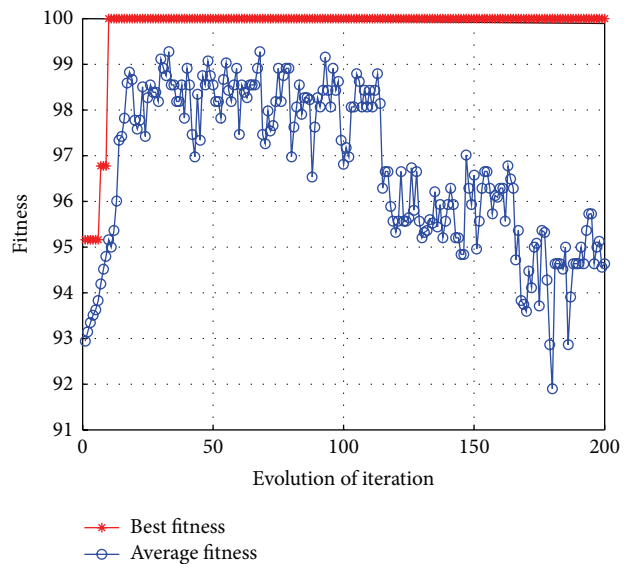
presented hybrid SPSO and SVM method outperformed the other three methods when the time and the prediction accuracy are considered simultaneously.

In order to further evaluate the performance of the proposed method in a quantitative way, let us calculate the prediction accuracy and the time of simulation for the above 4 different SVM models. The results are shown in Table 1. Compared with SVM, GA-SVM, and PSO-SVM methods, the experimental results show the superiority of the proposed method, which can largely improve the explanatory power and the stability of the SVM.

5. Conclusions

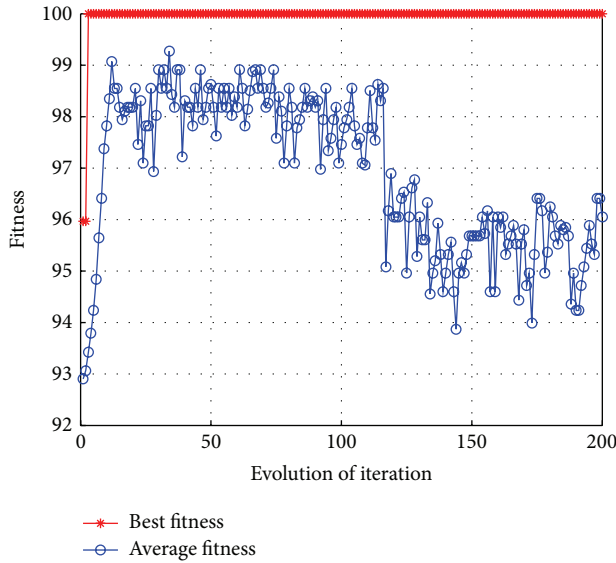In this paper, we have presented a hybrid SPSO and SVM algorithm for the bankruptcy prediction problem. Note that

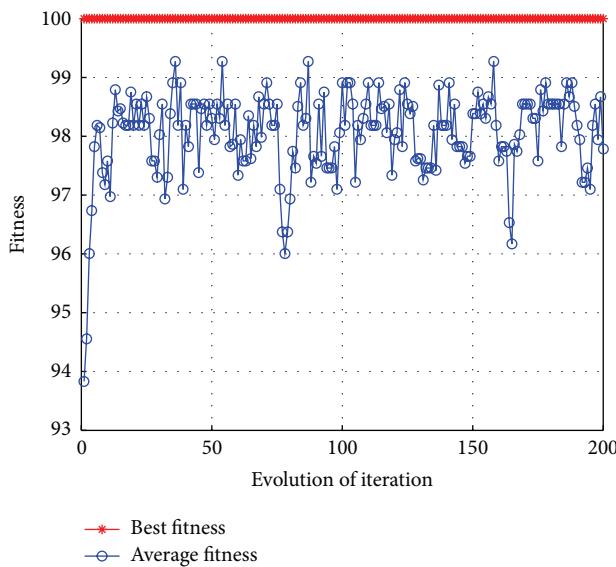FIGURE 4: The fitness curve of searching for the best parameters by the PSO method.
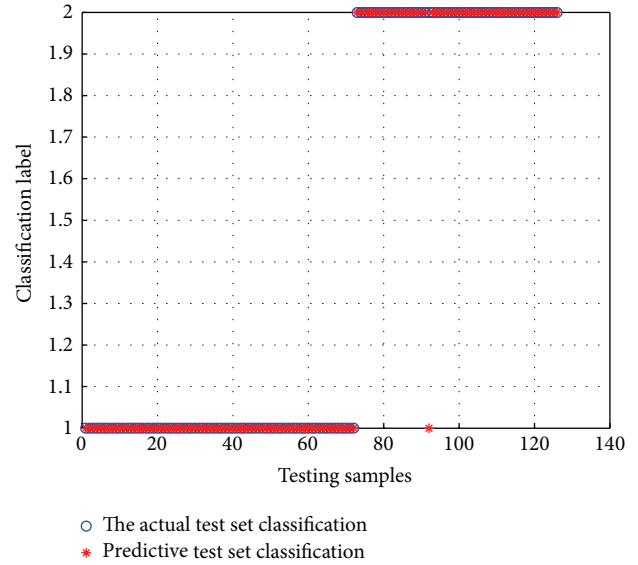

FIGURE 6: The classification result of the hybrid SPSO and SVM model.

parameter values of RBF kernel of the SVM. Compared with SVM, GA-SVM, and PSO-SVM methods, the experimental results show the superiority of the proposed method, which can largely improve the explanatory power and the stability of the SVM. Finally, the proposed algorithm is successfully applied in the bankruptcy prediction model based on the data sets which are originally from the UCI Machine Learning Repository. This can provide a new method for the bankruptcy prediction. Future research directions would include the application of the developed hybrid SPSO and SVM algorithm in the filtering and control problems [19–33].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

FIGURE 5: The fitness curve of searching for the best parameters by the SPSO method.

TABLE 1: Performance comparison.

Model name	c	γ	Prediction accuracy (%)	Time (s)
SVM	2.0	1.0	57.1429	2.144917
GA-SVM	1.62601	0.348282	98.4127	220.315372
PSO-SVM	0.676155	0.430108	94.4127	85.521112
SPSO-SVM	1.85627	0.939559	99.2063	74.847488

the bankruptcy prediction problem is a critical issue in the accounting and finance field. In particular, a recently developed SPSO algorithm is exploited to optimize the optimal

References

[1] E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The Journal of Finance*, vol. 23, no. 4, pp. 589–609, 1968.

[2] M. Zmijewski, "Methodological issues related to the estimation of financial distress prediction models," *Journal of Accounting Research*, vol. 22, pp. 59–82, 1982.

[3] R. L. Wilson and R. Sharda, "Bankruptcy prediction using neural networks," *Decision Support Systems*, vol. 11, no. 5, pp. 545–557, 1994.

[4] C.-F. Tsai and J.-W. Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2639–2649, 2008.

[5] K.-S. Shin and Y.-J. Lee, "A genetic algorithm application in bankruptcy prediction modeling," *Expert Systems with Applications*, vol. 23, no. 3, pp. 321–328, 2002.

[6] B. Back, T. Laitinen, and K. Sere, "Neural networks and genetic algorithms for bankruptcy predictions," *Expert Systems with Applications*, vol. 11, no. 4, pp. 407–413, 1996.

[7] D. L. Olson, D. Delen, and Y. Meng, "Comparative analysis of data mining methods for bankruptcy prediction," *Decision Support Systems*, vol. 52, no. 2, pp. 464–473, 2012.

[8] J. H. Min and Y.-C. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters," *Expert Systems with Applications*, vol. 28, no. 4, pp. 603–614, 2005.

[9] F. Melgani and Y. Bazi, "Classification of electrocardiogram signals with support vector machines and particle swarm optimization," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 5, pp. 667–677, 2008.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Network*, pp. 1942–1948, Perth, Australia, 1995.

[11] N. Zeng, Z. Wang, Y. Li, M. Du, and X. Liu, "A hybrid EKF and switching PSO algorithm for joint state and parameter estimation of lateral flow immunoassay models," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 2, pp. 321–329, 2012.

[12] Y. Tang, Z. Wang, and J.-A. Fang, "Parameters identification of unknown delayed genetic regulatory networks by a switching particle swarm optimization algorithm," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2523–2535, 2011.

[13] A. Asuncion and D. Newman, "UCI machine learning repository [EB/OL]," 2014, http://archive.ics.uci.edu/ml/datasets.html.

[14] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[15] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[16] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge, UK, 2004.

[17] N. Zeng, Y. S. Hung, Y. Li, and M. Du, "A novel switching local evolutionary PSO for quantitative analysis of lateral flow immunoassay," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1708–1715, 2014.

[18] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, 2011.

[19] Z. Wang, H. Dong, B. Shen, and H. Gao, "Finite-horizon H_∞ filtering with missing measurements and quantization effects," *IEEE Transactions on Automatic Control*, vol. 58, no. 7, pp. 1707–1718, 2013.

[20] Z. Wang, D. Ding, H. Dong, and H. Shu, "H_∞ consensus control for multi-agent systems with missing measurements: the finite-horizon case," *Systems & Control Letters*, vol. 62, no. 10, pp. 827–836, 2013.

[21] H. Dong, Z. Wang, S. X. Ding, and H. Gao, "Finite-horizon estimation of randomly occurring faults for a class of nonlinear time-varying systems," *Automatica*, vol. 50, no. 12, pp. 3182–3189, 2014.

[22] X. He, Z. Wang, X. Wang, and D. H. Zhou, "Networked strong tracking filtering with multiple packet dropouts: algorithms and applications," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 3, pp. 1454–1463, 2014.

[23] G. Wei, F. Han, L. Wang, and Y. Song, "Reliable H_∞ filtering for discrete piecewise linear systems with infinite distributed delays," *International Journal of General Systems*, vol. 43, no. 3-4, pp. 346–358, 2014.

[24] Q. Liu, Z. Wang, X. He, and D. H. Zhou, "A survey of event-based strategies on control and estimation," *Systems Science and Control Engineering*, vol. 2, no. 1, pp. 90–97, 2014.

[25] H. Dong, Z. Wang, J. Lam, and H. Gao, "Distributed filtering in sensor networks with randomly occurring saturations and successive packet dropouts," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 12, pp. 1743–1759, 2014.

[26] L. Qin, X. He, and D. Zhou, "A survey of fault diagnosis for swarm systems," *Systems Science & Control Engineering*, vol. 2, no. 1, pp. 13–23, 2014.

[27] H. Dong, S. X. Ding, and W. Ren, "Distributed filtering with randomly occurring uncertainties over sensor networks: the channel fading case," *International Journal of General Systems*, vol. 43, no. 3-4, pp. 254–266, 2014.

[28] L. Wang, G. Wei, and W. Li, "Probability-dependent H_∞ synchronization control for dynamical networks with randomly varying nonlinearities," *Neurocomputing*, vol. 133, pp. 369–376, 2014.

[29] S. Zhang, Z. Wang, D. Ding, and H. Shu, "Fuzzy filtering with randomly occurring parameter uncertainties, interval delays, and channel fadings," *IEEE Transactions on Cybernetics*, vol. 44, no. 3, pp. 406–417, 2014.

[30] S. Zhang, Z. Wang, D. Ding, and H. Shu, "H_∞ fuzzy control with randomly occurring infinite distributed delays and channel fadings," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 1, pp. 189–200, 2014.

[31] G. L. Wei, L. C. Wang, and F. Han, "A gain-scheduled approach to fault-tolerant control for discrete-time stochastic delayed systems with randomly occurring actuator faults," *Systems Science & Control Engineering*, vol. 1, no. 1, pp. 82–90, 2013.

[32] J. Hu, Z. Wang, B. Shen, and H. Gao, "Quantised recursive filtering for a class of nonlinear systems with multiplicative noises and missing measurements," *International Journal of Control*, vol. 86, no. 4, pp. 650–663, 2013.

[33] J. Hu, D. Chen, and J. Du, "State estimation for a class of discrete nonlinear systems with randomly occurring uncertainties and distributed sensor delays," *International Journal of General Systems*, vol. 43, no. 3-4, pp. 387–401, 2014.