Expert Systems with Applications 36 (2009) 11543–11551

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Applying enhanced data mining approaches in predicting bank performance: A case of Taiwanese commercial banks

Shih-Wei Lin [a,*], Yeou-Ren Shiue [b], Shih-Chi Chen [c], Hui-Miao Cheng [d]

[a] Department of Information Management, Chang Gung University, Taiwan
[b] Department of Information Management, Huafan University, Taiwan
[c] Department of Industrial Management, National Taiwan University of Science and Technology, Taiwan
[d] Department of Information Management, China University of Technology, Taiwan

ARTICLE INFO

Keywords:
Bank performance
Data mining
Particle swarm optimization
Parameter optimization
Feature selection

ABSTRACT

The prediction of bank performance is an important issue. The bad performance of banks may first result in bankruptcy, which is expected to influence the economics of the country eventually. Since the early 1970s, many researchers had already made predictions on such issues. However, until recent years, most of them have used traditional statistics to build the prediction model. Because of the vigorous development of data mining techniques, many researchers have begun to apply those techniques to various fields, including performance prediction systems. However, data mining techniques have the problem of parameter settings. Therefore, this study applies particle swarm optimization (PSO) to obtain suitable parameter settings for support vector machine (SVM) and decision tree (DT), and to select a subset of beneficial features, without reducing the classification accuracy rate. In order to evaluate the proposed approaches, dataset collected from Taiwanese commercial banks are used as source data. The experimental results showed that the proposed approaches could obtain a better parameter setting, reduce unnecessary features, and improve the accuracy of classification significantly.

1. Introduction

The structure of Taiwanese financial institutions has significantly been affected by the Southeast Asian Financial Crisis of July, 1997. Not only had the average overdue loan climbed from 3% in 1996 to 8.16% by the end of 2001, but also the average loss on assets before tax had reached 109 billion by 2002 (Condition & Performance of Domestic Banks). Undoubtedly, the statistical results show that if these problems are solved promptly it will become a great threat to the country.

The bank performance prediction model uses scientific and systematic approaches to diagnose the financial operations of institutes. According to a precise and strict evaluation, the model can detect the weakness of institutions in advance and provide early warning signals to related financial governments. The performance prediction for financial firms, especially banks, has been an extensively researched area since the late 1960s (Altman, 1968). Creditors, auditors, stockholders, and senior management are all interested in performance prediction (Wilson & Sharda, 1994). The most precise way of monitoring banks is by on-site examinations made by the Financial Supervisory Institutions. In Taiwan, the primary government agencies responsible for the supervision of financial institutions are the Central Bank of China (Taiwan), the Ministry of Finance, and the Central Deposit Insurance Corporation of Taiwan. These three regulators utilize a six-category rating system to indicate the safety and soundness of institutions. This rating, referred to as the CAMELS rating, evaluates banks according to their basic functional areas including Capital adequacy, Asset quality, Management expertise, Earnings strength, Liquidity, and Sensitivity to market risk. This system is made of various financial ratios, obtained from the periodic reports of the entities under jurisdiction (Kao & Liu, 2004).

Early research on this topic applied statistical methods. Beaver (1966) was the first one who used univariate analysis to build the financial prediction model for banks. Altman (1968) pointed out the drawback of Beaver's model, and further utilized discriminant analysis (DA) to build the model. After several years, Altman, Haldeman, and Narayanan (1977) developed a bankruptcy classification model called Zeta analysis, which incorporated comprehensive inputs. Martin (1977) presented logistic regression to predict the probability of bank failure based on the data obtained from the Federal Reserve System. West (1985) used factor analysis and logistic regression to create composite variables to describe a bank's financial and operating characteristics. Both Cielen, Peeters, and Vanhoof (2004) and Kao and Liu (2004) used data envelopment analysis (DEA) to predict the bankruptcy of banks. A comprehensive survey on statistical methods for the prediction of bank performance can be found in Ravi Kumar and Ravi (2007). Statistical

[*] Corresponding author. Tel.: +886 3 2118800; fax: +886 3 2118020.
E-mail address: swlin@mail.cgu.edu.tw (S.-W. Lin).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.03.029

methods in performance prediction build simple models using a small set of financial variables, and these simple models could classify about two-thirds of a holdout sample. These statistical models were succinct and were easy to explain. However, due to strict assumption, such as linearity and normality, independence among predictable variables limits the application in real world. The major problem in applying these methods to the performance prediction is that the multivariate-normality assumptions for independent variables are frequently violated in financial data sets, which may make these methods theoretically invalid for finite samples (Berrt & Linoff, 1997; Huang, Chen, Hsu, Chen, & Wu, 2004).

In recent years, it has been shown that data mining techniques are effective and efficient compared with the statistical methods in finance fields. Data mining techniques can find out the potential and significant information needed from the enormous data by rapid and deep exploration. For examples, Mochón, Quintana, and Sáez (2007) provided the rationale for using soft computing techniques in finance and presented several applications. Quintana, Saez, and Mochon (2007) proposed the evolutionary nearest neighbor classifier (evolutionary nearest neighbor classifier, ENPC) for early bankruptcy prediction. Oleda and Fernandez (1997) provided several classifiers to predict the case of bankruptcy. Their results pointed out that C4.5 and NN have better performance than DA and Logit. Ahn, Cho, and Kim (2000) proposed a hybrid intelligent system by combining a rough set approach and a neural network (NN) to predict the failure of firms based on the past financial performance data. The result showed that the proposed hybrid models outperform both DA and neural network models. de Andres, Landajo, and Lorca (2005) made a comparative analysis on forecasting business profitability of a Spanish case by various classification techniques. Their results pointed out that the artificial intelligence-based approaches are better than traditional statistical methods, such as LDA and Logit models. Ryu and Yue (2005) introduced a method called isotonic separation to the prediction of firm bankruptcy. They used feature reduction methods to reduce the ratios used in the prediction and then various classification methods, such as DA, decision tree (DT), neural networks (NN), learning vector quantization, rough sets, and isotonic separation, were used with reduced ratios. Their experimental results showed that the isotonic separation method is a promising technique, performing better than other methods for bankruptcy prediction. Rastogi and Shim (2000) developed an approach, called the PUBLIC, an improved decision tree classifier that integrated the second "pruning" phase with the initial "building" phase. Experimental results demonstrate the effectiveness of PUBLIC. Although the PUBLIC approach can reduce number of nodes and execution time, it is not designed for achieving the best classification accuracy.

To predict the performance of banks, feature data is required. Selecting the right set of features for classification is a difficult problem when designing a good classifier. Typically, one does not know a priori in which features are relevant for a particular classification task. One common approach is to collect as many features as possible prior to the learning and data-modeling phase. However, in most classification problems, given a large set of potential features, identifying a small subset to classify data object is generally necessary. Data without feature selection might be redundant or noisy, and decrease the classification efficiency.

Lee, Han, and Kwon (1996) used Korean bankruptcy data from 1979–1992 to build hybrid neural network models for bankruptcy prediction. In order to enhance the performance, they used multiple discriminant analysis (MDA) and DT methods, respectively, to reduce the number of input variables. Alam, Booth, Lee, and Thordarson (2000) adapted fuzzy clustering and self-organization neural networks for identifying potentially failing banks. The results showed that both the fuzzy clustering and self-organizing neural networks are promising classification tools for identifying potentially failing banks. Lin and McClean (2001) used four classifiers – DA, logistic regression, neural networks and DT in bankruptcy prediction. Each used two feature selection methods for predicting corporate failure: human judgment, based on financial theory, and the ANOVA statistical method. Park and Han (2002) applied the case-based reasoning (CBR) and feature weights to build the bankruptcy model. Huang et al. (2004) utilized support vector machine (SVM) and back propagation neural network (BPN) to build a model in a related research – credit analysis. Since there are many financial variables, they ran the ANOVA to test whether the differences are significant or not.

Tung, Quek, and Cheng (2004) proposed the use of a new neural fuzzy system to predict banking failure. Their experimental results reveal that hybrid approach has better performance to predict banking failure. Becerra, Galvao, and Abou-Seads (2005) proposed neural and wavelet network models for financial distress. The result showed that their approach is a valid alternative to the classical DA models. Moreover, wavelet networks may have advantages over the conventional multi-layer perceptron structures employed in neural network frameworks. However, feature selection problem was not investigated in their study. Ravi Kumar and Ravi (2006) proposed an ensemble classifier using simple majority voting scheme to bankruptcy prediction in Banks. The experimental results showed that ensemble classifier could perform better than stand-alone classifier. Shin, Lee, and Kim (2005) compared the predictive accuracy of SVM with that of artificial neural network (ANN), MDA and learning vector quantization (LVQ). The performance of SVM was found to be the highest. They applied two stages of input variable selection process. At the first stage, they selected 52 variables among more than 250 financial ratios by an independent-sample *t*-test. In the second stage, they selected 10 variables using an MDA stepwise method to reduce dimensionality. Min and Lee (2005) utilized a grid search method (Hsu, Chang, & Lin, 2003) to find the appropriate parameter setting for SVM and used principal component analysis (PCA) to reduce the number of multi-dimensional financial ratios to two factors. They compared the SVM model with MDA, logistic regression and BPN, and showed that the SVM model has the best accuracy rate. Nevertheless, these approaches are usually applied either to the statistical feature selection or to parameters pretest, or applied them in sequentially. To the best of our knowledge, few studies have simultaneously considered the feature selection and the optimal parameter setting in predicting bank performance.

The purpose of this study is to apply particle swarm optimization (PSO) for performing a feature selection and a parameter determination for two well-known data mining techniques: SVM and DT, and they are called PSO–SVM and PSO–DT, respectively. In order to verify the performance of the proposed approaches, the dataset collected from the Taiwanese commercial banks is used to predict the bank performance.

The remainder of this paper is organized as follows: Section 2 provides an overview of SVM, DT, feature selection and particle swarm optimization (PSO). Section 3 then introduces how the proposed PSO-based approaches are used to perform feature selection and parameter determination for SVM and DT. Experimental results are compared with some of the existing approaches in Section 4. Conclusions are finally drawn in Section 5, along with recommendations for future research.

2. Research background

2.1. Support vector machine

SVM can be briefly described as follows (Burgers, 1998; Cristianini & Shawe-Taylor, 2000). Let $(x_1,y_1\ \ ),\ldots,(x_m,y_m) \in X \times \{\pm 1\}$.

Here, X is some nonempty set form which the patterns x_i are taken, usually referred to as the domain; the y_i are called labels, targets or outputs.

It attempts to identify a hyper-plane in a multi-dimensional space. This hyper-plane functions as a separating plane for data classification. Consider the class of hyper-planes in some dot product space H. The parameters w and b are described as:

$$(\langle w \cdot x \rangle + b) = 0, \quad \text{where } w \in H,\ b \in R. \tag{1}$$

If a hyper-plane satisfying Eq. (1) exists, then a linear separation is obtained. In this case, w and b can be rewritten as

$$\max_{w \in H, b \in R} \min_{1 \leqslant i \leqslant N} \{\|x - x_i\| \mid x \in H, \langle w \cdot x \rangle + b = 0,\ i = 1, \ldots, m\}. \tag{2}$$

Let the distance from the data point to the hyper-plane be $1/\|w\|$. The separating hyper-planes include one optimal separating hyper-plane, OSH, which has the largest distance between the two support vector points on its two sides. Since the distance between the two support vector points is $2/\|w\|$, the minimal distance to OSH, can be derived from Eq. (2).

The margin of a separating hyper-plane, calculated as $2/\|w\|$, determines the hyper-plane's generalization ability. $2/\|w\|$ can be minimized with Eq. (2) and Lagrange's polynomial. Let α denote $(\alpha_1, \ldots, \alpha_m)$. Combining Lagrange's polynomial (in the order of m) with Eq. (2) produces the following maximization equations:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i x_j \rangle \tag{3}$$

$$\text{subject to} \quad \alpha_i \geqslant 0 \quad \text{for all } i = 1, \ldots, m \quad \text{and} \quad \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{4}$$

Quadratic programming can be applied to solve this maximization problem.

Given a vector which satisfies Eq. (3) in maximization, the OSH, expressed in terms of (w,b), can be written as

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i, \tag{5}$$

where the support vector points must satisfy $\alpha_i \geqslant 0$ and Eq. (2). When considering expanding the constraints of Eq. (4), the determinant function of the hyper-plane can be expressed as

$$f(x) = sgn\left(\sum_{i=1}^{m} y_i \alpha_i \langle x, x_i \rangle + b\right) = 0. \tag{6}$$

In most cases, the data is not linearly separable, and is thus mapped to a higher dimensional feature space. This means that if the data cannot be classified explicitly in the current dimensional space, then the SVM will map the data into a higher dimensional space for classification.

The input data are mapped to a higher dimensional feature space by plotting a nonlinear curve. The OSH is built into the feature space. The feature space vectors x, x' are constructed in terms of the kernel k, evaluated on input patterns x, x', $k(x,x') = \langle x,x' \rangle$.

The kernel function can be applied because all feature vectors occur only in dot products. The weight vector then becomes an expansion in the feature space, and therefore typically no longer corresponds to the Φ-image of a single input space vector.

The decision function is expressed as:

$$\begin{aligned} f(x) &= sgn\left(\sum_{i=1}^{m} a_i y_i \langle \Phi(x), \Phi(x_i) + b \rangle\right) \\ &= sgn\left(\sum_{i=1}^{m} a_i y_i k(x_i, x_j) + b\right). \end{aligned} \tag{7}$$

The quadratic program is given by

$$\max_{\alpha \in R^m} W(\alpha) = \sum_{i=1}^{m} a_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{8}$$

$$\text{subject to} \quad \alpha_i \geqslant 0 \quad \text{for all } i = 1, \ldots, m \quad \text{and} \quad \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{9}$$

Several kernel functions help the SVM in finding the optimal solution. The most frequently used functions are the polynomial kernel, sigmoid kernel and radial basis kernel function (RBF) (Lin & Lin, 2003; Müller, Mike, Rätsch, Tsuda, & Scholkopf, 2001; SchÖlkopf & Smola, 2002). The RBF is used most often in general cases, since it can classify multi-dimensional data, unlike a linear kernel function. Furthermore, the RBF has fewer parameters than a polynomial kernel. Overall, the performance of an RBF and other kernel functions are not significantly different. Consequently, the RBF is an effective choice for kernel function.

The two major parameters of RBF, applied in SVM, C and Υ, have to be set appropriately. The parameter C is the cost of penalty and its value influences the classification outcome. If C is too large, then the classification accuracy rate is very high in the training stage, but very low in the testing stage. If C is too small, then the classification accuracy rate is unsatisfactory, making the model useless. The parameter Υ has a much stronger impact than parameter C on classification outcomes, because its value influences the partitioning outcome in the feature space. An excessive value for parameter Υ leads to over-fitting, while a disproportionately small value results in under-fitting (López, Torres, & Batista, 2006). Grid search (Martin, 1977) is the most common method for determining the appropriate values for C and Υ. The values for parameters C and Υ leading to the highest classification accuracy rate in this interval can be found by setting appropriate values for the upper and lower bounds (the search interval) and the jumping interval in the search. Nevertheless, this approach is a local search method, and vulnerable to local optima. Additionally, setting the search interval is a problem. Too large a search interval wastes computational resources, while too small a search interval might render an unsatisfactory outcome.

2.2. Decision trees

Decision trees form a part of "machine learning", an important area of artificial intelligence (Gehrke, Ganti, Ramakrishnan, & Loh, 1999; Quinlan, 1987a). The method approximates the numerical or symbolical-valued target function that is robust to noisy data and capable of learning disjunctive expressions. Owing to the inherent characteristics of DTs, a decision tree employs a top-down divide-and-conquer strategy that recursively partitions a data set into smaller subdivisions. These procedures form the basis of a set of tests defined at each branch in the tree. The tree-like structure is composed of a root node (formed from all of the data), a set of internal nodes (splits), and a set of terminal nodes (leaves).

There are two major phases of DT induction process: the growth phase and the pruning phase (Kim & Koehler, 1995). In the growth phase, for a set of samples in a partition S, a test attribute X is selected for further partitioning the set into $S_1, S_2, \ldots, S_L$ and these are added to the decision tree as the children of node for S. Also the node for S is labeled with test X, and partitions $S_1, S_2, \ldots, S_L$ are then recursively partitioned.

The Interactive Dichotomimizer 3 (ID3) algorithms (Quinlan, 1986, 1987b) and their successor C4.5 (Quinlan, 1993), are the primary foci of research in the field of DT learning. In the tree-creation phase, the central choice in the ID3 algorithm is selection, which features/attributes testing at each node in the most useful way for classifying examples. C4.5 uses an information entropy evaluation function as the selection criteria. The entropy evaluation function is calculated in the following way.

Step 1: Calculate Info(S) to identify the class in the training set S.

$$Info(S) = -\sum_{i=1}^{k}\{[freq(C_i,S)/|S|]\log_2[freq(C_i,S)/|S|]\}, \tag{10}$$

where $|S|$ is the number of cases in the training set. C_i is a class, $i = 1,2,\ldots,k$. k is the number of classes and $freq(C_i,S)$ is the number of cases included in C_i.

Step 2: Calculate the expected information value, $Info_x(S)$ for feature X to the partition S.

$$Info_x(S) = -\sum_{i=1}^{L}[(|S_i|/|S|)Info(S_i)], \tag{11}$$

where L is the number of outputs for feature X, S_i is a subset of S corresponding to the ith output and $|S|$ is the number of cases of the subset S_i.

Step 3: Calculate the information gained after partitioning according to feature X.

$$Gain(X) = Info(S) - Info_x(S). \tag{12}$$

Step 4: Calculate the partition information value $SplitInfo(X)$ acquired for S partitioned into L subsets.

$$SplitInfo(X) = -\sum_{i=1}^{L}\left[\frac{|S_i|}{|S|}\log_2\frac{|S_i|}{|S|}\right]. \tag{13}$$

Step 5: Calculate the gain ratio of $Gain(X)$ over $SplitInfo(X)$

$$GainRatio(X) = Gain(X)/SplitInfo(X). \tag{14}$$

The $GainRatio(X)$ compensates the weak point of $Gain(X)$, which represents the quantity of information provided by X in the training set. Therefore, the feature with the highest $GainRatio(X)$ is taken as the root of the decision tree.

The pruning phase aims to generalize DT that was generated in the growth phase, by generating a sub-tree that avoids over-fitting the training data. The actions of the pruning phase are often referred to as post-pruning, in contrast to the pre-pruning that occurs during the growth phase. The pruning confidence level (confidence level, *CF*) is used to estimate the upper limit of the probability that an error occurs over the population at a leaf. This is done by using the *CF* as a confidence limit for a binomial distribution. The higher the *CF*, the more likely the current error rate is accepted and no pruning will be done (after all, the decision tree decided this was a good split with the given data). A lower *CF* means more errors than occurred in the train data will be predicted and hence there is more chance for pruning because we overestimate the error rate at the leaf (Osei-Bryson, 2007).

Unfortunately, the parameter value for minimum cases *M* in the growing phase and parameter value for pruning *CF* in phasing phase are varied with different problems. The determination of two parameter values becomes an optimization problem (Quinlan, 1993).

Different DT algorithms can be compared and evaluated in different aspects: accuracy, speed, size, robustness and scalability (Han & Kamber, 2006). As surveyed by Demšar (2006), the accuracy criterion is most used to evaluate the decision tree performance.

2.3. Feature selection

Datasets with unimportant, noisy or highly correlated features will significantly decrease the classification accuracy rate. By removing these features, the efficiency and classification accuracy rate can be obtained. Approaches for feature selections can be categorized into two models, namely a filter model and a wrapper model (Liu & Motoda, 1998). Statistical techniques, such as principal component analysis, factor analysis, independent component analysis, discriminant analysis and F-score can be adopted in filter-based feature selection approaches to investigate other indirect performance measures, most of which are based on distance and information. Even though the filter model is fast, the resulting feature subset may not be optimal (Liu & Motoda, 1998).

The wrapper model (Kohavi & John, 1997) applies the classifier accuracy rate as the performance measure. Some researchers have concluded that if the purpose of the model is to minimize the classifier error rate, and the measurement cost for all the features is equal, then the classifier's predictive accuracy is the most important factor. In other words, the classifier should be constructed to achieve the highest classification accuracy. The features adopted by the classifier are then chosen as the optimal features. In the wrapper model, the meta-heuristic approaches are commonly employed to help in looking for the best feature subset. Although meta-heuristic approaches are slow, they obtain the (near) best feature subset (Liu & Motoda, 1998).

2.4. Particle swarm optimization

Particle swarm optimization (PSO) (Kennedy & Eberhart, 1995) is an emerging population-based meta-heuristic that simulates social behavior such as birds flocking to a promising position to achieve precise objectives in a multi-dimensional space. Like evolutionary algorithms, PSO performs searches using a population (called swarm) of individuals (called particles) that are updated from iteration to iteration. To discover the optimal solution, each particle changes its searching direction according to two factors, its own best previous experience and the best experience of all other members. Shi and Eberhart (1998) called the former the cognition part, and the latter the social part.

Each particle represents a candidate position (i.e. solution). A particle is considered as a point in a D-dimensional space, and its status is characterized according to its position x_{id} and velocity v_{id} (Coello, Pulido, & Lechuga, 2004; Noorul Haq, Karthikeyan, Sivakumar, & Saravanan, 2006). The D-dimensional position for the particle i at iteration t can be represented as $x_i^t = \{x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t\}$. Likewise, the velocity (i.e. distance change), which is also a D-dimensional vector, for particle i at iteration t can be described as $v_i^t = \{v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t\}$.

Let $p_i^t = \{p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t\}$ represent the best solution that particle i has obtained until iteration t, and $p_g^t = \{p_{g1}^t, p_{g2}^t, \ldots p_{gD}^t\}$ denote the best solution obtained from p_i^t in the population at iteration t. To search for an optimal solution, each particle changes its velocity according to the cognition and social parts as

$$V_{id}^t = V_{id}^{t-1} + c_1 r_1 (P_{id}^t - x_{id}^t) + c_2 r_2 (P_{gd}^t - x_{id}^t), \quad d = 1,2,\ldots,D \tag{15}$$

where c_1 indicates the cognitive learning factor, c_2 indicates the social learning factor, and r_1 and r_2 are the random numbers uniformly distributed in $U(0,1)$. Each particle then moves to a new potential solution based on the following equation:

$$X_{id}^{t+1} = X_{id}^t + V_{id}^t, \quad d = 1,2,\ldots,D. \tag{16}$$

The basic process of the PSO algorithm is given by

Step 1: (Initialization) Randomly generate initial particles.
Step 2: (Fitness) Measure the fitness of each particle in the population.
Step 3: (Update) Compute the velocity of each particle with Eq. (15).
Step 4: (Construction) For each particle, move to the next position according to Eq. (16).
Step 5: (Termination) Stop the algorithm if the termination criterion is satisfied; return to Step 2 otherwise.

3. The proposed PSO-based approaches

This study adopted a PSO-based approach for parameter determination and feature selection in the SVM and DT, termed as PSO–SVM and PSO–DT. For the SVM without feature selection, two decision variables, designated as C and Υ, are necessary. Meanwhile, for the SVM with feature selection, if n features are required to determine the features to be selected, then $2 + n$ decision variables must be established. The value of n variables ranges between 0 and 1. If the value of a variable is less than or equal to 0.5, then its corresponding feature is not chosen. Conversely, if the value of a variable is greater than 0.5, then its corresponding feature is chosen. Fig. 1 depicts the solution representation of PSO–SVM.

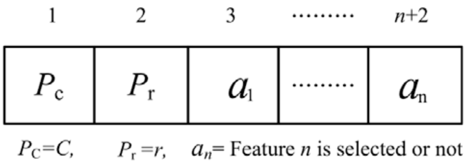

Fig. 1. Solution representation of PSO–SVM.

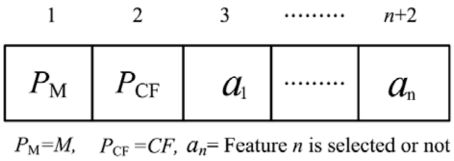

Fig. 2. Solution representation of PSO–DT.

For the DT without feature selection, two decision variables, designated M and CF, are necessary. For the DT with the feature selection, if n features are required to determine the features to be selected, then $2 + n$ decision variables must be established. Similar to the PSO–SVM, whether a feature is chosen or not depends on the value of its corresponding variable. The solution representation of PSO–DT is depicted in Fig. 2.

The PSO–SVM and PSO–DT approaches for bank performance prediction is constructed through the following main steps as shown in Fig. 3. These are explained in detail as in the following.

(1) Input data: The collected data is first input to the system, and then processed by other steps.
(2) Scaling: Scaling was applied to prevent feature values in greater numeric ranges from dominating those in smaller numeric ranges, and to prevent numerical difficulties in the calculation. Experimental results obtained in this study demonstrate that scaling the feature value improves the classification accuracy of SVM. In general, the range of each feature value can be linearly scaled to the range of [−1,+1]. Due to the fact that the scaling in DT is not necessary, the scaling is not applied in DT.
(3) Feature subset selection and parameter values determination: Each particle represents a solution, which denotes the selected subset of features and parameter values. The selected features, parameter values, and training dataset are used for building both SVM and DT classifier models.
(4) Fitness calculation: After the classification model is constructed, the testing dataset is used to calculate the fitness value. The higher the classification accuracy rate it has, the higher the fitness value of particle. If the particle's fitness is better than its best previous experience, the best previous

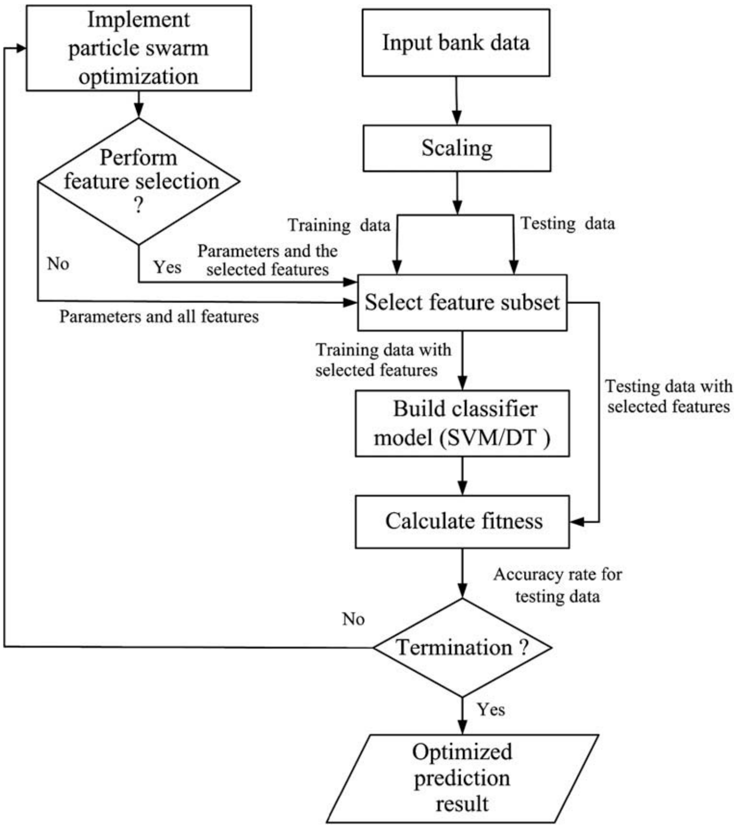
Implement particle swarm optimization
Perform feature selection ?
No
Yes
Parameters and the selected features
Parameters and all features
Input bank data
Scaling
Training data
Testing data
Select feature subset
Training data with selected features
Testing data with selected features
Build classifier model (SVM/DT)
Calculate fitness
Accuracy rate for testing data
No
Termination ?
Yes
Optimized prediction result

Fig. 3. Architecture of the proposed bank performance prediction system.

experience of the particle is updated accordingly. Furthermore, if the particle's fitness is better than the global best fitness, the global best fitness is also updated.

(5) Termination criteria: If the termination criteria are satisfied, then the process ends; otherwise, the next iteration occurs. The termination criterion used in this study is the maximum number of iterations.

(6) PSO process: In this step, the system generates other solutions for particles.

4. Experimental results

The platform adopted to develop the proposed PSO–SVM and PSO–DT approaches uses a PC with the following features: Intel Pentium IV 3.0 GHz CPU, 512 MB RAM, a Windows XP operating system and the Visual C++ 6.0 development environment.

In order to verify the proposed approaches in bank performance prediction, two datasets are used. The first dataset is collected from the Central Bank of China (Taiwan). Data from 44 banks were collected from 2000 to 2004, and had 220 records. Since there are fewer banks in the report based on Taiwanese ratings, the second dataset, from 34 banks, were collected from 2000 to 2005, which had 204 records. Both of the two datasets included 31 input variables and 1 output variable. Among those input variables, there were 28 continuous variables and 3 nominal variables. Table 1 shows the features/variables used in datasets.

Two evaluation criteria of bank performance were adopted for the output variable. One of the evaluation criteria is categorized by principal component analysis (PCA) scoring ("good" or "bad") which is used for the first dataset. Another is referenced by the report of Taiwanese ratings ("well", "average" and "risky") (Taiwan ratings) and is used for the second dataset.

Throughout the initial experiment, the parameter values used in the proposed PSO–SVM and PSO–DT approaches were set as follows. The cognition learning factor c_1 and the social learning factor c_2 for PSO–SVM were set to 2 and 1, respectively. Meanwhile, the c_1 and c_2 values for PSO–DT were set to 1.5 and 2, respectively. The number of particles and the maximum number of iterations were set to be 10 and 1000, respectively, for both the datasets. The searching range of SVM parameter C was between 0.01 and 35,000, while the searching range of SVM parameter Υ was between 0.0001 and 32 (Lin & Lin, 2003). Meanwhile, the searching range of the DT parameter M was between 2 and 10, and the searching range of DT parameter CF was between 0.01 and 0.31.

The k-fold approach (Salzberg, 1997) is used to evaluate the classification accuracy rate. This study set k as 5; that is, the data was divided into five portions. The data from each portion were formed based on the ratio of each category (classification) in the original data. Four portions of data were retrieved as training data and left one portion for testing each time. Since each portion was used as testing data once, five accuracy rates can be obtained. The final accuracy rate was the average of the five accuracy rates.

The results obtained by the proposed PSO–SVM and PSO–DT approaches are shown in Tables 2–4. In order to verify the proposed PSO–SVM approach, the result obtained by grid search is also shown in Tables 2 and 3.

As shown in Table 2 without any feature selection for the proposed PSO–SVM and grid-search approaches, the average classification accuracy rates of the PSO–SVM are 90.909% and 83.793% for dataset 1 and dataset 2, while the average classification accuracy rates of the grid search are 84.54% and 76.37%, respectively. It is evident that the PSO–SVM is better than the grid search. Furthermore, with feature selection for both the PSO–SVM and grid search, even though the numbers of selected features for two approaches are not the same, and there is no domination relationship

Table 1
The description and definition of features used in this study.

Description	Definition
Regulatory Capital to Risk-Weighted Assets	Regulatory capital/Risk-weighted assets
Tier 1 Capital to Risk-Weighted Assets	Tier 1 capital/Risk-weighted assets
Debt-Equity Ratio	Debt/Net Worth
Net Worth to Total Assets Ratio	Net Worth/Total Assets
Non-performing Loan (NPL) Ratio	Non Performing Loan/Loan and Discount
Allowance for Doubtful Debts than Lenders	Allowance for Bad Debt/Loans and Discount
Allowance for Doubtful Debts of Overdue Loans Ratio	Allowance for Bad Debt/Non Performing Loan
Operating Expense to Assets Ratio	Operating Expenses/Average Assets
Operating Expense to Revenue Ratio	Operating Expenses/Operating Sales
Administrative Expense to Sales Ratio	Administrative Expense/Operating Sales
Non-interest Revenue to Sales Ratio	(Operating Sales - Interest Income) /Operating Sales
Assets Turnover Ratio	Operating Sales/Average Assets
Returns on Assets	Pre-Tax Income/Average Assets
Return on Equity	Pre-Tax Income/Average Equity
Pre-Tax Income to Sales Ratio	Pre-Tax Income/Operating Sales
Contribution of Employee	Pre-Tax Income/Number of Employees
Operating Income to Sales Ratio	Operating Income/Operating Sales
Interest-Rate-Sensitive Liabilities than Assets	Interest-Sensitive Asset/Interest-Sensitive Liability
Primary Reserve Ratio	(Cash + Deposits in Other Banks)/Deposits
Liquid Reserve Ratio	Current Assets/Debt
Deposit and Loan Ratio	Loan/Deposit
Time Deposit to Deposit Ratio	Time Deposit/Deposit
Asset Growth Rate	(Current Assets – Prior Assets)/Prior Assets
Revenue Growth Rate	(Current Sales – Prior Sales)/Prior Sales
Earnings Growth Rate	(Current Earnings – Prior Earnings)/Prior Earnings
Deposit Growth Rate	(Current Deposits – Prior Deposits)/Prior Deposits
Lending Growth Rate	(Current Loan – Prior Loan)/Prior Loan
Investment Growth Rate	(Current Investment – Prior Investment)/Prior Investment
Guarantee Growth Rate	(Current Guarantee – Prior Guarantee)/Prior Guarantee
Banks Attribution	1: The New Banks; 2: Business Bank; 3: The old Private Banks; 4: Provinces Banks; 5: State-Owned Banks
Attending to Financial Holding Corporation?	0:No; 1: Yes
Lenders Share	1: The Lowest Lenders Share; 44: The Highest Lenders Share
Scale	*Ln* (Average Assets)

between the two approaches, the average classification accuracy rates of the PSO–SVM are better than those of the grid-search with feature selection by the F-score (Chen & Lin, 2006). The average classification accuracy rates of the PSO–SVM are 98.636% and 93.122% for two datasets, while those of the grid search are 85.795% and 84.187%, respectively, as shown in Table 3. It can be found in Tables 2 and 3 that the average classification accuracy rates of the PSO–SVM and grid search are increased when the feature selection is used.

As shown in Table 4 for the PSO–DT without feature selection, the average classification accuracy rates are 87.727% and 76.427% for dataset 1 and dataset 2, respectively. Meanwhile, the average classification accuracy rates are increased to 89.545% and 81.854% by PSO–DT with feature selection. Thus, with feature selection, the PSO–DT approach yields a higher classification accuracy rate compared with those without feature selection. It seems that by removing those noisy and highly correlated features is greatly beneficial in this study. The results obtained are also compared with those of C4.5 using default parameter values ($M = 2$ and $CF = 0.25$). The accuracy rates obtained by the proposed PSO–DT approach with/without feature selection are better than those of C4.5 using default parameter values, meanwhile the number of rules obtained are smaller than those of C4.5 using default parameter values. It further demonstrates that the PSO-based approach can be used to improve the performance of original C4.5 algorithm.

Even though the PSO–SVM has higher classification accuracy than the PSO–DT, the PSO–SVM cannot provide the rules that the PSO–DT can provide. The number of rules provided by PSO–DT is

Table 2
Experiment results of the proposed PSO–SVM and grid-search without feature selection.

Dataset	Fold	PSO–SVM			Grid search		
		C	*γ*	Accuracy (%)	*C*	*γ*	Accuracy (%)
1	1	28712.4528	0.1003	93.182	32768.0000	0.0001	81.818
	2	44.9900	0.1409	90.909	32768.0000	0.0078	93.181
	3	2240.2050	0.0020	97.727	8192.0000	0.0078	90.904
	4	44.9900	0.0020	84.091	32768.0000	0.0020	75.000
	5	7469.0438	0.0106	88.636	20488.0000	0.0078	81.818
	Avg.	–	–	90.909	–	–	84.544
2	1	44.9900	0.0578	90.244	32.0000	0.1250	87.804
	2	17505.0000	0.5005	85.366	2048.0000	0.0004	78.048
	3	44.9900	0.1030	82.927	2048.0000	0.0078	73.170
	4	44.9900	0.0020	82.927	32.0000	0.1250	82.822
	5	1519.5752	0.3156	77.500	8.0000	0.1250	60.000
	Avg.	–	–	83.793	–	–	76.369

Table 3
Experiment results of the proposed PSO–SVM and grid-search with feature selection.

Dataset	Fold	PSO–SVM with feature selection				Grid search with feature selection by F-score			
		C	*γ*	Accuracy (%)	No. features	C	*γ*	Accuracy (%)	No. features
1	1	16126.519	0.002	100.000	19	256.000	0.008	84.659	31
	2	22776.123	0.550	100.000	18	256.000	0.002	85.227	31
	3	8745.315	0.011	100.000	16	16.000	2.000	81.818	7
	4	4592.790	0.158	95.455	21	256.000	0.008	88.636	31
	5	33200.980	0.002	97.727	20	16.000	0.500	88.636	15
	Avg.	–	–	98.636	18.8	–	–	85.795	23.0
2	1	6524.454	0.880	97.561	17	16.000	0.500	81.595	15
	2	44.990	0.823	92.683	17	16.000	0.500	84.662	15
	3	6993.366	0.543	92.683	13	4.000	2.000	85.276	7
	4	662.929	0.985	92.683	18	4.000	0.500	82.208	15
	5	37178.199	0.589	90.000	17	4.000	2.000	87.195	7
	Avg.	–	–	93.122	16.4	–	–	84.187	11.8

Table 4
Experiment results of the proposed PSO–DT with/without feature selection.

Dataset	PSO–DT with feature selection						PSO–DT without feature selection				C4.5 (using default parameter values)			
	Fold	*M*	*CF*	Accuracy (%)	No. features	No. rules	*M*	*CF*	Accuracy (%)	No. rules	*M*	*CF*	Accuracy (%)	No. rules
1	1	2	0.048	93.182	23	11	11	0.062	90.909	11	2	0.25	75.000	13
	2	3	0.115	90.909	17	4	6	0.205	93.182	7	2	0.25	81.800	6
	3	3	0.094	93.182	19	5	6	0.256	88.636	6	2	0.25	86.400	7
	4	5	0.290	81.818	11	4	4	0.021	79.545	6	2	0.25	75.000	9
	5	10	0.216	88.636	13	6	8	0.199	86.364	5	2	0.25	71.500	10
	Avg.	–	–	89.545	16.6	6.0	–	–	87.727	7.0	–	–	77.940	9
2	1	6	0.110	90.244	18	11	9	0.214	87.805	7	2	0.25	78.000	18
	2	4	0.285	82.927	23	14	9	0.221	80.488	7	2	0.25	68.300	17
	3	12	0.266	73.171	14	8	11	0.011	73.171	7	2	0.25	71.700	18
	4	4	0.158	82.927	18	7	3	0.018	73.171	11	2	0.25	72.500	18
	5	6	0.020	80.000	17	10	9	0.040	67.500	7	2	0.25	63.400	15
	Avg.	–	–	81.854	18	10.0	–	–	76.427	7.8	–	–	70.780	17.2

Table 5
Paired *t*-test on classification accuracy rate.

	PSO–SVM without FS vs. grid search	PSO–SVM with FS vs. grid search with F-score	PSO–DT without FS vs. C4.5 (default parameter values)	PSO–DT with FS vs. C4.5 (default parameter values)
Difference	6.894%	10.888%	7.717%	11.340%
df	9	9	9	9
t-value	3.804	6.971	4.243	6.2193
Two-tail significant	0.0042	<0.0001	0.002	<0.0001

FS means feature selection.

Table 6
Comparison between various approaches (%).

Dataset	PSO–DT with FS	PSO–SVM with FS	C4.5+SSS-GC (López et al., 2006)	C4.5+GA (López et al., 2006)	GA+SVM (Huang et al., 2007)	Hybrid C4.5/GA (Carvalho & Freitas, 2002)	Flexible NBTree (Wang & Yuan, 2005)	DTMEN (Altınçay, 2007)	EBP (Esposito et al., 1997)
Australian	90.99	91.03	83.94	83.51	86.90	N/A	61.00	85.15	84.58
German	79.28	81.62	N/A	N/A	77.92	N/A	N/A	75.86	N/A
CRX	88.77	92.19	N/A	N/A	N/A	86.37	76.7	N/A	N/A

FS means feature selection.
N/A: Approach did not use this dataset for test.

shown in Table 4. From the table, it can be observed that, no matter with or without the feature selection, the number of rules was not too many. It can be found that the proposed PSO–DT can still obtain good results without too many rules.

The paired *t*-test is used to verify the effectiveness of the proposed approaches. At confidence level $\alpha = 0.05$, the paired *t*-test performed on classification accuracy rate showed that the proposed approaches are better than grid search and DT algorithm using default parameter values as shown in Table 5.

To compare the proposed PSO–SVM and PSO–DT approaches with other approaches, three financial related datasets collected from UCI Machine Learning Repository (Hettich, 2004) are used. Three datasets are Australian, German and CRX. Other approaches are DT-related and SVM-related approaches which reported at least one classification accuracy rate of above datasets in the literature. These approaches included sequential scatter search with greedy combination (C4.5+SSS-GC) (López et al., 2006), genetic-algorithm (C4.5+GA) (López et al., 2006), and GA+SVM (Huang, Chen, & Wang, 2007), Hybrid C4.5/GA (Carvalho & Freitas, 2002), flexible NBTree (Wang & Yuan, 2005), decision trees using model ensemble-based nodes (DTMEN) (Altınçay, 2007), and error-based pruning (EBP) (Esposito, Malerba, & Semeraro, 1997). The classification accuracy rates were cited from their original studies. Among them, C4.5+SSS-GC, C4.5+GA, and GA+SVM adopted the feature selection methods in their approach. Even through there is not enough degree of freedom for the statistical test, it is clear that the proposed PSO–SVM and PSO–DT have the highest accuracy rates as shown in Table 6. It further demonstrated that the proposed approaches can be applied to the financial dataset, included the bank performance.

5. Conclusions and future research

The prediction of bank performance is an important and widely studied topic since it can have significant outcomes. Recently, SVM and DT have been applied to the problem of bank performance. This study gives evidence for the improvement of two data mining techniques in two aspects: feature selection and parameter determination, by the PSO-based approach. Although SVM has a high accuracy of prediction, the processes among them are more like a black-box. Professionals cannot take these results into their future judgments. By the tree structure obtained from the proposed PSO–DT architecture, experts can obtain the decision rules and thus make a further evaluation. This study evaluated the proposed model using the real data set. Experimental results showed that the proposed PSO–SVM and PSO–DT models were effective in searching for the beneficial subset of features and parameter values.

More studies are necessary in the future. First, the proposed PSO-based approaches may be sensitive to the parameter settings of the PSO; to perform a comprehensive study on alternative-parameter tuning policies gives room for further investigation. Second, since the proposed PSO-based approach is quite versatile, it would be worthwhile to explore the potential of this approach further using other data mining techniques. Third, the proposed approaches can be applied to other fields and to other real world problems. Finally, the ensemble strategy may be applied for increasing the performance of prediction.

Acknowledgements

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC96-2416-H-211-002.

References

Ahn, B. A., Cho, S. S., & Kim, C. Y. (2000). The integrated methodology of rough set theory and artificial neural network for business failure prediction. *Expert Systems with Applications, 18*, 65–74.

Alam, P., Booth, D., Lee, K., & Thordarson, T. (2000). The use of fuzzy clustering algorithm and self-organization neural networks for identifying potentially failing banks: An experimental study. *Expert Systems with Applications, 18*, 185–199.

Altınçay, H. (2007). Decision trees using model ensemble-based nodes. *Pattern Recognition, 40*, 3540–3551.

Altman, E. I. (1968). Financial ratios discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance, 23*, 589–609.

Altman, E. I., Haldeman, R. G., & Narayanan, P. (1977). ZETATM analysis, a new model to identify bankruptcy risk of corporations. *Journal of Banking & Finance, 1*, 29–54.

Beaver, W. H. (1966). Financial ratios as predictors of failure. *Journal of Accounting Research, 4*, 71–111.

Becerra, V. M., Galvao, R. K. H., & Abou-Seads, M. (2005). Neural and wavelet network models for financial distress classification. *Data Mining and Knowledge Discovery, 11*, 35–55.

Berrt, M. J. A., & Linoff, G. (1997). *Data mining techniques: For marketing, sales, and customer support*. Canada: John Wiley & Sons.

Burgers, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery, 2*, 121–167.

Carvalho, D. R., & Freitas, A. A. (2002). A genetic-algorithm for discovering small-disjunct rules in data mining. *Applied Soft Computing, 2*, 75–88.

Chen, Y.-W., & Lin, C.-J. (2006). Combining SVMs with various feature selection strategies. In I. Guyon, S. Gunn, M. Nikravesh, & L. A. Zadeh (Eds.), *Feature extraction, foundations and applications* (pp. 315–324). Springer.

Cielen, A., Peeters, L., & Vanhoof, K. (2004). Bankruptcy prediction using a data envelopment analysis. *European Journal of Operational Research, 154*, 526–532.

Coello, C. A. C., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 8*, 256–279.

Condition and performance of domestic banks. <http://www.cbc.gov.tw/EngHome/eeconomic/statistics/quarterly.asp>.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. England: Cambridge University.

de Andres, J., Landajo, M., & Lorca, P. (2005). Forecasting business profitability by using classification techniques: A comparative analysis based on a Spanish case. *European Journal of Operational Research, 167*, 518–542.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

Esposito, F., Malerba, D., & Semeraro, G. (1997). A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(5), 476–491.

Gehrke, J., Ganti, V., Ramakrishnan, R., & Loh, W.-Y. (1999). BOAT-optimistic decision tree construction. In *Proceedings of ACM SIGMOD international conference management of data, Philadelphia, Pennsylvania* (pp. 169–180).

Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques*. Morgan Kaufmann.

Hettich, S. (2004). *UCI machine learning repository*. School of Information and Computer Science, University of California, Irvine. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003). *A practical guide to support vector classification* (pp. 1–12). Technical report, University of National Taiwan, Department of Computer Science and Information Engineering.

Huang, Z., Chen, H., Hsu, C.-J., Chen, W.-H., & Wu, S. (2004). Credit rating analysis with support vector machines and neural networks: A market comparative study. *Decision Support Systems, 37*, 543–558.

Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*, 847–856.

Kao, C., & Liu, S.-T. (2004). Prediction bank performance with financial forecasts: A case of Taiwan commercial banks. *Journal of Banking & Finance, 28*, 2353–2368.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE conference on neural network* (Vol. 4, pp. 1942–1948).

Kim, H., & Koehler, G. J. (1995). Theory and practice of decision tree induction. *Omega, 23*, 637–652.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*, 273–324.

Lee, K. C., Han, I., & Kwon, Y. (1996). Hybrid neural network models for bankruptcy predictions. *Decision Support Systems, 18*, 63–72.

Lin, H.-T., Lin, C.-J. (2003). *A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods* (pp. 1–32). Technical report, University of National Taiwan, Department of Computer Science and Information Engineering.

Lin, F. Y., & McClean, S. (2001). A data mining approach to the prediction of corporate failure. *Knowledge-Based Systems, 14*, 189–195.

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Boston: Kluwer Academic.

López, F. G., Torres, G. M., & Batista, B. M. (2006). Solving feature subset selection problem by parallel scatter search. *European Journal of Operation Research, 169*, 477–489.

Martin, D. (1977). Early warning of bank failure a logit regression approach. *Journal of Banking & Finance, 1*, 249–276.

Min, J. H., & Lee, Y.-C. (2005). Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Systems with Applications, 28*, 603–614.

Mochón, A., Quintana, D., & Sáez, Y. (2007). Soft computing techniques applied to finance. *Applied Intelligence*. doi:10.1007/s10489-007-0051-5.

Müller, K.-R., Mike, S., Rätsch, G., Tsuda, K., & Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks, 12*, 181–201.

Noorul Haq, A., Karthikeyan, K., Sivakumar, K., & Saravanan, R. (2006). Particle swarm optimization (PSO) algorithm for optimal machining allocation of clutch assembly. *International Journal of Advanced Manufacturing Technology, 27*, 865–869.

Oleda, I., & Fernandez, E. (1997). Hybrid classifiers for financial multicriteria decision making: The case of bankruptcy prediction. *Computational Economics, 10*, 317–335.

Osei-Bryson, K.-M. (2007). Post-pruning in decision tree induction using multiple performance measures. *Computers and Operations Research, 34*, 3331–3345.

Park, C.-S., & Han, I. (2002). A case-based reasoning with feature weights derived by analytic hierarchy process for bankruptcy prediction. *Expert Systems with Applications, 23*, 255–264.

Quinlan, J. R. (1986). Introduction of decision trees. *Machine learning, 1*, 81–106.

Quinlan, J. R. (1987a). Decision trees as probabilistic classifiers. In *Proceedings of 4th international workshop machine learning, Irvine, California* (pp. 31–37).

Quinlan, J. R. (1987b). Simplifying decision trees. *International Journal of Man-Machine Studies, 27*, 221–234.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann.

Quintana, D., Saez, Y., & Mochon, A. (2007). Early bankruptcy prediction using ENPC. *Applied Intelligence*. doi:10.1007/s10489-007-0049-z.

Rastogi, R., & Shim, K. (2000). PUBLIC: A decision tree classifier that integrates building and pruning. *Data Mining and Knowledge Discovery, 4*, 315–344.

Ravi Kumar, P., & Ravi, V. (2006). Bankruptcy prediction in banks by an ensemble classifier. In *Proceedings of IEEE international conference on industrial technology, Mumbai, India* (pp. 2032–2036).

Ravi Kumar, P., & Ravi, V. (2007). Bankruptcy prediction in banks and firms via statistical and intelligent techniques-a review. *European Journal of Operational Research, 180*, 1–28.

Ryu, Y. U., & Yue, W. T. (2005). Firm bankruptcy prediction: Experimental comparison of isotonic separation and other classification approaches. *IEEE transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, 35*, 727–737.

Salzberg, S. L. (1997). On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery, 1*, 317–327.

SchÖlkopf, B., & Smola, A. J. (2002). *Learning with kernels support vector machines, regularization, optimization, and beyond*. London: MIT.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceeding of the IEEE congress on evolutionary computation* (pp. 69–73).

Shin, K.-S., Lee, T. S., & Kim, H.-J. (2005). An application of support vector machines in bankruptcy prediction model. *Expert Systems with Applications, 28*, 127–135.

Taiwan ratings. <http://www.taiwanratings.com/en/>.

Tung, W. L., Quek, C., & Cheng, P. (2004). GenSo-EWS: A novel neural-fuzzy based early warning system for predicting bank failures. *Neural Networks, 17*, 567–587.

Wang, L.-M., & Yuan, S.-M. (2005). Learning hybrid decision tree based on post-discretization strategy. *Knowledge-Based Systems*. doi:10.1016/j.knosys.2005.10.006.

West, R. C. (1985). A factor-analytic approach to bank condition. *Journal of Banking & Finance, 9*, 253–266.

Wilson, R. L., & Sharda, R. (1994). Bankruptcy prediction using neural networks. *Decision Support Systems, 11*, 545–557.