2010 International Conference on Business and Economics Research
vol.1 (2011) © (2011) IACSIT Press, Kuala Lumpur, Malaysia

Applying particles swarm optimization for support vector machines on predicting company financial crisis

Bo-Tsuen Chen
Department of Information Management
National Taichung Institute of Technology
Taichung, Taiwan (R.O.C)
s1899b109@ntit.edu.tw

Mu-Yen Chen
Department of Information Management
National Taichung Institute of Technology
Taichung, Taiwan (R.O.C)
mychen@ntit.edu.tw

Abstract—**The prediction of company financial crisis is an important and widely research issue since it could be had significant benefits and impacts. In this paper, we applied particle swarm optimization (PSO) to obtain optimized parameters setting for support vector machine (SVM) with feature selections. In addition, we also used integrated PSO with SVM approach to construct the financial crisis prediction model. We hope our proposed model will become a great analysis tool for the listed companies.**

In order to evaluate our proposed approach, the dataset collected from Taiwanese listed companies which are used as source data. Besides, the dataset is composed of 68 companies from 1996-2005, and has 544 records (8 seasons); which each records includes 37 financial and non-financial ratio indexes. This research is divided into three steps: (1) selecting features by principal component analysis (PCA) to reduce unnecessary features (2) applying PSO-SVM data mining techniques for training data, cross-validation and test data to obtain the classification accuracy rate (3) comparing with grid search for SVM and original SVM. Besides, comparing that whether do selecting features will influence the classification accuracy rate or not.

Finally, the experimental results showed that our proposed approach was effective in finding for the better parameter settings, and improve the hit ratio on predicting company financial crisis significantly. In fact, it can be found that the average classification accuracy rates are increased when the feature selection is applied. Furthermore, the average classification accuracy rate of the approach is 100% in the training subset, and be 88.98% in the test subset. It is evident that the PSO-SVM approach is as good as the grid search for SVM and original SVM.

Keywords-Particle swarm optimization; Support vector machine; Financial bankruptcy crisis

I. INTRODUCTION

Taiwanese enterprises have a complete industry supply chain, and there existed a high degree of cooperation and competition situations. Therefore, when a company went into financial crisis, it is liable to bring disaster to the upstream and downstream companies. This chain reaction will cause bounced checks, default and bankruptcy events like the butterfly effect. In order to prevent to invest the companies which have financial distress by the sponsors of investment enterprises, companies and investors, a set of predict company financial crisis model is essential.

There is considerable evidence that financial market behavior is not fully efficient and is highly nonlinear, so current analysis models cannot meet our research desires. Applying Artificial Intelligence (AI) techniques into financial crisis in recent years becomes the common prediction methods. Besides, using SVM to predict the company financial crisis also suits a common method. The results showed that the SVM method is a promising technique, performing better than other methods for bankruptcy prediction.

Consequently, this paper proposed a hybrid intelligent system by combining a PSO method on SVM approach to predict the company financial crisis. It can found that few studies have simultaneously considered the feature selection and the optimal parameter setting in predicting financial structure. Therefore, this approach will apply PSO with performing a feature selection and parameter determination, and use feature reduction methods to reduce the noisy ratios used in the prediction.

The remainder of this paper is organized as follows: Section 2 provides an overview of SVM and PSO. Section 3 describes the PSO-SVM hybrid model. Section 4 presents the experimental results from a simulate dataset. Conclusions are finally drawn in Section 5, along with recommendations for future research.

II. LITERATURE REVIEW

A. Support vector machine

Recently, a novel neural network technique, called support vector machine (SVM), was proposed by Vapnik in 1995[1]. SVM are a set of related supervised learning methods used for classification and regression. Originally, SVM have been developed for pattern recognition problems [2, 3]. Recently, with the introduction of Vapnik's ε-insensitive loss function, SVM have been extended to solve nonlinear regression estimation problems, and they exhibit excellent performance [4, 5].

To create a SVM, first introduced using a simple linear hyper-planes do classification, assuming the hyperplane formula [6] : $\langle w \cdot x \rangle + b = 0, w \in R^n, b \in R$, then finds a hyperplane function $f(x) = sign(\langle w \cdot x \rangle + b)$, w is the vector perpendicular to the hyperplane, b is the distance from the origin to the hyperplane, x is the input dataset. Ability of the classification based on training point to hyperplane distance, the distance was called margin. Vapnik [7, 8] proposed a Generalized Portrait formula as follow:

$$\max_{w,b} \min\{\|x - x_i\|: x \in \mathrm{R}^n,$$
$$\langle w \cdot x \rangle + b = 0, i = 1,2, \dots, j \quad (1)$$

It is subject to find that the maximum margin of all hyperplane would be the optimization hyperplane. Assume the training set: $X = \{(x_1, y_1), (x_2, y_2), \dots, (x_j, y_j): x_j \in \mathrm{R}^n, y_j \in \{-1, +1\}\}$, $\|w\| = 1$ and b is a fixed number, then we can express the (1) description to:

$$\langle w \cdot x_j \rangle + b > 0 \text{ for } y_j = 1$$
$$\langle w \cdot x_j \rangle + b < 0 \text{ for } y_j = -1 \quad (2)$$

Margin representation as follows: Margin $\gamma_i(w, b)$ in training point x_i represent the distance from x_i to the hyperplane, $\gamma_i(w, b) = y_i(\langle w \cdot x \rangle + b)$. Margin $\gamma_v(w, b)$ in the training set vector $v = \{x_1, x_2, \dots, x_j\}$ represent the distance from the vector v to the hyperplane. According to the definition of margin and hyperplane, we can define the Optimal Separating Hyperplance (OSH) which is in the training set v as follow [9]:

$$\text{minimize } \frac{1}{2}\|w\|^2 \quad (3)$$

There are $\langle w \cdot x_j \rangle + b \geq 1$ for $y_j = 1$, and $\langle w \cdot x_j \rangle + b \leq -1$ for $y_j = -1$. Because the OSH just admit linearly separable, no noise suppression and not allow any classification errors, however, many of the problems in the real world is not so perfect, OSH classification cannot be used alone. There is solution by adding variables $\xi_i (i = 1,2, \dots, j)$ for this problem, where ξ_i are the distance from the margin to the classification points. Consequently, the (3) can be express by $\xi_i = \max\{0, 1 - y_i(\langle w \cdot x_j \rangle + b)\}$ as follow:

$$\langle w \cdot x_j \rangle + b \geq 1 - \xi_i \text{ for } y_j = 1$$
$$\langle w \cdot x_j \rangle + b \leq -1 + \xi_i \text{ for } y_j = -1 \quad (4)$$

There are three cases in this formula, the error classification $\xi_i \geq 1$, the correct classification $0 \leq \xi_i \leq 1$ or $\xi_i = 0$. We convert the input data into feature space with kernel function, and different kernel function cause different effect of classification, so it is important for the choice of the kernel functions in SVM.

B. Particle swarm optimization

The Particle swarm optimization (PSO) algorithm was first introduced by Eberhart and Kennedy [10], it is motivated from the simulation of social behavior. It was developed by the authors comprises a very simple concept, and paradigms can be implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. Early testing has found the implementation to be effective with several kinds of problems [11].

Particle represents a potential problem solution move through a d-dimensional search space. Each particle i represents a candidate position, and they remembered the best value and the current position which had resulted in that value. The value was called *pbest.* When a particle takes the entire population as its topological neighbors, the best value is a global best and is called *gbest.* All particles can share information about the search space. The d-dimensional position for the particle i at iteration t can be represented as $x_i^t = \{x_{i1}^t, x_{i2}^t, \dots, x_{id}^t\}$. Likewise, the velocity, which is also a d-dimensional vector, for particle i at iteration t can be described as $v_i^t = \{vx_{i1}^t, v_{i2}^t, \dots, v_{id}^t\}$. Let P_{id} denote the best previous position encountered by the ith particle. P_{gd} denotes the global best position thus far. The current velocity of the d th dimension of the ith particle at iteration t is as follows [12]:

$$v_{id}^t = v_{id}^{t-1} + c_1 r_1 (P_{id}^t - x_{id}^t) + c_2 r_2 (P_{gd}^t - x_{gd}^t),$$
$$d = 1,2, \dots, D \quad (6)$$

In the above formula, $r(\,)$ is a random function in the range [0, 1], positive constant c_1 and c_2 are personal and social learning factors, and w is the inertia weight [13]. The velocity is restricted to the $[-v_{max}, v_{max}]$ range in which v_{max} is a predefined boundary value. The new position of a particle is calculated using the following formula:

$$x_{id}^t = x_{id}^t + v_{id}^t, d = 1,2, \dots, D \quad (7)$$

Unlike in genetic algorithms, evolutionary programming, and evolution strategies, in PSO, the selection operation not performed. All particles in PSO are kept as members of the population through the course run. It is the velocity of the particle which is updated according to its own previous best position and the previous best position of it companions. The particles fly with the updated velocities. PSO is the only evolutionary algorithm that does not implement survival of the fittest [14].

III. THE RESEARCH METHODOLOGY

In order to evaluate our proposed approach, the dataset collected from Taiwanese listed companies which were used as source data. Besides, the dataset is composed of 68 companies from 1996-2005, including 34 well financial position companies and 34 failed companies. The detail data description contains the past 2 seasons (136 records), the past 4 seasons (272 records), past 6 seasons (408 records) and past 8 seasons (544 records) prior to the occurrence of the financial distress. Besides, each record includes totally 37 financial and non-financial ratio indicators and lists in the Table 1.

Another pre-processing technique used in this paper is data scaling. All the data records are scaled into the range of [0, 1] as the data records include both positive values and negative values. This modification method for the data distribution will improve the prediction performance of the accuracy hit ratio. After all the datasets were scaled, the input variables were selected by the principal component analysis (PCA) and the step by step processes illustrated in the Table 1, 2 and 3. In this experiment, we adopt STATSOFT STATISTICA 9.0 software to implement PCA process. After PCA selection, the total explained variance is 75.874%, 80.962% and 96.035% for Input 37, 31, and 10 variables respectively. Therefore, there are three kinds datasets could be evaluated, including the Input 10 and 31 variables by PCA, and Input 37 variables without PCA selection. Moreover, we could also compare different feature subsets and prediction performance with these three datasets.

Hereafter, all of the four original data sets (past 2, 4, 6, and 8 seasons) were divided into two portions according to the financial position in 8:2 ratios. The former portion is used for training and cross validation, and the latter portion used for purpose of testing. For example, there are a total of 432 records in both the training dataset and cross validation dataset, 112 records in the testing dataset with the past 8 seasons (544 records), while the patterns in the training dataset is equally 216 records both of "good" and "bad" financial structure, and the patterns in the testing dataset is also equally 56 records both of "good" and "bad" financial structure.

TABLE I. PCA WITH INPUT VARIABLES

Principal component analysis

No	*Variables*	*Factor Loadings*		
		37 input	*31 input*	*10 input*
1	Acid test ratio	0.877	0.888	0.974
2	Book Value Per Share	0.744	0.757	
3	Cash Flow ratio	0.873	0.917	0.952
4	Cash Flow to Long Term Debt	0.732	0.723	0.956
5	Cash Flow to Short Term & Long Term Debt ratio	0.813	0.891	
6	Cash Flow to Total Debt	0.823		
7	Cash ratio	0.503	0.679	
8	Current Assets to Total Assets	0.871	0.876	0.973
9	Current ratio	0.833	0.841	
10	Debt to Assets	0.820	0.818	
11	Debt to Equity	0.968	0.971	
12	Dividend Payout ratio	0.578		0.992
13	Earnings Per Share	0.920	0.922	
14	Financial Leverage ratio	0.970	0.972	0.926
15	Fixed Assets to Total Assets ratio	0.772	0.786	0.994
16	Gross Margin	0.659		
17	Gross Margin Growth ratio	0.526	0.651	
18	Gross Margin to Total assets ratio	0.716		
19	Insider Holding ratio	0.631	0.681	
20	Inventory to Sales ratio	0.803	0.809	
21	Inventory to Total assets ratio	0.889	0.893	
22	Investment ratio	0.758	0.746	
23	Pretax Margin	0.816	0.823	
24	Pretax Margin Growth ratio	0.523		
25	Price-book ratio	0.462		
26	Return on Equity	0.889	0.914	
27	Return on Total Assets	0.866	0.908	0.929
28	Sales Growth ratio	0.641	0.538	0.933
29	Short Term & Long Term Debt to Book Value	0.963	0.967	0.974
30	the proportion of collateralized shares by the broad of directors	0.398	0.405	
31	Times Interest Earned	0.778	0.806	
32	Turnover rate of Accont Receivable	0.692	0.888	
33	Turnover rate of Equity	0.794	0.782	
34	Turnover rate of Fixed Assets	0.803	0.793	
35	Turnover rate of Inventory	0.757	0.777	
36	Turnover rate of Total Assets	0.824	0.826	
37	Turnover rate of Working Capital	0.790	0.851	
	Total Explained Variance (%)	75.874	80.962	96.035

The selected features and training dataset are used for building PSO-SVM, Grid-SVM and SVM models. The parameter values used in our experiments were set as following. The cognition learning factor c_1 and the social learning factor c_2 for PSO were set to 2. The number of particles and the maximum number of iterations were set to be 20 and 100, respectively. The searching range of SVM parameter C was between 0 and 100, and the search range of SVM parameter Y was the same between 0 and 100. The searching bound of Grid search method parameter C,Y were set to -8 and 8, and both of step value were set to 0.5. To ensure the robustness experiment, the value of cross validation parameter k was set to 5 folds.

IV. EXPERIMENTAL RESULTS

The comparison results obtained by the proposed PSO-SVM, Grid-SVM and SVM are shown in Table 2-10. The accuracy rate for the three models with 37 input variables datasets are shown in the Table 2-4, without any feature selection for the PSO-SVM, Grid-SVM and SVM approaches. The average classification accuracy rate of the PSO-SVM is 87.64%, while the average classification accuracy rates of the Grid-SVM is 89.65%, and the average classification accuracy rate of the SVM is 79.31%. It is evident that the PSO-SVM result without feature selection by the PCA is not better than the Grid-SVM, although there are 100% training accuracy on all the 2, 4, 6 and 8 seasons, the results also show that SVM without any parameter values determination is not better than PSO-SVM and Grid-SVM models.

However, with feature selection for all the PSO-SVM, Grid-SVM and SVM, which total explained variance of the 10 input variables are achieve 96.035% highly. The Table 8-10 shows the average cross validation, training and testing accuracy with the 5-fold results. These accuracy rates for PSO-SVM are 85.64%, 90.89%, and 74.70% respectively. The Grid-SVM accuracies are 85.95%, 90.52% and 76.86% respectively, The SVM accuracies are 78.43%, 79.12% and 76.71% respectively. The highest total explained variance for feature selection by PCA cause PSO-SVM get the lower testing classification accuracy than SVM, and Grid-SVM also get the lower test classification accuracy than the original 37 input variables dataset without any feature selection.

Thus, we selected features by PCA with the factor loadings more than 0.6, and got the total explained variance achieve 80.962% with obtained 31 variables. The results of 31 variables by proposed approaches are shown in Table 5-7. The average cross validation, training and testing accuracy of PSO-SVM are 97.76%, 100% and 88.98%, respectively. The Grid-SVM accuracies are 97.87%, 100% and 89.43%, respectively. Finally, the SVM accuracies are 81.15%, 82.96%, and 82.36%, respectively.

It is obvious that the classification accuracy could be enhanced by removing those noisy and highly correlated features, no matter the PSO-SVM, Grid-SVM or SVM approaches. But it can't obtain higher classification accuracy by removing too much variables with PCA, in order to obtain higher total explained variance. On the contrary, it has lost many of the available input variables for learning and training.

Finally, the comparison with various approaches results are shown in Table 11, it can be observed that, no matter with or without the feature selection, the PSO-SVM can effectively determine the parameter values and find a subset of features without lowering SVM classification accuracy , and approximation in Grid-SVM.

TABLE II. PSO-SVM WITH 37 INTPUT VARIABLES

Accuracy	seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	96.29%	98.61%	98.45%	98.84%	98.05%
best c	2.2337	9.7129	10.2203	4.5392	-
best g	2.2136	1.8625	2.8231	2.0162	-
Training accuracy	100% (108/108)	100% (216/216)	100% (324/324)	100% (432/432)	100%
Testing accuracy	92.85% (26/28)	85.71% (48/56)	88.09% (74/84)	83.92% (94/112)	87.64%

TABLE III. GRID-SVM WITH 37 INTPUT VARIABLES

Accuracy	seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	96.29%	98.61%	98.76%	98.84%	98.12%
best c	2	2	2	5.6569	-
best g	1	2	2.8284	2	-
Training accuracy	99.07% (107/108)	99.53% (215/216)	100% (324/324)	100% (432/432)	99.65%
Testing accuracy	96.42% (27/28)	89.28% (50/56)	88.09% (74/84)	84.82% (95/112)	89.65%

TABLE IV. SVM WITH 37 INTPUT VARIABLES

Accuracy	seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	81.48%	81.94%	79.01%	77.54%	79.99%
best c	1	1	1	1	-
best g	0	0	0	0	-
Training accuracy	85.18% (92/108)	84.72% (183/216)	81.17% (263/324)	79.62% (344/432)	82.67%
Testing accuracy	75% (21/28)	82.14% (46/56)	79.76% (67/84)	80.35% (90/112)	79.31%

TABLE V. PSO-SVM WITH 31 INTPUT VARIABLES

Accuracy	Seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	98.14%	96.29%	98.45%	98.14%	97.76%
best c	20.99	3.1549	3.8353	15.0445	-
best g	2.4414	1.2624	2.0335	3.9657	-
Training accuracy	100% (108/108)	100% (216/216)	100% (324/324)	100% (432/432)	100%
Testing accuracy	89.28% (25/28)	91.07 (51/56)	88.09% (74/84)	87.5% (98/112)	88.98%

TABLE VI. GRID-SVM WITH 31 INTPUT VARIABLES

Accuracy	Seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	98.14%	96.75%	98.45%	98.14%	97.87%
best c	4	5.6569	4	5.6569	-
best g	2	1	2	4	-
Training accuracy	100% (108/108)	100% (216/216)	100% (324/324)	100% (432/432)	100%
Testing accuracy	92.85% (26/28)	89.28% (50/56)	88.09% (74/84)	87.5% (98/112)	89.43%

TABLE VII. SVM WITH 31 INTPUT VARIABLES

Accuracy	Seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	84.25%	81.94%	79.93%	78.47%	81.15%
best c	1	1	1	1	-
best g	0	0	0	0	-
Training accuracy	85.18% (92/108)	84.25% (182/216)	82.09% (266/324)	80.32% (347/432)	82.96%
Testing accuracy	78.57% (22/28)	87.5% (49/56)	82.14% (69/84)	81.25% (91/112)	82.36%

TABLE VIII. PSO-SVM WITH 10 INTPUT VARIABLES

Accuracy	Seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	87.03%	87.03%	85.18%	83.33%	85.64%
best c	69.0077	18.6498	27.4593	100	-
best g	1.3956	16.4626	7.6532	4.2055	-
Training accuracy	91.66% (99/108)	94.90% (205/216)	89.50% (290/324)	87.5% (378/432)	90.89%
Testing accuracy	96.42% (27/28)	50% (28/56)	77.38% (65/84)	75% (84/112)	74.70%

TABLE IX. GRID-SVM WITH 10 INTPUT VARIABLES

Accuracy	Seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	88.88%	87.03%	84.56%	83.33%	85.95%
best c	256	22.6274	32	8	-
best g	0.70711	16	8	8	-
Training accuracy	91.66% (99/108)	94.90% (205/216)	90.12% (292/324)	85.41% (369/432)	90.52%
Testing accuracy	96.42% (27/28)	50% (28/56)	79.76% (67/84)	81.25% (91/112)	76.86%

TABLE X. SVM WITH 10 INTPUT VARIABLES

Accuracy	Seasons				
	2 seasons	*4seasons*	*6seasons*	*8seasons*	*Avg.*
C.V. accuracy	80.55%	83.33%	76.23%	73.61%	78.43%
best c	1	1	1	1	-
best g	0	0	0	0	-
Training accuracy	85.18% (92/108)	83.79% (181/216)	74.38% (241/324)	73.14% (316/432)	79.12%
Testing accuracy	85.71% (24/28)	60.71% (34/56)	80.95% (68/84)	79.46% (89/112)	76.71%

TABLE XI. COMPARISON WITH VARIOUS APPROACHES

Dataset	*Input 37*			
Approach	***PSO-SVM***	***Grid-SVM***	***SVM***	***Avg.***
C.V.	98.0517%	98.1289%	79.9962%	92.0589%
Training	100%	99.6528%	82.6774%	94.1101%
Testing	87.6488%	89.6577%	79.3155%	85.5407%
	Input 31			

Dataset	*Input 37*			
Approach	***PSO-SVM***	***Grid-SVM***	***SVM***	***Avg.***
Approach	***PSO-SVM***	***Grid-SVM***	***SVM***	***Avg.***
C.V.	97.7623%	97.8766%	81.1522%	92.2637%
Training	100%	100%	82.9669%	94.3223%
Testing	88.9880%	89.4345%	82.3661%	86.9295%
	Input 10			
Approach	***PSO-SVM***	***Grid-SVM***	***SVM***	***Avg.***
C.V.	85.6481%	85.9567%	78.4337%	83.3462%
Training	90.8951%	90.5286%	79.1281%	86.8506%
Testing	74.7024%	76.8601%	76.7113%	76.0913%

V. CONCLUSIONS AND FEATURE RESEARCH

The experimental results showed that our proposed approach was effective in finding for the better parameter settings, and improving the hit ratio on predicting company financial crisis significantly. In fact, it can be found that the average classification accuracy rates are increased when the feature selection is applied. Furthermore, the average classification accuracy rate of the approach is 100% in the training subset, and be 88.98% in the testing subset. It is evidenced that the PSO-SVM approach is as good as the grid search for SVM and original SVM.

In this research, with feature selection, the PSO-SVM approach yields a higher classification accuracy compared with those without feature selection. It is proven that by removing those noisy and highly correlated features is greatly benefit in this research, but when it removes too many features would cause the lower classification accuracy compared with the original reversely. However, the data mining technology or artificial intelligence are not the same methods with statistics methods; it should needs more information to learning and training, so as to obtain the better classification accuracy.

According to the literatures, most researchers extracted and sorted out the reduction variables for convenient analysis while they adopted crisis prediction model. However, these researches reveal that there is an error existed between reduction variables and original variables if the number of variables were small, that artificial intelligence or neural network would not have enough data to learning and training. As the results, the researchers must estimate whether the classification and extraction to original variables changed the real result while constructing financial crisis prediction models.

More researches could be considered in the future. First, results in this research were obtained with PSO method; however, other soft computing methods can also be applied into the SVM-based approach. Second, the experimental results obtained from other public datasets or real-world problems can be tested in the future to verify and extend this approach.

ACKNOWLEDGMENT

The authors thank the support of National Scientific Council (NSC) of the Republic of China (ROC) to this work under Grant No. NSC-98-2410-H-025-011. Moreover, we also thank for STATSOFT STATISTICA 9.0 software to support related experiments.

REFERENCES

[1] V.N. Vapnik, “The nature of statistical learning theory,” Springer, New York, 1995.

[2] B. Scholkopf, C. Burges and V. Vapnik, “Extracting support data for a given task,” in: U.M. Fayyad, R. Uthurusamy (Eds.), Proceedings of the First International Conference on Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, CA, 1995.

[3] M. Schmidt, “Identifying speaker with support vector networks,” Interface ’96 Proceedings, Sydney, 1996.

[4] K.R. Muller, J.A. Smola and B. Scholkopf, “Prediction time series with support vector machines,” Proceedings of International Conference on Arti cial Neural Networks, Lausanne, Switzerland, pp. 999–1004, 1997.

[5] K.R. Muller, J.A. Smola, G. Ratsch, B. Scholkopf and J. Kohlmorgen, “Prediction time series with support vector machines,” in: B. Scholkopf, A.J. Smola (Eds.), Advances in Kernel Methods, The MIT Press, London, England, pp. 243–254, 1999.

[6] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and other kernel-based learning methods," Cambridge University Press, Cambridge, UK, 2000.

[7] V. Vapnik and A. Chervonenkis, “A Note on one Class of Perceptrons,” Automation and Remote Control ,vol.25,1964.

[8] V. Vapnik and A. Lerner, “Pattern Recognition using Generalized Portrait Method,” Automation and Remote Control , vol.24,1963.

[9] Y.C. Liu, "The applications of support vector machines in financial prediction," Department of Computer Science and Information Management, Providence University, unpublished master thesis, 2004.

[10] R.C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan), IEEE Service Center, Piscataway, NJ, pp. 39-43, 1995.

[11] J. Kennedy and R. Eberhart, “Particle swarm optimization.” Proc. IEEE International Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995, in press.

[12] S.W. Lin, K.C. Ying, S.C. Chen and Z.J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," Expert Systems with Applications 35, pp.1817-1824, 2008.

[13] P.J. Angeline, “Using selection to improve particle swarm optimization,” IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.

[14] R.C. Eberhart and Y.H. Shi, “Comparison between genetic algorithms and particle swarm optimization,” 1988 Annual Conference on Evolutionary Programming, San Diego, 1988.