# A Combination of Models for Financial Crisis Prediction: Integrating Probabilistic Neural Network with Back-Propagation based on Adaptive Boosting

**Lu Wang, Chong Wu**

*School of Economics and Management, Harbin Institute of Technology,*
*West Str. 92, Nangang District*
*Harbin, 150001, P. R. China*
*E-mail: wanglu_hit@163.com, wuchong_hit@163.com*

Received 11 May 2015

Accepted 12 December 2016

**Abstract**

It is very important to enhance the accuracy of financial crisis prediction (FCP). Because the traditional probabilistic neural network (PNN) has some deficiencies, such as the difficult estimation of parameters and the high computational complexity, this paper proposes a new combination model, which combines back-propagation (BP) with PNN on the basis of adaptive boosting algorithm, to predict financial crisis. The BP algorithm is introduced to modify weights and smoothing parameters of PNN. In process of constructing BP-PNN models, the training set is divided into study and training samples to save the computational time. And the trained models are regarded as weak classifiers. Then these weak classifiers are integrated to constitute a stronger classifier by adaboost algorithm. To verify the superiority of the new model in terms of FCP, this article uses financial data of Chinese listed companies from Shenzhen and Shanghai Stock Exchange, and compares with adaboost BP, PNN and support vector machine models. The result shows that the new model has the highest prediction accuracy. Therefore, the new combination model is an excellent method to predict financial crisis.

*Keywords*: Financial Crisis Prediction, Probabilistic Neural Network, Back-Propagation, Adaptive Boosting.

## 1. Introduction

The company will face a dangerous state when it suffers from financial crisis. The financial crisis includes low liquidity, inability to pay debts or dividend, substantial and continual reduction in profitability and many other aspects of financial information. The serious financial crisis brings managers, employees, investors and government to heavy financial loss. Therefore, financial crisis prediction (FCP) has important meaning for enterprise risk management. FCP is to predict whether one firm will encounter financial crisis based on the current financial condition by mathematics theory, statistical method or intelligent models. The task of FCP mainly belongs to one kind of binary classification problem, which is to distinguish healthy firms from crisis firms. For the model, the prediction accuracy is an important factor reflecting model validity in real-world life. One model possesses the practical application value only if it has good enough prediction accuracy. Therefore, it is the main problem on FCP that how to enhance the prediction accuracy of models.

Beaver was the first scholar who used a univariate model to predict enterprises bankruptcy[1]. Altman attempted to use multiple discriminant analysis (MDA) method to constitute the Z-score model to predict business failure[2]. Balcaen and Ooghe summarized and contrasted the classical statistical methodologies in business failure field[3]. These techniques included univariate analysis, MDA, risk index models and conditional probability models. Ohlson used the logistic model for FCP[4]. Zavgren used factor analysis to acquire the input variables on the basis of this research method[5]. But these models require data to content strict hypothesis condition, which are unsuitable to financial data.

With the development of artificial intelligence techniques, the artificial neural network (ANN) has been widely applied to FCP. Odom and Sharda applied ANN to predict business failure and compared with the MDA model[6]. Becerra et al. forecasted business failure of British companies by wavelet neural network models[7]. Li et al. used the improved back-propagation neural network (BPNN) to predict financial crisis[8]. Support vector machine (SVM) is a new intelligence method which can obtain ideal results in the terms of FCP[9-12]. There are other prediction methods, such as fuzzy sets, rough sets， genetic algorithms and so on. Chaudhuri and De employed fuzzy support vector machine for bankruptcy prediction problem[13]. Trabelsi et al. made classification based on rough sets[14]. Acosta-Gonzalez et al. employed genetic algorithms to forecast the firm failure for the Spanish building industry[15]. Moreover, the ensemble method could effectively enhance the prediction accuracy[16]. Sun and Li attempted to employ combination of multiple classifiers to predict business failure[17]. They used the data of Chinese listed companies to test this model and acquired the conclusion that combination of multiple classifiers had better prediction results than the single classifiers. Nanni and Lumini compared the performance of ensemble classifiers with other models in terms of bankruptcy prediction[18]. It is the conclusion that the ensemble system performed better. Tsai et al. made a comprehensive research about the ensemble classifier[19]. They respectively integrated multilayer perceptron neural networks, SVM, and decision trees based on bagging and boosting algorithm. The real-world datasets of Taiwan bankruptcy companies was used to make an experiment and obtained good performance. Therefore, it is very clear that combination models have good prediction ability.

As an excellent classifier, the probabilistic neural network (PNN) is applied to the domain of prediction[20]. Wang et al. employed traditional PNN model and BP neural network model for FCP[21]. Ticknor proposed a Bayesian regularized artificial neural network to test financial market situation[22]. Khashei et al. combined autoregressive integrated moving average (ARIMA) with probabilistic neural networks (PNNs) to obtain higher accuracy than traditional ARIMA models[23]. Adaptive boosting (AdaBoost) is generally regarded as one efficient integrated algorithm. Zhou employed the Adaboost algorithm to constitute the ensemble model to predict financial distress for Chinese listed companies[24]. Heo and Yang certified that AdaBoost method could obtain more excellent accuracy for the financial risk prediction of Korean construction companies[25]. Nie et al. presented a probability estimation model based on AdaBoost algorithm[26]. Zhou solved the problem of multiclass classification for Chinese listed companies by Adaboost method[27].

Although the PNN model is widely applied, the deficiency still exists, such as the difficult estimate of parameters, the high computational complexity, etc. In addition, the combination of prediction models can greatly enhance the classification accuracy. Therefore, the present paper puts forward to a new combination of prediction models for FCP, which integrates back-propagation with PNN (BP-PNN) on the basis of AdaBoost algorithm. First, this paper uses back-propagation algorithm to improve classical PNN model, which can adjust weights and parameters of PNN and enhance prediction accuracy. What's more, it is creatively mentioned that the training sample set are, during the training process, divided into study and training samples in order to reduce space complexity and computational time. Then, BP-PNN models are regarded as weak classifiers. The stronger classifier is made up of these weak classifiers by the AdaBoost algorithm. In the experiment section, this paper uses the financial data of listed companied in China and makes a comparison with the traditional prediction models to testify the superiority of this proposed model. It is the conclusion that the proposed model has the best performance. The paper is organized as follows: Section 2 introduces classical PNN. Section 3 improves the PNN with back-propagation, integrates BP-PNN models by Adaboost algorithm, and discusses the advantage of the combination model. Section 4 does empirical study and makes comparative analysis. Conclusion is provided in Section 5.

## 2. PNN Model

ANN is one operational model that is constituted by plenty of interconnected nodes. Every node stands for a specific activation function, and every two nodes are connected by weighted method. The different type of ANN mainly depends on different connection methods, different weight values and different activation functions. ANN includes the perceptron neural network, the BP neural network and so on. Among them, the PNN model is one classical ANN model. It is one multilayer neural network in which every layer could be respectively operated in parallel. But the PNN model is different from the universal ANN model. It has the structure of four layers and adopts activation functions which derive from statistical approach instead of general S-type activation functions. Hence, the Bayesian classification and Parzen window theorem are the basic method for this model. One example is to explain the Bayesian classification theorem. For one classification problem, $x^{(m)} = [x_1, x_2, ..., x_n]$ (m is number of samples, n is spatial dimension) are input vectors, which belong to several different classes (1,2,…,k). The prior probability that a sample belongs to the *kth* class is $h_k$, and the cost that a sample is misclassified is $c_k$, and the probability density function is $f_k(\mathbf{x})$ . Bayesian classification theorem is expressed as follows:

$$h_i c_i f_i(\mathrm{x}) > h_j c_j f_j(\mathrm{x}) \qquad \forall j \neq i \tag{1}$$

So the sample should belong to the *ith* class. But it is the hardest problem to find the corresponding probability density function $f_k(\mathbf{x})$ . The Parzen window method attempts to solve this problem. This method uses a supervised training set to estimate probability density functions. Gaussian function is often used. So probability density function is expressed as follows:

$$f_j(X) = \exp\left[-\frac{\left(X - X_j\right)^2}{2\sigma^2}\right] \tag{2}$$

where $X$ is an independent vector, $X_j$ is a central vector from training set, and $\sigma$ is the smoothing parameter or window width that represents standard deviation.

PNN model is a four-layer neural network which includes input layer, pattern layer, summation layer and output layer. The concrete network architecture is shown in Fig. 1.

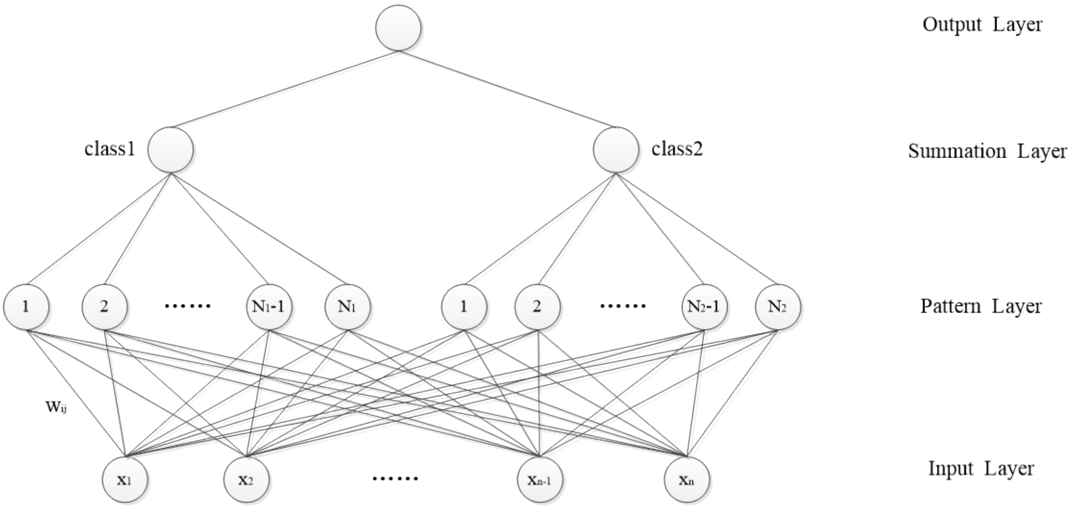

Fig. 1 Architecture of PNN contained two classes

Because FCP belongs to one kind of binary classification problem, this article presents two classes in the network architecture. In the input layer, n nodes express one input vector with n-dimensionality, which are fully connected with nodes in the next layer. In the pattern layer, one node represents one training sample and all nodes are segmented into two classes. Every node and the input vector are calculated by probability density function to obtain the activation of each node. In the summation layer, two nodes respectively accumulate outputs of nodes of the previous layer in the same class. The node which has the bigger output value is passed to the next layer. In the output layer, there is one neuron that outputs discrete value 1 or -1 (or 0) representing the classification result.

PNN has many excellent performances, such as rapid convergence, easy training, stable neuron number and so on. This model has been widely applied to non-linear filter, pattern recognition and probability density estimation. But there are many disadvantages in this model, such as the difficult estimate of parameters and the high computational complexity. FCP uses financial indicators to predict economic conditions of one firm. The financial indicators are one kind of high-dimensional and complicated data. PNN is difficult to deal with financial indicators directly. Hence, PNN should be improved to obtain adjustable parameters and reduce space complexity and computational time.

## 3. The Proposed Combination Model

To solve PNN shortages, the present paper integrates BP algorithm with PNN model. To enhance prediction accuracy, the AdaBoost algorithm is adopted to integrate the improved model. First, this section introduces how to improve the PNN model with BP algorithm. Then, it illustrates the BP-PNN model on the basis of the AdaBoost algorithm. Finally, it lays emphasis on the advantage of the combination model.

### *3.1. Improving PNN model with BP algorithm*

The BP neural network has two stages which are feed forward of input and back propagation of errors[28]. Based on the idea of back propagation, this article proposes one model combining back-propagation algorithm and PNN model, which is named as back-propagation + probabilistic neural network (BP-PNN). In the feed forward stage, PNN model is employed to calculate input data. In the back propagation stage, the error between the actual and desired output is transmitted from the last layer to the first layer in order to update weights and smoothing parameters. The concrete network architecture of BP-PNN model is showed in Fig. 2.

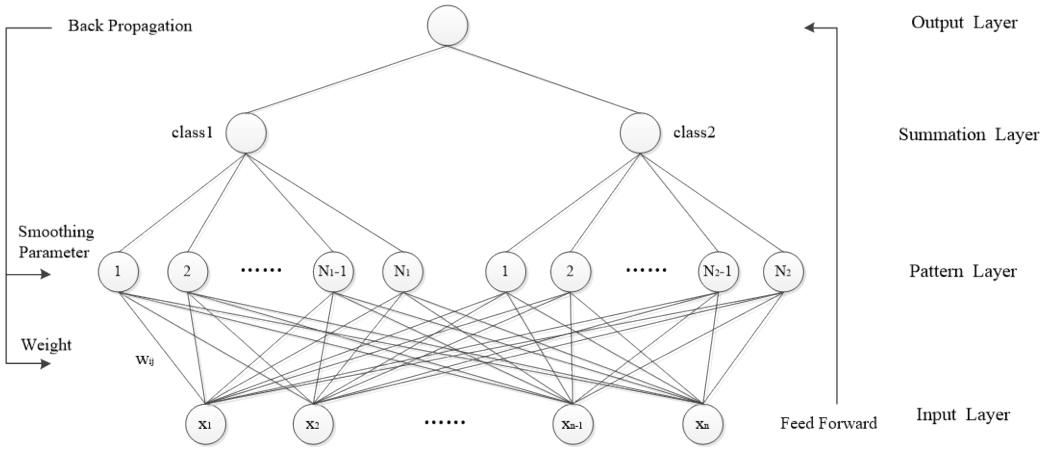

Fig. 2 BP-PNN Architecture

where the node number of pattern layer is $N$ ( $N = N_1 + N_2$ ).

As shown in the Eq.(2), we can acquire probability density function $f_j(X)(\forall j = 1,2,...,N)$ of the *jth* node in the pattern layer. $F = [f_1(X), f_2(X),..., f_j(X),...f_N(X)]$ represents one vector which is constituted by all of probability density functions in the pattern layer. $D$ is one $N \times 2$ matrix that means to which class node of the summation layer nodes of the pattern layer should belong. $D_{N\times 2} = [d_{ij}](\forall i = 1,2,...,N\,;\ \forall j = 1,2)$ . The expression of classification vector $S$ in the summation layer is shown as follows:

$$S = F \cdot D = [f_1(X), f_2(X),..., f_j(X),...f_N(X)] \cdot [d_{ij}]_{N\times 2} \quad (3)$$

where $S$ is one $1\times 2$ row vector. $S$ is normalized as follows:

$$\overline{S} = S / \left(S \cdot V^T\right) \quad (4)$$

where $V$ is one (1,1)-vector. The actual classification result of one sample is expressed by one vector $S'$ . In this vector, the *ith* element is 1 to express sample's class and the other element is 0. Then, the error function of this sample is showed as follows:

$$E = 1 - \overline{S} \cdot \left(S'\right)^T \quad (5)$$

The error value is between 0 and 1. When $\overline{S}$ is more close to $S'$ , $E$ is more close to 0. Specially, $\overline{S} = S'$ , $E = 0$ . Putting Eq.(3) and (4) to Eq.(5), we can obtain the specific error function as follows:

$$E = 1 - \frac{F \cdot D \cdot \left(S'\right)^{T}}{F \cdot D \cdot V^{T}} \tag{6}$$

(1) The weight modification

As the PNN model, $w_{ij}$ expresses one weight between the *ith* node in the input layer and the *jth* node in the pattern layer. The concrete expression is shown as follows:

$$w_{ij} = x_i / (\sum_{d=1}^{n} {x_d}^2)^{\frac{1}{2}} \tag{7}$$

where $x_i$ represents the *ith* variable of an input vector. Therefore, $X$ and $X_j$ in the Eq.(2) should be separately expressed as follows:

$$X = \sum_{i=1}^{n} x_i \cdot w_{ij} \tag{8}$$

$$X_j = \sum_{i=1}^{n} x_i' \cdot w_{ij} \tag{9}$$

where $x_i$ is the *ith* variable of n-dimensional input vector, $x_i'$ is the *ith* variable of the *jth* training sample vector. Because of $F=[f_1(X), f_2(X), ..., f_j(X), ...f_N(X)]$ and the definition of Eq.(8) and (9), we can build the update expression of weights.

$$\Delta w_{ij} = -\beta \cdot \frac{\partial E}{\partial w_{ij}} = -\beta \cdot \frac{\frac{\partial F}{\partial w_{ij}} \cdot D \cdot \left(S'\right)^{T} \cdot F \cdot D \cdot V^{T} - F \cdot D \cdot \left(S'\right)^{T} \cdot \frac{\partial F}{\partial w_{ij}} \cdot D \cdot V^{T}}{\left(F \cdot D \cdot V^{T}\right)^{2}} \tag{10}$$

where $\beta$ is the learning rate value. As the *jth* element in $F$, we can obtain the update expression of weights based on Eq.(1) as follows:

$$\frac{\partial f_j}{\partial w_{ij}} = -\frac{w_{ij} \cdot \left(x_i - x_i'\right)^2}{{\sigma_j}^2} \cdot \exp\left[ -\frac{\sum_{p=1}^{n} {w_{pj}}^2 \cdot \left(x_p - x_p'\right)^2}{2{\sigma_j}^2} \right] \tag{11}$$

Therefore, we can obtain the following vector.

$$\frac{\partial F}{\partial w_{ij}} = \left( \frac{\partial f_1(X)}{\partial w_{ij}}, \frac{\partial f_2(X)}{\partial w_{ij}}, ..., \frac{\partial f_j(X)}{\partial w_{ij}}, ..., \frac{\partial f_n(X)}{\partial w_{ij}} \right) \tag{12}$$

Then, it is easy to obtain the final update weight by putting Eq.(11) and (12) to Eq.(10).

(2) The smoothing parameter modification

Similarly, we can build the update expression of smoothing parameters.

$$\Delta \sigma_j = -\beta \cdot \frac{\partial E}{\partial \sigma_j} = -\beta \cdot \frac{\frac{\partial F}{\partial \sigma_j} \cdot D \cdot \left(S'\right)^{T} \cdot F \cdot D \cdot V^{T} - F \cdot D \cdot \left(S'\right)^{T} \cdot \frac{\partial F}{\partial \sigma_j} \cdot D \cdot V^{T}}{\left(F \cdot D \cdot V^{T}\right)^{2}} \tag{13}$$

Therefore, as the *jth* node of the pattern layer, the specific update expression is shown as follows:

$$\frac{\partial f_j}{\partial \sigma_j} = \frac{\sum_{p=1}^{n} {w_{pj}}^2 \cdot \left(x_p - x_p'\right)^2}{{\sigma_j}^3} \cdot \exp\left[ -\frac{\sum_{p=1}^{n} {w_{pj}}^2 \cdot \left(x_p - x_p'\right)^2}{2{\sigma_j}^2} \right] \tag{14}$$

We can obtain the following vector.

$$\frac{\partial F}{\partial \sigma_j} = \left( \frac{\partial f_1(X)}{\partial \sigma_j}, \frac{\partial f_2(X)}{\partial \sigma_j}, ..., \frac{\partial f_j(X)}{\partial \sigma_j}, ..., \frac{\partial f_n(X)}{\partial \sigma_j} \right) \tag{15}$$

The parameter modification of BP-PNN model has been completed. It is another important problem to reduce space complexity and computational time. As the same as the PNN model, the fewer nodes the pattern layer has, the less time this new model spends. However, the number of nodes in the pattern layer equals the number of samples in the training set. In another words, if one training set has fewer samples, this model can save more time. Because of the introduction of BP algorithm, this model needs not to be trained by all of training samples. Hence, this article divides the training set into study and training samples. The study samples are used to confirm the node number of the pattern layer and initial weights in order to create the original PNN model. The training samples are adopted to revise weight value and smoothing parameter by the back-propagation algorithm above. Therefore, it can save the computational time that the training set is divided into study and training samples. It is means that the ratio between study and training samples affects the computational time directly. It is easy to make poor performance for this model if the ratio is too low, whereas the computational time can't be saved and this method loses practical significance. This paper select the 3:7 ratio to distribute study and training samples. The reason is that such ratio can not only conform to the training characteristic of BP algorithm, but meet the selection rule for two classes of samples in extensive studies. In the empirical study, the comparison of different ratio is provided to testify the superiority of this ratio.

The concrete process is that, first, the training set are classified according to healthy and unhealthy companies. Then, 30% of data are randomly selected from every classification to be study samples, and the others are training samples. This process can help the balance of every category for study and training samples.

### 3.2. The AdaBoost Algorithm with BP-PNN

AdaBoost algorithm is derived from Boosting algorithm. This algorithm can integrate some weak classifiers to construct one stronger classifier to enhance prediction accuracy. Its adaptivity can be shown that the samples which are misclassified by the previous weak classifier are drawn more attention, and the weighted samples are used to train the next weak classifier again. As the same time, one new weak classifier is joined every time until this model satisfies some specific conditions. The concrete realizing process is described: every input sample is assigned one equal weight. Then, if one sample is misclassified, its weight should become bigger. If not, its weight should become smaller. The samples whose weights are modified are to train next weak classifier. Last, the strong classifier is constituted by all weak classifiers.

This article proposes one new model which integrates BP-PNN models by adaBoost algorithm. The framework of the new model is shown in Fig. 3. First, the raw data are preprocessed to reduce the dimensionality and simplify the complexity of raw data. Then, the processed data are divided into training and testing set by cross validation method to increase the number of samples. Next, the training set is separated into study and training samples to train the BP-PNN weak classifier. The weight of weak classifier is calculated according to the error rate after the completion of the training. Besides, the weights of the data are updated. If one sample is misclassified, its weight is increased, and the next weak classifier pays more attention to this sample; otherwise, its weight is decreased. After all BP-PNN weak classifiers finish the training, the stronger classifier is constructed. Finally, the testing set is calculated by the stronger classifier to obtain the prediction result.

The concrete calculation process of the proposed model can be summarized as follows:

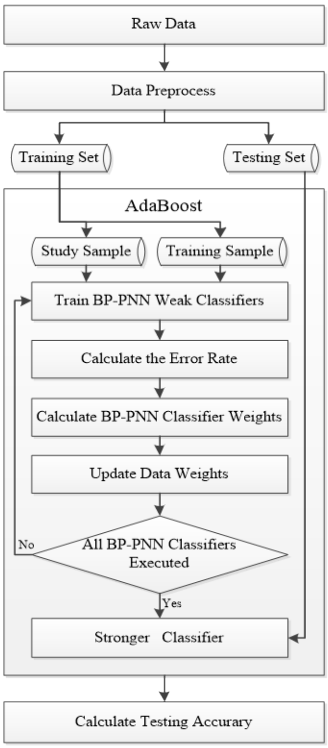

Fig. 3 Framework of the Proposed Model.

Step 1: The training set $S=\{(X_1,Y_1),\dots,(X_i,Y_i),\dots,(X_n,Y_n)\}$ is given, where $X$ represents an input vector and $Y$ is a (0 or 1) vector expressing classification results.

Step 2: The weights of training samples are initialized as $D_1(i)=1/n$.

Step 3: The max iteration time is set as T, $t=1,2,\dots,T$.

Step 3.1: The model is trained by the study and training samples with weights $D_t$ to construct a BP-PNN classifier $h_t$, its classification outputs are $h_t=\{h_t(1),h_t(2),\dots,h_t(n)\}$.

Step 3.2: According to prediction results, the total error is calculated, $e_t=\sum_{h_t(i)\neq Y} D_t(i)$.

Step 3.3: The iteration can't be stopped until $e_t$ is enough small.

Step 3.4: Calculate weight of the BP-PNN weak classifier $h_t$, $\alpha_t=\dfrac{\ln((1-e_t)/e_t)}{2}$.

Step 3.5: The weight $D_t$ can be updated as follows:

$D_{t+1}(i)=\frac{D_t(i)}{Z_t}\times\begin{cases}\beta_t & h_t(X_i)=Y_i \\ 1 & other\end{cases}$ , where $\beta_t=e_t/(1-e_t)$ , $Z_t$ is standardization constant.

Step 4: The stronger classifier is obtained, $H(x)=sign\left(\sum_{t=1}^{T}\alpha_t\cdot h_t\right)$.

Then, the final result is one positive number, this sample belongs to one classification. Inversely, this sample belongs to another classification.

### 3.3. The Advantage of the Combination Model

The proposed model considers BP-PNN as a weak classifier and uses adaboost algorithm to build a stronger classifier, which combines the advantage of the PNN model, BP and adaboost algorithm.

The BP-PNN model combines the feature of the PNN and BP algorithm: The PNN model can deal with deficiency or noise data set, decide to the model structure, and easily confirm the node number of the middle layer, which solve problems for the BP algorithm. The BP algorithm can modify the parameters and weight values, which can overcome simplex learning-method of the PNN model and enhance the accuracy. In addition, the method that the training set is divided into study and training samples can save the computational time.

The adaBoost algorithm demands that the accuracy of weak classifiers are merely better than estimation value, which may further save computational time by reducing the training iterations. The accuracy can be obviously enhanced because of the advantage of the adaBoost algorithm.

In addition, the over-fitting problem widely exists in the intelligence algorithm. But this proposed model can't be troubled in this problem. There are three reasons that, first, the data can be reduced the dimensionality by preprocessing to simplify the complexity of data, which can't result in a complicated classifier for dealing with the data. Second, this model adopts cross validation to increase the number of trained samples, which can prohibit the over-fitting situation that too few samples induce. Third, as the weak classifier, the BP-PNN is one optimization model with the simple structure. And many optimization models are integrated by adaboost algorithm, which can't create the over-fitting situation to which one singular sample or one weak classifier lead.

## 4. Empirical study

### 4.1. Selection of sample companies and financial indicators

If Chinese listed companies had negative net profit in two consecutive years or deliberately announced the false financial condition, they would be specially treated by China Securities Supervision and Management Committee (CSSMC). And ST is added to the front of such companies' names in the stock exchange. According to the principle of the same time, the same industry and the similar capital scale, 200 samples, which are the manufacture industry, are selected from Shenzhen and Shanghai Stock Exchange from 2010 to 2012. These samples contain 100 normal companies and 100 ST firms.

Table 1 Financial Indicators' Definition and Category

| Category | Variables | Definition |
|---|---|---|
| Activity Ability | $X_1$ | Inventory turnover ratio |
| | $X_2$ | Receivables turnover ratio |
| | $X_3$ | Current assets turnover |
| | $X_4$ | Fixed assets turnover |
| | $X_5$ | Total assets turnover |
| Solvency | $X_6$ | Current ratio |
| | $X_7$ | Quick ratio |
| | $X_8$ | Debt to total assets ratio |
| | $X_9$ | Cash flow to current liability |
| Profitability | $X_{10}$ | Net assets income rate |
| | $X_{11}$ | Return on assets ratio |
| | $X_{12}$ | Net capital income rate |
| | $X_{13}$ | Net profit margin of sales |
| | $X_{14}$ | Earnings per share |
| Growth ratio | $X_{15}$ | Growth ratio of income per share |
| | $X_{16}$ | Growth ratio of total assets |
| | $X_{17}$ | Growth ratio of operating income |
| | $X_{18}$ | Growth ratio of operating profit |
| | $X_{19}$ | Growth ratio of total profit |
| | $X_{20}$ | Growth ratio of net profit |

The selection of financial indicators need not only directly reflect the financial situation, but also consider the dimensionality of sample data. Hence, 20 financial indicators are selected, which are showed in Table 1. Some financial indicators have mutual effect to reflect one aspect of financial situation. As shown in Table 1, these financial indicators reflect 4 important aspects which are activity ability, solvency, profitability and growth ratio.

### *4.2. The preprocessing of financial data*

Although such financial indicators can accurately reflect financial situation, so many variables increase computational time and easily obtain poor performance. Furthermore, there are strong or weak correlations among these indicators, which increase the probability of repeated information and make many unnecessary troubles. We use factor analysis method to solve these problems. Factor analysis method can use a few comprehensive variables to stand for original data variables based on the loss of the least information. These comprehensive variables are named as common factors. Therefore, we adopt the factor analysis to reduce the complexity of the calculation and enhance the accuracy.

In this paper, we use software SPSS 19.0 to run factor analysis method to obtain common factors from 20 financial indicators of 200 samples. First, the correlations among these indicators are examined by KMO and Bartlett's test in order to determine whether the data were appropriate for factor analysis. As shown in Table 2, the value of KMO is 0.691, which is bigger than 0.5. It means that partial correlations are enough strong for these indicators. For Bartlett's test of sphericity, the value of Sig. is 0.000 (<0.05), which signifies that strong correlations exist among these indicators. Therefore, these sample data are suitable for factor analysis.

Table 2 KMO and Bartlett's Test

| Kaiser-Meyer-Olkin Measure of Sampling Adequacy. | | .691 |
|---|---|---|
| Bartlett's Test of Sphericity | Approx. Chi-Square | 5067.647 |
| | df | 190 |
| | Sig. | .000 |

Next, total variance explained is obtained in Table 3. 5 factors' eigenvalues are greater than 1, and the cumulative rate of contribution is 82.838%, which is more than 80%. Therefore, 5 common factors are selected, which contain most of the main financial information.

5 common factors respectively include different financial indicators. The first common factor $F_1$ represents profitability which is explained by $X_{10}$, $X_{11}$, $X_{12}$, $X_{13}$, $X_{14}$. The second common factor $F_2$ represents growth ability which is explained by $X_{15}$, $X_{16}$, $X_{17}$, $X_{18}$, $X_{19}$, $X_{20}$. The third common factor $F_3$ represents long-term activity ability which is explained by $X_1$, $X_4$. The fourth common factor $F_4$ represents repay debt ability which is explained by $X_6$, $X_7$, $X_8$, $X_9$. The fifth common factor $F_5$ represents short-term activity ability which is explained by $X_2$, $X_3$, $X_5$. These 5 common factors are regarded as input variables of the model.

### *4.3. Experiment design*

This article adopts the 10-folds cross validation to construct training set and testing set. First, normal and ST companies are averagely divided into 10 portions respectively. The random 5 portions are selected as training set and the others are testing set, which can guarantee that the training set has 50 normal companies and 50 ST firms, and testing set includes 50 normal companies and 50 ST firms. Then, during the training process, normal and ST companies, respectively, are divided into study and training samples according to the 3:7 ratio. Besides, 10 weak classifiers are used to construct prediction model, and every model adopts 100 iterations or one standard, that the gradient of an trained parameter is less than 1‰, to gradually modify weights and smoothing parameters.

The 2:8 and 4:6 ratios between study and training samples are made comparative analysis in order to verify the performance of the 3:7 ratio. In addition, because the new model is on the basis of BP-PNN and adaBoost algorithm, adaBoost BP, PNN and SVM are selected as contrast models. All of models are implemented in the MATLAB R2012b package software.

### *4.4. Result and comparison*

#### *4.4.1. Results on Ada BP-PNN model*

This article distributes study and training samples according to the 3:7 ratio. To testify the superiority of such ratio, another two kinds of ratios, which are 2:8 and 4:6 ratios, are made comparison analysis. All of the prediction performances are shown in the Fig. 4-6, which express results of 2:8 ratio, 3:7 ratio and 4:6 ratio in sequence. The X-axis represents the number of sample companies, and Y-axis represents classification result. Y-axis contains two values that are 1 and 0, which respectively stand for healthy companies and ST companies. In figures, the red asterisk (*) stands for the classification of sample companies in fact, and the blue round (o) expresses prediction results. The sample company is correctly classified if two signs coincide. Otherwise, it is misclassified.

The detailed results of prediction performances and computational time are listed in the Table 4. For the 2:8 ratio, 45 normal companies and 45 ST companies are correctly classified during the training stage. It has the total accuracy of 90%. In the testing stage, 40 companies and 41 ST companies are correctly classified and the total accuracy is 81%. The averaged running time is 13.973s. For the 3:7 ratio, 47 normal companies and 46 ST companies in the training set are correctly classified, and 39 normal companies and 44 ST companies in the testing set are correctly classified. The total accuracy of training and testing, respectively, are 93% and 83%. It spends 15.073s on averaged computational time. For the 4:6 ratio, 47 normal companies and 46 ST companies are correctly predicted during the training stage, and 40 normal companies and 43 ST companies are correctly predicted for the testing set. 92% and 83% are the total accuracies of training set and testing set respectively. The averaged running time is 25.486s.

Table 3 Total Variance Explained

| Component | Initial Eigenvalues | | | Extraction Sums of Squared Loadings | | | Rotation Sums of Squared Loadings | | |
|---|---|---|---|---|---|---|---|---|---|
| | Total | % of Variance | Cumulative % | Total | % of Variance | Cumulative % | Total | % of Variance | Cumulative % |
| 1 | 7.052 | 35.258 | 35.258 | 7.052 | 35.258 | 35.258 | 5.787 | 28.934 | 28.934 |
| 2 | 3.375 | 16.874 | 52.132 | 3.375 | 16.874 | 52.132 | 3.020 | 15.099 | 44.033 |
| 3 | 2.535 | 12.677 | 64.809 | 2.535 | 12.677 | 64.809 | 2.997 | 14.987 | 59.021 |
| 4 | 1.863 | 9.317 | 74.126 | 1.863 | 9.317 | 74.126 | 2.506 | 12.529 | 71.550 |
| 5 | 1.742 | 8.711 | 82.838 | 1.742 | 8.711 | 82.838 | 2.258 | 11.288 | 82.838 |
| 6 | .899 | 4.496 | 87.333 | | | | | | |

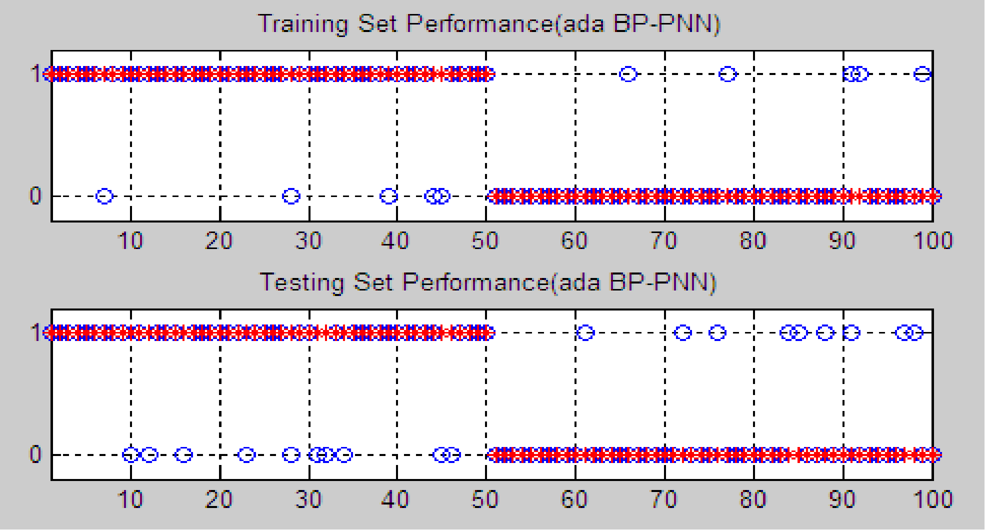

Fig. 4 Performance of Ada BP-PNN Model According to the 2:8 Ratio

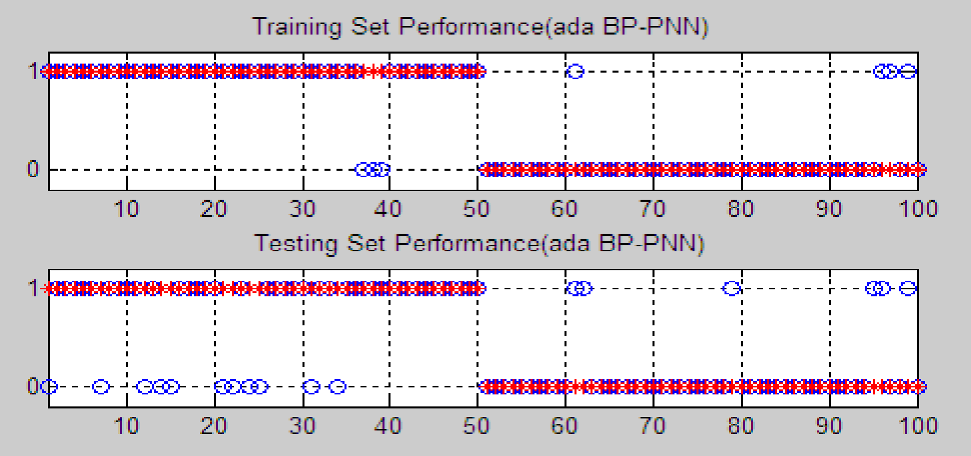

Fig. 5 Performance of Ada BP-PNN Model of the 3:7 Ratio

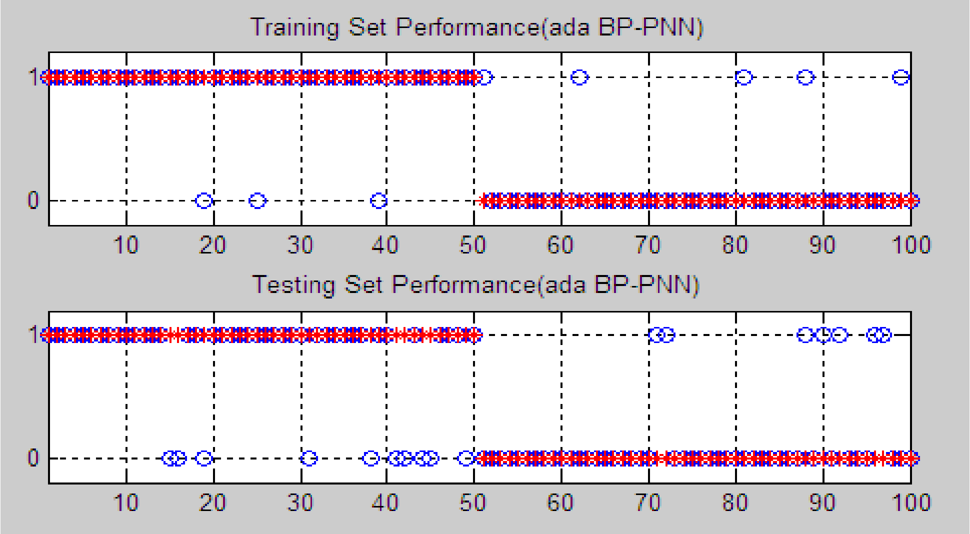

Fig. 6 Performance of Ada BP-PNN Model of the 4:6 Ratio

From the prediction results of different ratios, although the 3:7 ratio needs a litter more time than the 2:8 ratio, the 3:7 ratio has better performance. That is the reason that the node number of pattern layer should not be too few. Although there are about the same accuracies between 3:7 and 4:6 ratios, the latter needs more computational time. Therefore, the 3:7 ratio are the most suitable allocation proportion.

Besides, 18 different financial indicators are selected from 200 sample companies that are employed by this article to make comparative analysis. These indicators are preprocessed by the same method, and are calculated by the proposed model. The prediction accuracies of training and testing set are 86% and 79% respectively, which are lower than the accuracy in Table 4.

Table 4 Prediction Performance of Ada BP-PNN Model

| Ratio | Description | Training Set | | Testing Set | | Averaged Time |
|---|---|---|---|---|---|---|
| | | Quantity | Accurate number | Quantity | Accurate number | |
| 2:8 | Normal Companies | 50 | 45 | 50 | 40 | 13.973s |
| | ST Companies | 50 | 45 | 50 | 41 | |
| | Accuracy | 90% | | 81% | | |
| 3:7 | Normal Companies | 50 | 47 | 50 | 39 | 15.073s |
| | ST Companies | 50 | 46 | 50 | 44 | |
| | Accuracy | 93% | | 83% | | |
| 4:6 | Normal Companies | 50 | 47 | 50 | 40 | 25.486s |
| | ST Companies | 50 | 45 | 50 | 43 | |
| | Accuracy | 92% | | 83% | | |

### *4.4.2 Comparison between Ada BP-PNN model and other classifiers*

In order to testify whether the proposed model can enhance the accuracy, this section uses three classifiers, which are the Ada BP, PNN and SVM model, to make comparative analysis. Their concrete prediction performances are shown in the Fig. 7-9. Every figure contains the outcome of training set and testing set.

In this article, the ada BP model adopts back-propagation as the weak classifier, and the SVM model uses the radial basis function as the kernel function. It respectively shows the prediction performances and computational time of ada BP model, PNN model and SVM model in Table 5. It is very clear that, for the training set or testing set, the ada BP model has the greatest prediction accuracies, which are 87% and 80% respectively. The next one is SVM model, which are 87% and 71% respectively. The last one is PNN model, which are 76% and 56% respectively. Such result can testify the superiority of adaBoost algorithm. Then, to compare the data in Table 4 with Table 5, the new model proposed by this article has the best prediction accuracy than the other three models, and which is also shown in Fig. 10. Besides, three contrast models cost 15.018s, 4.080s and 12.897s respectively. The new model needs the time which approximates to the ada BP model. Although the new model needs a litter more time than PNN and SVM, it can sharply enhance prediction accuracy.

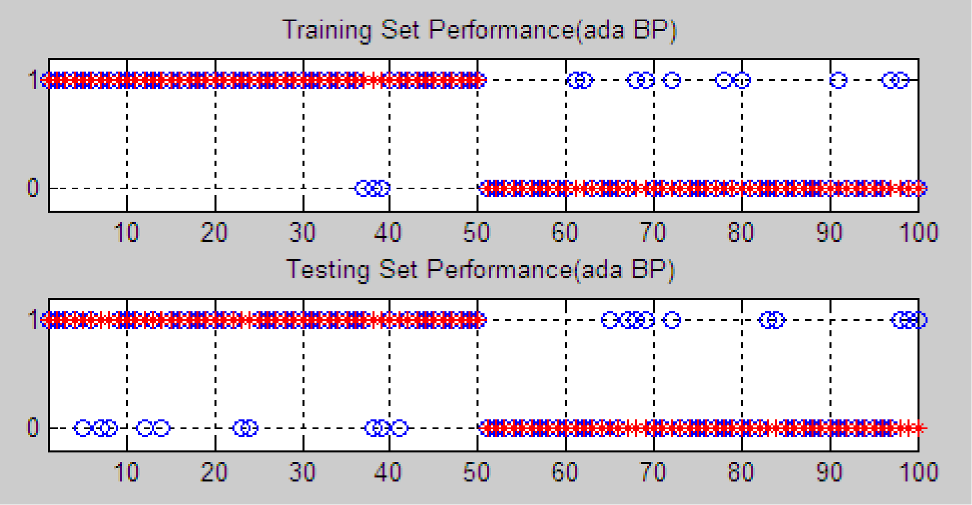

Fig. 7 Performance of Ada BP Model

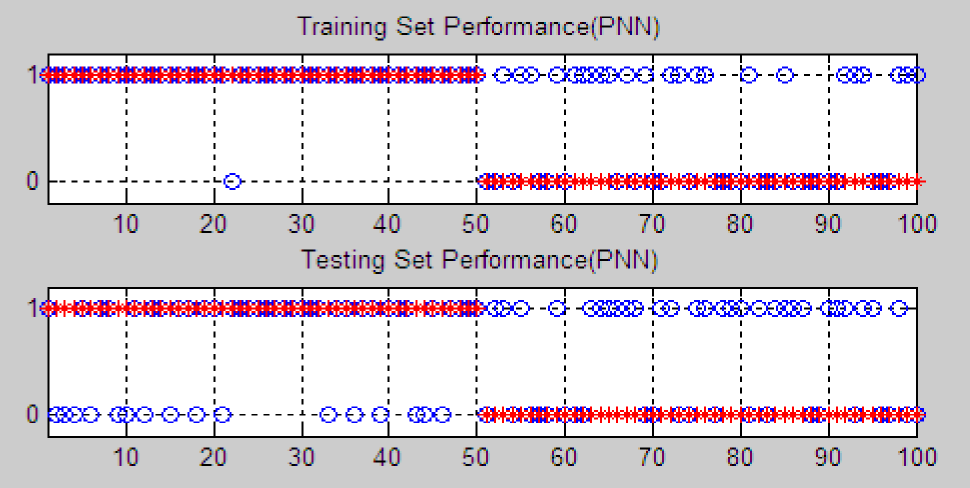

Fig. 8 Performance of PNN Model

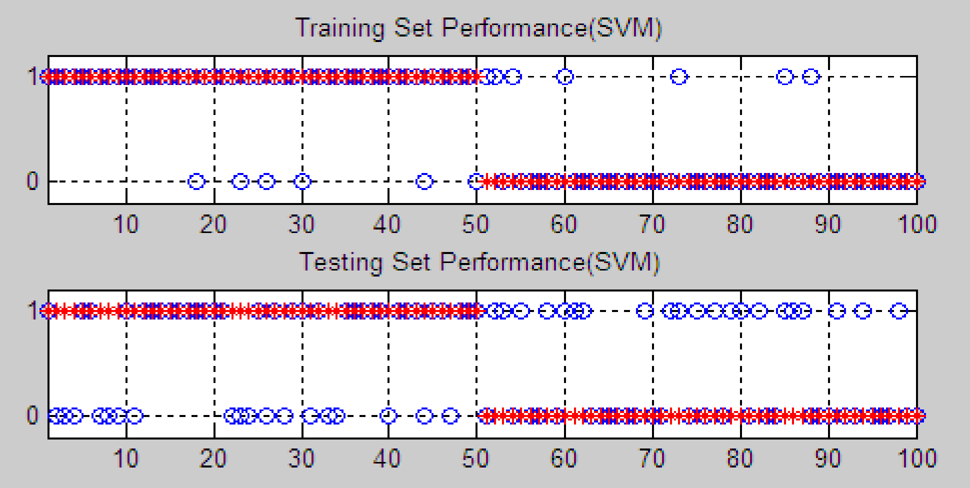

Fig. 9 Performance of SVM Model

Table 5 Prediction Performance of Contrast Models

| Model | Description | Training Set | | Testing Set | | Averaged Time |
|---|---|---|---|---|---|---|
| | | Quantity | Accurate number | Quantity | Accurate number | |
| Ada BP | Normal Companies | 50 | 47 | 50 | 40 | 15.018s |
| | ST Companies | 50 | 40 | 50 | 40 | |
| | Accuracy | 87% | | 80% | | |
| PNN | Normal Companies | 50 | 49 | 50 | 34 | 4.080s |
| | ST Companies | 50 | 27 | 50 | 22 | |
| | Accuracy | 76% | | 56% | | |
| SVM | Normal Companies | 50 | 44 | 50 | 32 | 12.897s |
| | ST Companies | 50 | 43 | 50 | 39 | |
| | Accuracy | 87% | | 71% | | |

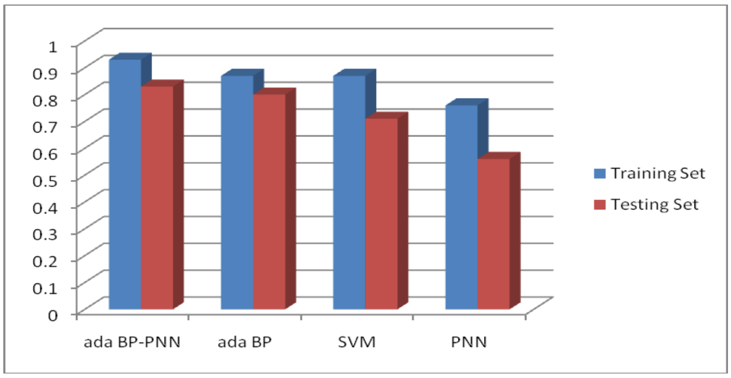

Fig. 10 Prediction Accuracies of Models

First, it compares prediction results of ada BP-PNN and ada BP models with the SVM and PNN. The models of adaboost algorithm, whether from the training and testing set, have better performance than the latter two models, which can demonstrate that adaboost algorithm is good at FCP. Then, it compares classification performance of ada BP-PNN with ada BP. As shown in Fig. 10, the accuracies of training and testing set from the proposed model are better than the result of the latter model. It is the reason that the new model uses the theory of back-propagation to optimize parameters of PNN. And as a weak classifier, the improved BP-PNN model is integrated by adaboost algorithm. The stronger classifier has better prediction accuracy. Therefore, it can demonstrate that ada BP-PNN model is one better classifier.

## 5. Conclusion

This paper proposes one new combination of prediction models for financial crisis. This model combines BP algorithm with PNN model on the basis of adaBoost algorithm. In the first section, this article employs BP algorithm to improve the structure of PNN model. It is creatively mentioned that the training set is divided into study and training samples during the training process. The study samples are used to create the BP-PNN model, and the training samples are used to modify weights and smoothing parameters for the BP-PNN model. In the next section, the BP-PNN models that are trained by randomly ranking samples are considered as weak classifiers. According to the AdaBoost algorithm, the stronger classifier is constructed by these weak classifiers. Whether companies are specially treated is the criterion that crisis companies and normal firms are distinguished. The specially treated companies are regarded as financial crisis enterprises, otherwise they are normal enterprises. To compare with other models, this new model can obtain ideal prediction outcomes.

The main contributions in this paper have three aspects. First, the BP algorithm is used to improve the computation rule of PNN model, which can modify simplex learning-method for the PNN model and enhance the prediction performance. Second, since the advantages of the adaBoost algorithm, the BP-PNN on the basis of adaBoost algorithm can efficiently enhance the prediction accuracy. Third, the key point is to enhance the models' prediction accuracy for FCP. The comparative experiment among different models has shown that this new ensemble model can more successfully predict the result whether the financial crisis will occur. Therefore, the ensemble model presented by this paper is more suitable to FCP.

## Acknowledgement

The research in this paper has been supported by the National Natural Science Foundation of China under Grant No. 71271070.

## References

1. Beaver, W.H.: Financial Ratios as Predictors of Failure, Empirical Research in Accounting: Selected Studies. *Journal of Accounting Research*, Vol. 4, 1966, pp. 79-111.
2. Altman, E.I.: Financial ratios discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, 23, 1968, pp. 589–609.
3. Balcaen S. and Ooghe H.: 35 Years of studies on business failure: an overview of the classical statistical methodologies and their related problems. *British Account.* Rev, Vol. 38, 2006, pp. 63–93.

4. Ohlson J.A.: Financial ratios and probabilistic prediction of bankruptcy. *Journal of Accounting Research*, Vol. 18, 1980, pp. 109–131.
5. Zavgren C.V.: Assessing the vulnerability to failure of American industrial firm: a logistic analysis. *Journal of Business Finance and Accounting*, Vol. 12, 1985, pp. 19–45.
6. Odom M.D. and Sharda R.: A neural network model for bankruptcy prediction. *IJCNN International Joint Conference on. IEEE*, Vol. 2, 1990, pp. 163–168.
7. Becerra V.M., Galvao, R.K.H. and Abou-Seada M.: Neural and wavelet network models for financial distress classification. *Data Mining and Knowledge Discovery*, Vol. 11, 2005, pp. 35–55.
8. Li F., Wu C. and Wu K.: An Improved Back Propagation Neural Network Model and Its Application. *Journal of Computers*, Vol. 9(8) , 2014, pp. 1858-1862.
9. Li H., Li C.J. and Wu X.J.: Statistics-based wrapper for feature selection: An implementation on financial distress identification with support vector machine. *Applied Soft Computin*, Vol. 19, 2014, pp. 57-67.
10. Li S., Wang M. and He J.: Prediction of Banking Systemic Risk Based on Support Vector Machine. *Mathematical Problems in Engineering*, Vol. 2013(2013), 2013, pp. 133-174.
11. Hsu M.F. and Pai P.F.: Incorporating support vector machines with multiple criteria decision making for financial crisis analysis. *Quality & Quantity*, Vol. 47(6), 2013, pp. 3481-3492.
12. Wu C., Wang, L. and Shi J.: Financial Distress Prediction Based on Support Vector Machine with a Modified Kernel Function : Journal of Intelligent Systems. *Journal of Intelligent Systems*, 33(33), 2015, pp. 177-187.
13. Chaudhuri A. and De K.:. Fuzzy support vector machine for bankruptcy prediction. *Applied Soft Computing*, Vol. 11(2), 2011, pp. 2472-2486.
14. Trabelsi S., Elouedi Z. and Lingras P.: Classification systems based on rough sets under the belief function framework. *International Journal of Approximate Reasoning*, Vol. 52(10), 2011, pp. 1409-1432.
15. Acosta-González E. and Fernández-Rodríguez F.: Forecasting Financial Failure of Firms via Genetic Algorithms. *Computational Economics*, Vol. 43(2), 2014, pp. 133-157.
16. Li Y.l., Wu C., Liu J.M. and Luo P.: A Combination Prediction Model of Stock Composite Index Based on Artificial Intelligent Methods and Multi-Agent Simulation. *International Journal of Computational Intelligence Systems*, Vol. 7(5), 2014, pp. 853-864.
17. Sun J. and Li H.: Financial distress prediction based on serial combination of multiple classifiers. *Expert Systems with Applications*, Vol. 36, 2009, pp. 8659–8666.
18. Nanni L., and Lumini A.: An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, Vol. 36(2), 2009, pp. 3028–3033.
19. Tsai C.F., Hsu Y.F., and Yen D.C.: A comparative study of classifier ensembles for bankruptcy prediction. *Applied Soft Computing*, Vol. 24, 2014, pp. 977-984.
20. Specht D.: Probabilistic neural networks. Neural Networks. Vol. 3, 1990, pp. 109–118.
21. Wang Q., Yang Q. and Zhang M.: Analyzing financial distress of listed companies using neural network. *System Science, Engineering Design and Manufacturing Informatization (ICSEM)*, 2012 3rd International Conference on. IEEE, Vol. 2, 2012, pp. 64-67.
22. Ticknor J.L.: A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*, Vol. 40(14), 2013, pp. 5501-5506.
23. Khashei M., Bijari M. and Raissi Ardali G.A.: Hybridization of autoregressive integrated moving average (ARIMA) with probabilistic neural networks (PNNs). *Computers & Industrial Engineering*, Vol. 63(1), 2012, pp. 37-45.
24. Zhou L.G.: Predicting the Removal of Special Treatment or Delisting Risk Warning for Listed Company in China with Adaboost. *Procedia Computer Science*, Vol. 17, 2013, pp. 633-640.
25. Heo J. and Yang J.Y.: AdaBoost based bankruptcy forecasting of Korean construction companies. *Applied Soft Computing*, Vol. 24, 2014, pp. 494-499.
26. Nie Q., Jin L. and Fei S.: Probability estimation for multi-class classification using AdaBoost. *Pattern Recognition*, Vol. 47(12), 2014, pp. 3931-3940.
27. Zhou L.G.: Predicting listing status of listed companies in China using adaboost approach. *Natural Computation (ICNC), 2014 10th International Conference on. IEEE*, 2014, pp. 731-735.
28. Yang Y. and Luo Y.: Using the back propagation neural network approach to bias correct TMPA data in the arid region of Northwest China. *Journal of Hydrometeorology*, Vol. 15(1), 2014, pp. 459-473.