Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 34 (2008) 1434–1444
Expert Systems with Applications
www.elsevier.com/locate/eswa

Credit risk assessment with a multistage neural network ensemble learning approach

Lean Yu [a,b,*], Shouyang Wang [a], Kin Keung Lai [b]

[a] Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100080, China
[b] Department of Management Sciences, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong

Abstract

In this study, a multistage neural network ensemble learning model is proposed to evaluate credit risk at the measurement level. The proposed model consists of six stages. In the first stage, a bagging sampling approach is used to generate different training data subsets especially for data shortage. In the second stage, the different neural network models are created with different training subsets obtained from the previous stage. In the third stage, the generated neural network models are trained with different training datasets and accordingly the classification score and reliability value of neural classifier can be obtained. In the fourth stage, a decorrelation maximization algorithm is used to select the appropriate ensemble members. In the fifth stage, the reliability values of the selected neural network models (i.e., ensemble members) are scaled into a unit interval by logistic transformation. In the final stage, the selected neural network ensemble members are fused to obtain final classification result by means of reliability measurement. For illustration, two publicly available credit datasets are used to verify the effectiveness of the proposed multistage neural network ensemble model.

Keywords: Credit risk assessment; Neural network; Ensemble learning; Bagging; Reliability

1. Introduction

Without doubt credit risk assessment and modeling is one of the most important topics in the field of financial risk management (Lai, Yu, Wang, & Zhou, 2006a, 2006b). Due to recent financial crises and regulatory concern of Basel II, credit risk assessment has been the major focus of financial and banking industry. Especially for any credit-granting institution, such as commercial banks and certain retailers, the ability to discriminate good customers from bad ones is crucial. The need for reliable models that predict defaults accurately is imperative so that the interested parties can take either preventive or corrective action (Lai et al., 2006a, Lai, Yu, Wang, & Zhou, 2006b, 2006c; Wang, Wang, & Lai, 2005). Therefore, credit risk evaluation becomes very important for sustainability and profit of enterprises. Furthermore, an accurate prediction of credit risk could be transformed into a more efficient use of economic capital in business.

Usually, the generic approach of credit risk assessment is to apply some classification techniques on similar data of previous customers – both faithful and delinquent customers – in order to find a relation between the characteristics and potential failure. One important ingredient needed to accomplish this goal is to seek an accurate classifier in order to categorize new applicants or existing customers as good or bad. Due to its importance of credit risk assessment, there is a growing research stream about credit risk evaluation. First of all, many statistical models and optimization techniques, such as linear discriminant analysis (Fisher, 1936) logit analysis (Wiginton, 1980), probit analysis (Grablowsky & Talley, 1981), linear programming (Glover, 1990), integer programming (Mangasarian, 1965), *k*-nearest neighbor (KNN) (Henley & Hand, 1996)

* Corresponding author. Address: Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 55 Zhongguancun East Road, Haidian District, Beijing, PR China. Tel.: +86 10 62565817; fax: +86 10 62568364.
E-mail address: yulean@amss.ac.cn (L. Yu).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.01.009

and classification tree (Makowski, 1985) are widely applied to credit risk assessment and modeling tasks. Although these methods can be used to assess credit risk, the ability to discriminate good customers from bad ones is still worth improving further. Recent studies have revealed that emerging artificial intelligent (AI) techniques, such as artificial neural networks (ANN) (Lai et al., 2006b; Malhotra & Malhotra, 2003; Smalz & Conrad, 1994), genetic algorithm (GA) (Chen & Huang, 2003; Varetto, 1998) and support vector machine (SVM) (Huang, Chen, Hsu, Chen, & Wu, 2004; Van Gestel, Baesens, Garcia, & Van Dijcke, 2003) are advantageous to statistical models and optimization techniques for credit risk evaluation.

Although almost all classification methods can be used to assess credit risk, some combined classifiers, which integrate two or more single classification methods, have shown higher correctness of predictability than any individual methods. Combined classifier research is currently flourishing in credit risk assessment. Recent examples are Neural Discriminant Technique (Lee, Chiu, Lu, & Chen, 2002), neuro-fuzzy (Malhotra & Malhotra, 2002; Piramuthu, 1999) and fuzzy SVM (Wang et al., 2005). Some comprehensive literature review about credit risk assessment and modeling can be referred to two recent surveys (Thomas, 2002; Thomas, Oliver, & Hand, 2005) for more details.

Motivated by the combined or hybrid classifiers, integrating multiple classifiers into an aggregated output, i.e., ensemble technique has been turned out to be an efficient strategy for achieving high classification performance, especially in fields where the development of a powerful single classifier system requires considerable efforts. In this study, ANN is selected as the generic instrument to construct an ensemble classifier. The main reason of selecting ANN reflects the following two aspects. First of all, a neural network is often viewed as a "universal approximator" (Hornik, Stinchocombe, & White, 1989). Usually, a three-layer back propagation neural network (BPNN) with an identity transfer function in the output unit and logistic functions in the middle-layer units can approximate any continuous function arbitrarily well given a sufficient amount of middle-layer units (Hornik et al., 1989; White, 1990). That is, neural networks have the ability to provide flexible mapping between inputs and outputs. Secondly, neural networks are far from being optimal classifier (Yang & Browne, 2004). Many experimental results have shown the generalization of individual neural networks is not unique. Even for some simple problems, different neural networks with different settings (e.g., different network architecture and different initial conditions) may result in different generalization results. This characteristic makes neural networks have large improvement space in performance.

To achieve high classification performance, there are some essential requirements to the ensemble members and the ensemble strategy. First of all, a basic condition is that the individual neural network classifiers must have enough training data. Secondly, the ensemble members are diverse or complementary, i.e., classifiers must show different classification properties. Thirdly, a wise ensemble strategy is also required on a set of complementary classifiers in order to obtain high classification performance.

For the first requirement, some sampling approaches, such as bagging (Breiman, 1996), have been used for creating different training samples by varying the data subsets selected or perturbing training sets (Yang & Browne, 2004). In credit risk assessment, the available data samples are often limited (Lai et al., 2006c). Due to the features of its random sampling with replacement, the bagging approach can rightly remedy the shortcoming.

For the second requirement, diverse ensemble members can be obtained by varying the initial conditions or using different training data. Because neural network is an unstable learner, it is sensitive to the initial conditions and different train data. Furthermore, the architecture of the neural network itself is determined by trial and error. Thus, constructing different neural ensemble members is rather easy.

In the ensemble model, the most important point is to select an appropriate ensemble strategy, which mentioned in the third requirement. Generally, the variety of ensemble methods can be grouped into three categories according to the level of classifier outputs: abstract level (crisp class), rank level (rank order) and measurement level (class score) (Suen & Lam, 2000; Xu, Krzyzak, & Suen, 1992). In the existing studies, many ensemble systems still use empirical heuristics and ad hoc ensemble schemes at the abstract level. Typically, majority voting (Xu et al., 1992; Yang & Browne, 2004) uses the abstract level of output of ensemble members. An important drawback of this ensemble strategy is that it does not take confidence degree of neural network output into account. Actually, ensemble at the measurement level is advantageous in that the output measurements contain richer information of class measures. In a sense, an appropriate ensemble strategy is more crucial, especially for integrating the classifiers that output diverse measurements. Furthermore, the intensive investigation of neural network ensemble for credit risk evaluation has not formulated a convincing theoretical foundation and overall process model yet.

In such situations, we propose a novel multistage reliability-based neural network ensemble learning approach that differs in that the final ensemble strategy is determined the reliability of neural network output at the measurement level. In this study, the proposed neural network ensemble learning model consists of six stages. In the first stage, a bagging sampling approach is used to generate different training data subsets especially for data shortage. In the second stage, the different neural network models are created with different training subsets obtained from the previous stage. In the third stage, the generated neural network models are trained with different training datasets and accordingly the classification score and reliability value of neural classifier can be obtained. In the fourth stage, a decorrelation maximization algorithm is used to select the appropriate ensemble members. In the fifth stage, the

reliability values of the selected neural network models (i.e., ensemble members) are scaled into a unit interval by logistic transformation. In the final stage, the selected neural network ensemble members are fused to obtain final classification result by means of reliability measurement. For testing and illustration purposes, two publicly available credit datasets are used to verify the effectiveness of the proposed neural network ensemble model.

The motivation of this study is to formulate a multistage reliability-based neural network ensemble learning model for credit risk evaluation and compare its performance with other existing credit risk assessment techniques. The rest of the study is organized as follows. The next section presents a formulation process of the multistage neural network ensemble learning model in detail. To verify the effectiveness of the proposed method, two real examples are performed and accordingly the experiment results are reported in Section 3. And Section 4 concludes the study.

2. The formulation process of neural network ensemble model

In this section, a six-stage reliability-based neural network ensemble learning model is proposed for classification purpose. The basic idea of neural network ensemble originated from using all the valuable information hidden in neural network classifiers, where each can contribute to the improvement of generalization. In our proposed multistage neural network ensemble model, a bagging sampling approach is first used to generate different training sets for guaranteeing enough training data. In terms of different training datasets, multiple individual neural classifiers are trained. Accordingly some classification results and reliability values of each neural classifier are also obtained. Then a decorrelation maximization algorithm is used to select the appropriate ensemble members from the multiple trained neural classifiers. Subsequently the reliability values are transformed into a unit interval for avoiding the situation that member classifier with large absolute value often dominates the final decisions of the ensemble. Finally the ensemble members are aggregated in terms of some criteria, and their generated results are output based upon reliability measure. The final result is called the ensemble output. The general architecture of the multistage reliability-based neural network ensemble learning model is illustrated in Fig. 1.

2.1. Partitioning original data set

Due to data shortage in some data analysis problems, some approaches, such as bagging (Breiman, 1996) have been used for creating samples by varying the data subsets selected or perturbing training sets (Yang & Browne, 2004). Bagging (Breiman, 1996) is a widely used data sampling method in the machine learning field. Given that the size of the original data set DS is P, the size of new training data is N, and the number of new training data items is m, the bagging sampling algorithm can be shown in Fig. 2.

The bagging algorithm (Breiman, 1996) is very efficient in constructing a reasonable size of training set due to the feature of its random sampling with replacement. Therefore, bagging is a useful data preparation method for machine learning. In this study, we use the bagging algorithm to generate different training data subsets when the original data is scarcity.

2.2. Creating diverse neural network classifiers

According to the definition of effective ensemble classifiers by Hansen and Salamon (1990), 'a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is if the classifiers are accurate and diverse'. Generally, an effective ensemble classifier consisting of diverse models with much disagreement is more likely to have a good generalization performance in terms of the principle of bias-variance trade-off (Yu, Lai, Wang, & Huang, 2006). Therefore, how to generate the diverse model is a crucial factor. For neural network model, several methods have been investigated for the generation of ensemble members making different errors (Sharkey, 1996). Such methods basically rely on varying the parameters related to the design and to the training of neural networks. In particular, the main methods include the following four aspects:

(1) Different initial conditions: diverse ensemble members can be created by varying the initial conditions, such as initial random weights, learning rate and momentum rate, from which each network is trained.
(2) Different network architecture: by changing the number of hidden layers and the number of nodes in every layer, different neural networks with different architectures can be created.
(3) Different training data: by re-sampling and preprocessing data, we can obtain different training sets, thus making different network generations. There are six techniques that can be used to obtain diverse training data sets (Yang & Browne, 2004): bagging (Breiman, 1996), noise injection (Raviv & Intrator, 1996), cross-validation (Krogh & Vedelsby, 1995), stacking (Wolpert, 1992), boosting (Schapire, 1990) and input decimation (Tumer & Ghosh, 1996).
(4) Different training algorithm: diverse ensemble member can also be generated by selecting different core learning algorithms. For example, a multilayer feedforward network can use the steep-descent algorithm (Hornik et al., 1989; White, 1990), Levenberg–Marquardt algorithm (Tumer & Ghosh, 1996) and other training algorithms.

In our study, the third way is selected because the previous phase has created many different training data subsets. In addition, the three-layer back-propagation neural

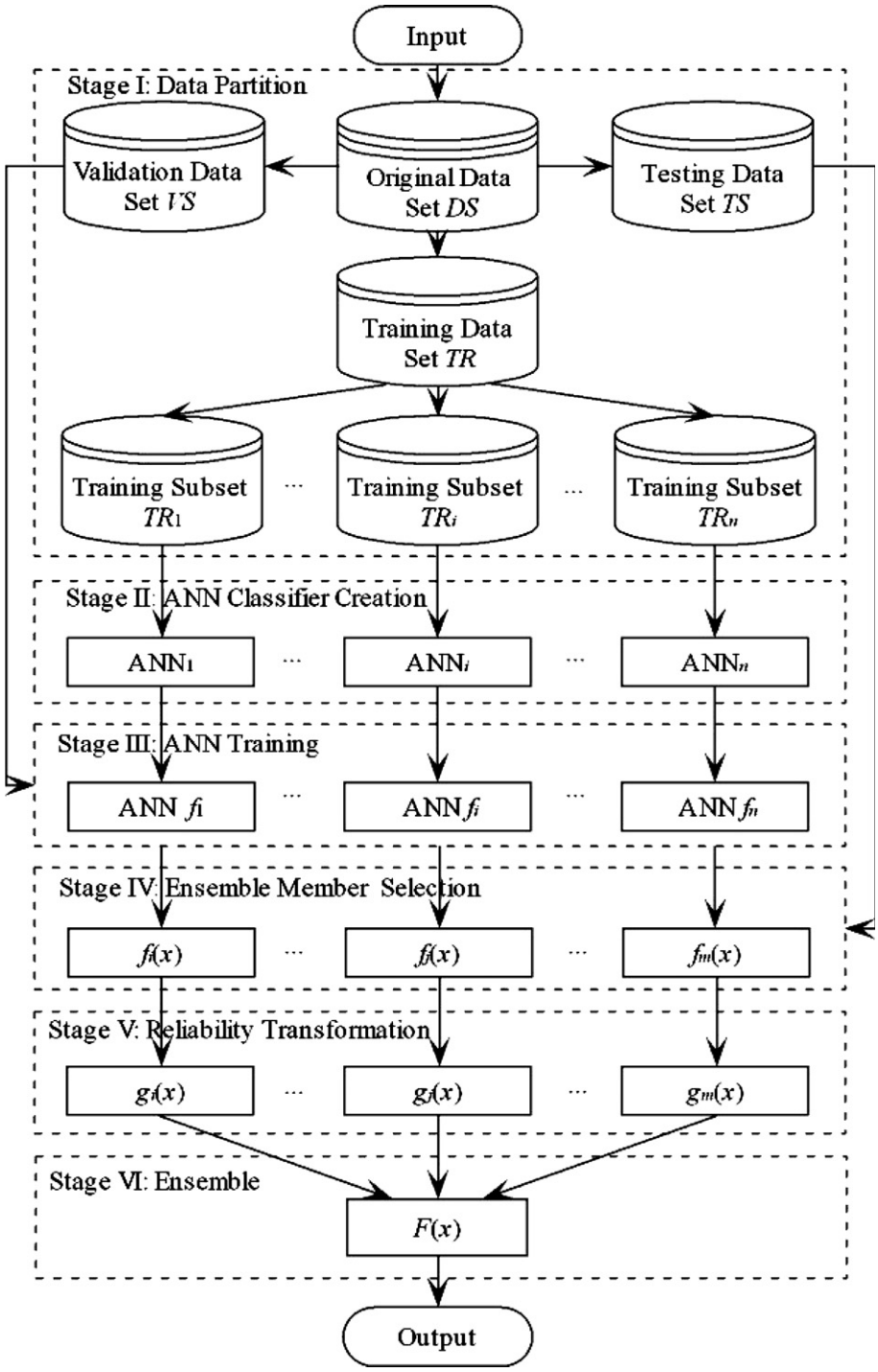

Fig. 1. The general formulation process of multistage neural network ensemble learning model.

networks (BPNN) (Hornik et al., 1989; White, 1990) are selected because a three-layer BPNN with an identity transfer function in the output unit and logistic transfer functions in the middle-layer units can approximate any continuous function arbitrarily well given a sufficient amount of middle-layer units (Hornik et al., 1989; White, 1990). With these different training datasets, diverse neural network classifiers will be generated.

2.3. Neural network learning and confidence value generation

After creating diverse neural network classifiers, the next step is to train the neural network with different training datasets. In our study, the selected BPNN is a class of supervised error back-propagation learning mechanism in the form of the neural network associative memory. Usually, the back-propagation learning mechanism consists

Input: original data set DS
Output: The generated new training subsets (TR_1, TR_2, …, TR_m)
For $t = 1$ **to** m
  For $i = 1$ **to** N
    $RandRow = P * rand()$
    If $RandRow <= P$
      $P_t(i, AllColumns) = DS(RandRow, AllColumns)$
    End If
  Next i
Next t
Output the final training subsets (TR_1, TR_2, …, TR_m)

Fig. 2. The bagging algorithm.

of two phases: forward-propagation and back-propagation phase. Suppose we have s samples. Each is described by $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ and $T_i = (t_{i1}, t_{i2}, \ldots, t_{in})$ where X_i is an input vector, T_i is a target output vector and $1 \leqslant i \leqslant s$.

In the first phase (forward-propagation phase), X_i is fed into the input layer, and an output $\boldsymbol{Y}_i = (y_{i1}, y_{i2}, \ldots, y_{in})$ is generated based on the current weight vector W. The objective is to minimize an error function E defined as

$$E = \sum_{i=1}^{s} \sum_{j=1}^{n} \frac{(y_{ij} - t_{ij})^2}{2}, \tag{1}$$

by changing W so that all input vectors are correctly mapped to their corresponding output vectors.

In the second phase (back-propagation phase), a gradient descent in the weight space, W, is performed to locate the optimal solution. The direction and magnitude change Δw_{ij} can be computed as

$$\Delta w_{ij} = -\frac{\partial E}{\partial w_{ij}} \varepsilon, \tag{2}$$

where $0 < \varepsilon < 1$ is a learning parameter controlling the algorithm's convergence rate.

The total squared error calculated by Eq. (1) is propagated back, layer by layer, from the output units to the input units in the second phase. Weight adjustments are determined on the way of propagation at each level. The two phases are executed during each iteration of the back-propagation algorithm until E converges.

For classification task, a neural network can usually be trained by the in-sample dataset and applied to out-of-sample dataset for verification. The model parameters (connection weights and node biases) will be adjusted iteratively by a process of minimizing the error function E. Basically, the final output of the FNN model can be represented as

$$y = f(x) = a_0 + \sum_{j=1}^{q} w_j \varphi\left(a_j + \sum_{i=1}^{p} w_{ij} x_i\right), \tag{3}$$

where $a_j (j = 0, 1, 2, \ldots, q)$ is a bias on the jth unit, and $w_{ij} (i = 1, 2, \ldots, p;\ j = 1, 2, \ldots, q)$ is the connection weight between layers of the model, $\phi(\cdot)$ is the transfer function of the hidden layer, p is the number of input nodes and q is the number of hidden nodes.

By training neural network, model parameters in Eq. (3) can be determined and accordingly the neural network classifier can be shown as

$$F(x) = \operatorname{sign}\left(a_0 + \sum_{j=1}^{q} w_j \varphi\left(a_j + \sum_{i=1}^{p} w_{ij} x_i\right)\right). \tag{4}$$

In our study, we mainly use neural network output value $f(x)$ as its classification score at the measurement level, instead of the classification results $F(x)$ directly. For credit risk classification problem, a credit analyst can adjust the parameter a_0 to modify the cutoff to change the percent of accepted. Only when the applicant's credit score is larger than the cutoff, his application will be accepted.

In addition, the neural network output value $f(x)$ is a good indicator for the reliability degree of ensemble classifiers. The larger the $f(x)$, the higher the neural network classifier for positive class is. Therefore the neural network output value $f(x)$ as a reliability measure is used to integrate the ensemble members. By means of this treatment, we can realize the decision fusion at the measurement level.

2.4. Selecting appropriate ensemble members

After training, each individual neural classifier has generated its own result. However, if there are a great number of individual members, we need to select a subset of representatives in order to improve ensemble efficiency. Furthermore, in the neural network ensemble model, it does not follow the rule of "the more, the better", as mentioned by Yu, Wang, and Lai (2005). In addition, this is the necessary requirement of diverse neural network classifier for ensemble learning. In this study, a decorrelation maximization method (Lai et al., 2006b) is used to select the appropriate number of neural network ensemble members.

As earlier noted, the basic starting point of the decorrelation maximization algorithm is the principle of ensemble model diversity. That is, the correlations between the selected classifiers should be as small as possible, i.e., decorrelation maximization. Supposed that there are p neural classifiers $(f_1, f_2, \ldots, f_p)$ with n forecast values. Then the error matrix $(e_1, e_2, \ldots, e_p)$ of p predictors can be represented by

$$E = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1p} \\ e_{21} & e_{22} & \cdots & e_{2p} \\ \vdots & \vdots & & \vdots \\ e_{n1} & e_{n2} & \cdots & e_{np} \end{bmatrix}_{n \times p}. \tag{5}$$

From the matrix, the mean, variance and covariance of E can be calculated as

$$\text{Mean}: \bar{e}_i = \frac{1}{n} \sum_{k=1}^{n} e_{ki} \quad (i = 1, 2, \ldots, p). \tag{6}$$

$$\text{Variance}: V_{ii} = \frac{1}{n}\sum_{k=1}^{n}(e_{ki} - \bar{e}_i)^2 \quad (i = 1, 2, \ldots, p). \tag{7}$$

$$\text{Covariance}: V_{ij} = \frac{1}{n}\sum_{k=1}^{n}(e_{ki} - \bar{e}_i)(e_{kj} - \bar{e}_j) \quad (i, j = 1, 2, \ldots, p). \tag{8}$$

Considering Eqs. (7) and (8), we can obtain a variance–covariance matrix:

$$V_{p\times p} = (V_{ij}). \tag{9}$$

Based upon the variance–covariance matrix, correlation matrix R can be calculated using the following equations:

$$R = (r_{ij}), \tag{10}$$

$$r_{ij} = \frac{V_{ij}}{\sqrt{V_{ii}V_{jj}}}, \tag{11}$$

where r_{ij} is correlation coefficient, representing the degree of correlation classifier f_i and classifier f_j.

Subsequently, the plural-correlation coefficient $\rho_{fi|(f1,f2,\ldots,fi-1,fi+1,\ldots,fp)}$ between classifier f_i and other $p-1$ classifiers can be computed based on the results of Eqs. (10) and (11). For convenience, $\rho_{fi|(f1,f2,\ldots,fi-1,fi+1,\ldots,fp)}$ is abbreviated as ρ_i, representing the degree of correlation between f_i and $(f_1, f_2, \ldots, f_{i-1}, f_{i+1}, \ldots, f_p)$. In order to calculate the plural-correlation coefficient, the correlation matrix R can be represented with block matrix, i.e.,

$$R \xrightarrow{\text{after transformation}} \begin{bmatrix} R_{-i} & r_i \\ r_i^T & 1 \end{bmatrix}, \tag{12}$$

where R_{-i} denotes the deleted correlation matrix. It should be noted that $r_{ii} = 1 (i = 1, 2, \ldots, p)$. Then the plural-correlation coefficient can be calculated by

$$\rho_i^2 = r_i^{\mathrm{T}} R_{-i}^{\mathrm{T}} r_i \quad (i = 1, 2, \ldots, p). \tag{13}$$

For a pre-specified threshold θ, if $\rho_i^2 > \theta$, then the classifier f_i should be taken out from the p classifiers. On the contrary, the classifier f_i should be retained. Generally, the decorrelation maximization algorithm can be summarized into the following steps:

(1) Computing the variance–covariance matrix V_{ij} and correlation matrix R with Eqs. (9) and (10).
(2) For the ith classifier $(i = 1, 2, \ldots, p)$, the plural-correlation coefficient ρ_i can be calculated with Eq. (13).
(3) For a pre-specified threshold θ, if $\rho_i < \theta$, then the ith classifier should be deleted from the p classifiers. Conversely, if $\rho_i > \theta$, then the ith classifier should be retained.
(4) For the retained classifiers, we can also perform the procedure Eqs. (1)–(3) iteratively until satisfactory results are obtained.

2.5. Reliability value transformation

In the previous phase, the neural classifier outputs are used as reliability measure. It is worth noting that the reliability value falls into the interval $(-\infty, +\infty)$. The main drawback of this confidence value is that ensemble classifier with large absolute value often dominate the final decision of the ensemble model.

In order to overcome this weakness, one simple strategy is to re-scale the output values into zero mean and unit standard deviation, i.e.,

$$g_i^+(x) = \frac{f_i(x) - \mu}{\sigma}, \tag{14}$$

where μ and σ are the mean and standard deviation of the pooled classifier outputs, respectively. However, for classifiers that output dissimilarity measures, the sign of the original outputs should be reversed before this transformation.

For convenience, transforming the confidence value into the unit interval $[0, 1]$ is a good solution (Lai et al., 2006b). In neural network, the logistic function behaves well in squashing neural output to approximate probability measures. Therefore we can take it as a scaling function for reliability transformation, i.e.,

$$g_i^+(x) = \frac{1}{1 + \mathrm{e}^{-f_i(x)}}. \tag{15}$$

In a binary classification problem, if the reliability degree for positive class is $g_i^+(x)$, the reliability degree for negative class can be represented as

$$g_i^-(x) = 1 - g_i^+(x). \tag{16}$$

According to the transformed reliability values, multiple classifiers can also be fused into an ensemble output, as illustrated in the following section.

2.6. Integrating multiple classifiers into an ensemble output

Depended upon the work done in previous several stages, a set of appropriate number of ensemble members can be collected. The subsequent task is to combine these selected members into an aggregated classifier in an appropriate ensemble strategy. Generally, there are some ensemble strategy in the literature at the abstract level and the rank level. Typically, majority voting, ranking and weighted averaging are three popular ensemble approaches. Majority voting is the most widely used ensemble strategy for classification problems due to its easy implementation. Ensemble members' voting determines the final decision. Usually, it takes over half the ensemble to agree a result for it to be accepted as the final output of the ensemble regardless of the diversity and accuracy of each network's generalization. Majority voting ignores the fact some neural network that lie in a minority sometimes do produce the correct results. At the ensemble stage, it ignores the existence of diversity that is the motivation for ensembles (Yang & Browne, 2004). In addition, majority voting is only a class of ensemble strategy at the abstract level.

Ranking is where the members of an ensemble are called low level classifiers and they produce not only a single result but a list of choices ranked in terms of their likelihood. Then

the high level classifier chooses from this set of classes using additional information that is not usually available to or well represented in a single low level classifier (Yang & Browne, 2004). However, ranking strategy is a class of fusion strategy at the rank level, as earlier mentioned.

Weighted averaging is where the final ensemble decision is calculated in terms of individual ensemble members' performances and a weight attached to each member's output. The gross weight is one and each ensemble member is entitled to a portion of this gross weight based on their performances or diversity (Yang & Browne, 2004). Although this approach is a class of ensemble strategy at the measurement level, but it is difficult for classification problem to obtain the appropriate weights for each ensemble member.

In such situations, this study proposes a reliability-based ensemble strategy to make the final decision of the ensemble at the measurement level. The following five strategies can be used to integrate the individual ensemble members (Lai et al., 2006b):

(1) Maximum strategy:

$$F(x)=\begin{cases}1, & \text{if } \max\limits_{i=1,\ldots,m} g_i^+(x) \geqslant \max\limits_{i=1,\ldots,m} g_i^-(x),\\ -1, & \text{otherwise.}\end{cases} \tag{17}$$

(2) Minimum strategy:

$$F(x)=\begin{cases}1, & \text{if } \min\limits_{i=1,\ldots,m} g_i^+(x) \geqslant \min\limits_{i=1,\ldots,m} g_i^-(x),\\ -1, & \text{otherwise.}\end{cases} \tag{18}$$

(3) Median strategy:

$$F(x)=\begin{cases}1, & \text{if } \text{median}_{i=1,\ldots,m}(g_i^+(x)) \geqslant \text{median}_{i=1,\ldots,m}(g_i^-(x)),\\ -1, & \text{otherwise.}\end{cases} \tag{19}$$

(4) Mean strategy:

$$F(x)=\begin{cases}1, & \text{if } \sum\limits_{i=1}^{m} g_i^+(x) \geqslant \sum\limits_{i=1}^{m} g_i^-(x),\\ -1, & \text{otherwise.}\end{cases} \tag{20}$$

(5) Product strategy:

$$F(x)=\begin{cases}1, & \text{if } \prod\limits_{i=1,\ldots,m} g_i^+(x) \geqslant \prod\limits_{i=1,\ldots,m} g_i^-(x),\\ -1, & \text{otherwise.}\end{cases} \tag{21}$$

To summarize, the multistage reliability-based neural network ensemble learning model can be concluded in the following steps:

(1) Partitioning original dataset into n training datasets, $\text{TR}_1, \text{TR}_2, \ldots, \text{TR}_n$.
(2) Training n individual neural network models with the different training dataset $\text{TR}_1, \text{TR}_2, \ldots, \text{TR}_n$ and obtaining n individual neural network classifiers, i.e., ensemble members.
(3) Selecting m decorrelated neural network classifiers from n neural classifiers using decorrelation maximization algorithm.
(4) Using Eq. (3) to obtain the m neural classifiers' output values of new unlabeled sample $x, f_1(x), f_2(x), \ldots, f_m(x)$, as in Fig. 1 illustrated.
(5) Using Eqs. (15) and (16) to transform output value to reliability degrees for positive class $g_1^+(x), \ldots, g_m^+(x)$ and for negative class $g_1^-(x), \ldots, g_m^-(x)$.
(6) Fusing the multiple neural classifiers into an aggregated output in terms of reliability value using Eqs. (17)–(21).

3. Experiments

In this section, two published credit datasets from real world are used to test the performance of the proposed approach. For comparison purposes, three individual classification models: logit regression (LogR) (Wiginton, 1980), artificial neural network (ANN) (Henley & Hand, 1996; Makowski, 1985) and support vector machine (SVM) (Huang et al., 2004; Lai et al., 2006a; Van Gestel et al., 2003), two hybrid classification models: neuro-fuzzy system (Malhotra & Malhotra, 2002; Piramuthu, 1999) and fuzzy SVM (Wang et al., 2005) are also conducted the experiments. In addition, the classification accuracy in testing set is used as performance evaluation criterion. Typically, three evaluation criteria are used to measure the classification results.

$$\text{Type I accuracy}=\frac{\text{number of both observed bad and classified as bad}}{\text{number of observed bad}}. \tag{22}$$

$$\text{Type II accuracy}=\frac{\text{number of both observed good and classified as good}}{\text{number of observed good}}. \tag{23}$$

$$\text{Total accuracy}=\frac{\text{number of correct classification}}{\text{the number of evaluation sample}}. \tag{24}$$

3.1. Consumer credit risk assessment

This experimental dataset in this subsection is about Japanese consumer credit card application approval obtained from UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/databases/credit-screening). For confidentiality all attribute names and values have been changed to meaningless symbols. After deleting the data with missing attribute values, we obtain 653 data, with 357 cases were granted credit and 296 cases were refused. To delete the burden of resolving multicategory, we use the 13 attributes A1–A5, A8–A15. Because we generally should substitute k-class attribute with $k-1$ binary attribute, which will greatly increase the dimensions of input space, we do not use two attributes: A6 and A7.

In this empirical analysis, we randomly draw 400 data from the 653 data as the initial training set, 100 data as

the validation set and the else as the testing set. For single ANN model, a three-layer back-propagation neural network with 25 TANSIG neurons in the hidden layer and one PURELIN neuron in the output layer is used. That is, the neural network with architecture of 13-25-1 is used. Besides, the learning rate and momentum rate is set to 0.1 and 0.15. The accepted average squared error is 0.05 and the training epochs are 2000. In the single SVM model, the kernel function is Gaussian function with regularization parameter $C = 50$ and $\sigma^2 = 5$. Similarly, the above parameters are obtained by trial and error. For the proposed neural ensemble model, 40 different neural network models with different initial weights are generated to increase model accuracy for credit risk evaluation. Using the decorrelation maximization algorithm, 18 diverse neural network classifiers are selected. The basic setting of the ensemble members is similar to the single ANN model, as previously mentioned. For two hybrid model, we use the reported results in the literature.

To reflect model robustness, each class of experiment is repeated 10 times and the final Type I, Type II and total accuracy is the average of the results of the 10 individual tests. According to the experiment design, the final results are presented in Table 1. Note that the results of two hybrid classification models are from the original literature (Piramuthu, 1999; Wang et al., 2005). Because the results of type I and type II in (Piramuthu, 1999) are not reported, the result of neuro-fuzzy system is kept to be blank in Table 1. Based on the similar reason, the standard deviations of the fuzzy SVM model are not shown in Table 1.

As can be seen in Table 1, we can find the following several conclusions.

(1) Of the three single models, neural network model performs the best, followed by single SVM and logit regression. Using two tailed *t*-test, we find that the difference between performance of ANN and SVM is insignificant at five percent level of significance, while the difference between logit regression and ANN is significant at ten percent level of significance.
(2) In the two listed hybrid models, the neuro-fuzzy system performs worse than that of two single AI models, i.e., ANN and SVM. The main reason reflects the following aspects. First of all, the neuro-fuzzy system used the approximations of both the inputs as well as the output, as it fuzzified the inputs and defuzzified the output. Comparatively, the ANN and SVM did not use any such approximations. Secondly, the classification accuracies using neuro-fuzzy system is also influenced by the overlap in the way the range of values of a given attribute is split into its various categories (e.g., range of values for small, medium, and large). Again, these are pitfalls associated with the mechanisms used for both fuzzification and defuzzification of input and output data, respectively (Piramuthu, 1999). However, the fuzzy SVM obtain good performance relative to single classification models. The main reason is that the fuzzy SVM can reduce the effect of outliers and yield higher classification rate than single SVM and ANN do.
(3) In the ensemble model, five reliability-based neural network ensemble models consistently outperform the majority voting based ensemble model, implying that the proposed reliability-based neural network ensemble model is a class of promising approach to handle credit risk analysis. Among the five reliability-based neural network ensemble models, the neural network ensemble model with minimum ensemble rule perform the best, followed by maximization ensemble rule and mean ensemble rule. Although there is no significant difference in performance of the five reliability-based neural network ensemble models, the main reason resulting in such a small difference is still unknown, which is worth further exploring in the future.

3.2. Corporation credit risk assessment

In this subsection, the used dataset are about UK corporation credit from the Financial Analysis Made Easy (FAME) CD-ROM database which can be found in the Appendix of (Beynon & Peel, 2001). It contains the detailed information of 60 corporations, in which including 30

Table 1
Consumer credit risk evaluation results with different methods[a]

Category	Model	Rule	Type I (%)	Type II (%)	Total (%)
Single	Log R		74.58 [6.47]	76.36 [5.81]	75.82 [6.14]
	ANN		80.08 [7.23]	82.26 [6.25]	80.77 [6.86]
	SVM		78.41 [5.71]	81.43 [6.13]	79.91 [5.87]
Hybrid	Neuro-fuzzy (Piramuthu, 1999)				77.91 [5.10]
	Fuzzy SVM (Wang et al., 2005)		82.70	85.43	83.94 [4.75]
Ensemble	Voting-based	Majority	84.37 [5.73]	86.58 [6.11]	85.22 [6.01]
	Reliability-based	Maximum	88.43 [4.34]	86.54 [5.25]	87.24 [4.89]
		Minimum	88.86 [4.41]	87.44 [4.74]	88.08 [4.63]
		Median	86.52 [4.96]	85.63 [5.03]	86.03 [4.99]
		Mean	86.17 [5.28]	87.85 [5.43]	86.89 [5.35]
		Product	85.75 [5.11]	86.46 [6.08]	85.96 [5.73]

[a] Standard deviations appear in brackets.

failed and 30 non-failed firms. Twelve variables are used as the firms' characteristics:

(01) Sales;
(02) ROCE: profit before tax/capital employed;
(03) FFTL: funds flow (earnings before tax and depreciation)/total liabilities;
(04) GEAR: (current liabilities + long-term debt)/total assets;
(05) CLTA: current liabilities/total assets;
(06) CACL: current assets/current liabilities;
(07) QACL: (current assets − stock)/current liabilities;
(08) WCTA: (current assets − current liabilities)/total assets;
(09) LAG: number of days between account year end and the date the annual report and accounts were failed at company registry;
(10) AGE: number of years the company has been operating since incorporation date;
(11) CHAUD: coded 1 if changed auditor in previous three years, 0 otherwise;
(12) BIG6: coded 1 if company auditor is a Big6 auditor, 0 otherwise.

In our experiments, all samples are randomly splitted into three parts: 30 training dataset, 10 validation dataset and 20 testing dataset. Fifty different training subset are randomly generated by bagging algorithm due to the scarcity of data samples. In addition, we make the number of good firms equal to the number of bad firms in both the training and testing samples, so as to avoid the embarrassing situations that just two or three good (or bad, equally likely) inputs in the test sample. Thus the training sample includes 15 data of each class. This way of composing the sample of firms was also used by several researchers in the past, e.g., (Dimitras, Slowinski, Sunsmaga, & Zopounidis, 1999; Zavgren, 1985), among others. Its aim is to minimize the effect of such factors as industry or size that in some cases can be very important. Except from the above training sample, the validation sample and testing sample are also collected using a similar approach. The validation dataset is composed of 5 failed and 5 non-failed firms and the testing set consists of 10 failed and 10 non-failed firms.

In the ANN model, a three-layer BPNN with the architecture of 12-21-1 is used. That is, it has 12 input neurons, 21 TANSIG neurons in the hidden layer and one PURELIN neuron in the output layer. The network training function is the TRAINLM. Besides, the learning rate and momentum rate is set to 0.15 and 0.35. The accepted average squared error is 0.05 and the training epochs are 2000. The above parameters are obtained by trial and error. In the single SVM model, the kernel function is Gaussian function with regularization parameter $C = 10$ and $\sigma^2 = 1$. Similarly, the above parameters are obtained by trial and error. For the proposed neural ensemble model, 50 different neural network models with different initial weights are generated to increase model accuracy for credit risk evaluation. Using the decorrelation maximization algorithm, 22 diverse neural network classifiers are selected. The basic setting of the ensemble members is similar to the single ANN model, as previously mentioned. For two hybrid model, the results of the neuro-fuzzy system are not reported at all and this model is excluded here. That is, we only use one hybrid model – fuzzy SVM (Wang et al., 2005) and shown their experiment results reported in the literature.

To reflect model robustness, each class of experiment is repeated 20 times and the final Type I accuracy, Type II accuracy and total accuracy are the average of the results of the 20 individual tests. According to the previous experiment design, the final computational results are shown in Table 2.

From Table 2, several important conclusions can be found in the following:

(1) For three evaluation criteria, the proposed reliability-based ensemble learning model performs the best, followed by the majority voting ensemble model, fuzzy SVM, SVM and ANN, the logit regression is the worst, indicating that the ensemble and hybrid models can consistently outperform other individual classification models in credit risk assessment and meantime implying the strong capability of the multistage reliability-based neural network ensemble learning

Table 2
Corporation credit risk evaluation results with different methods[a]

Category	Model	Rule	Type I (%)	Type II (%)	Total (%)
Single	Log R		70.51 [5.47]	71.36 [6.44]	70.77 [5.96]
	ANN		72.14 [7.85]	74.07 [7.03]	73.63 [7.29]
	SVM		76.54 [6.22]	78.85 [5.51]	77.84 [5.82]
Hybrid	Fuzzy SVM [3]		79.00	79.00	79.00 [5.65]
Ensemble	Voting-based	Majority	80.15 [7.57]	82.06 [7.18]	81.63 [7.33]
	Reliability-based	Maximum	83.48 [5.64]	85.42 [5.75]	84.14 [5.69]
		Minimum	82.89 [6.28]	86.36 [5.51]	85.01 [5.73]
		Median	82.17 [5.89]	85.63 [5.37]	84.25 [5.86]
		Mean	83.05 [6.33]	86.24 [6.23]	85.09 [5.68]
		Product	84.34 [6.06]	87.23 [7.15]	85.87 [6.59]

[a] Standard deviations appear in brackets.

model in credit risk classification. The main reason is that integrating multiple diverse models can remedy the shortcomings of any individual methods thus increasing the classification accuracy.

(2) For three single models, the accuracy of the SVM model is better than that of the other two single models. This conclusion is different from that of the previous experiment. The possible reason has two aspects. First of all, different methods may have different classification capability for different datasets. Second, different datasets may have different classification property. The two possible reasons lead to this interesting result.

(3) For the reliability-based neural ensemble learning model, the performance of the product strategy is the best of the five ensemble strategy, followed by mean strategy. The main reason leading to this conclusion is unknown and is worth exploring further. However, through two-tail paired *t*-test, the average performance difference of the five ensemble strategies is insignificant at 10% significant level. This finding is consistent with the results of previous experiment. It is also obvious that the performances of the five ensemble strategies are quite close.

From two experiments in this study, the proposed multistage reliability-based neural network ensemble learning model generally performs the best in terms of Type I accuracy, Type II accuracy, and total accuracy, revealing that the proposed reliability-based neural network ensemble learning technique is a feasible solution to improve the accuracy of credit risk evaluation.

4. Conclusions

Due to the huge outstanding amount and increasing speed of bankruptcy filings, credit risk assessment has attracted much research interests from both academic and industrial communities. A more accurate, consistent, and robust credit evaluation technique can significantly reduce future costs for the credit industry.

In this study, a multistage neural network ensemble learning model is proposed for credit risk assessment. Different from commonly used "one-member-one-vote" or "majority-rule" ensemble, the novel neural network ensemble aggregates the decision values from the different neural ensemble members, instead of their classification results directly. The new ensemble strategy consists of two critical steps: scaling, which transforms decision values to degrees of reliability, and fusion, which aggregates degrees of reliability to generate final classification results.

For verification two publicly available credit dataset have been used to test the effectiveness and classification power of the proposed neural network ensemble learning approach. All results reported in the experiment clearly show that the proposed neural network ensemble model can consistently outperform the other comparable models including three single models, two hybrid models and majority-voting-based ensemble model. These results obtained reveal that the proposed neural network ensemble learning model can provide a promising solution to credit risk analysis and meantime implying that the proposed multistage neural network ensemble learning technique has a great potential to other binary-class classification problems.

Acknowledgements

The authors would like to thank the Editor-in-Chief and reviewers for their recommendation and comments. This work is partially supported by the grants from the National Natural Science Foundation of China (NSFC Nos. 70221001 and 70601029), the Chinese Academy of Sciences (CAS No. 3547600), the Academy of Mathematics and Systems Science (AMSS No. 3543500) of CAS and Strategic Research Grant of City University of Hong Kong (SRG Nos. 7001677 and 7001806).

References

Beynon, M. J., & Peel, M. J. (2001). Variable precision rough set theory and data discretisation: an application to corporate failure prediction. *Omega, 29*, 561–576.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 26*, 123–140.

Chen, M. C., & Huang, S. H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications, 24*, 433–441.

Dimitras, A. I., Slowinski, R., Sunsmaga, R., & Zopounidis, C. (1999). Business failure prediction using rough sets. *European Journal of Operational Research, 114*, 263–280.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*, 179–188.

Glover, F. (1990). Improved linear programming models for discriminant analysis. *Decision Science, 21*, 771–785.

Grablowsky, B. J., & Talley, W. K. (1981). Probit and discriminant functions for classifying credit applicants: A comparison. *Journal of Economic Business, 33*, 254–261.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*, 993–1001.

Henley, W. E., & Hand, D. J. (1996). A k-NN classifier for assessing consumer credit risk. *Statistician, 45*, 77–95.

Hornik, K., Stinchocombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359–366.

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

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles cross validation and active learning. In G. Tesauro, D. Touretzky, & T. Leen (Eds.), *Advances in neural information processing systems* (pp. 231–238). Cambridge, MA: MIT Press.

Lai, K. K., Yu, L., Wang, S. Y., & Zhou, L. G. (2006a). Credit risk evaluation with least square support vector machine. *Lecture Notes in Computer Science, 4062*, 490–495.

Lai, K. K., Yu, L., Wang, S. Y., & Zhou, L. G. (2006b). Credit risk analysis using a reliability-based neural network ensemble model. *Lecture Notes in Computer Science, 4132*, 682–690.

Lai, K. K., Yu, L., Wang, S. Y., & Zhou, L. G. (2006c). Neural network metalearning for credit scoring. *Lecture Notes in Computer Science, 4113*, 403–408.

Lee, T. S., Chiu, C. C., Lu, C. J., & Chen, I. F. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert Systems with Application, 23*(3), 245–254.

Makowski, P. (1985). Credit scoring branches out. *Credit World, 75*, 30–37.

Malhotra, R., & Malhotra, D. K. (2002). Differentiating between good credits and bad credits using neuro-fuzzy systems. *European Journal of Operational Research, 136*, 190–211.

Malhotra, R., & Malhotra, D. K. (2003). Evaluating consumer loans using neural networks. *Omega, 31*, 83–96.

Mangasarian, O. L. (1965). Linear and nonlinear separation of patterns by linear programming. *Operations Research, 13*, 444–452.

Piramuthu, S. (1999). Financial credit-risk evaluation with neural and neurofuzzy systems. *European Journal of Operational Research, 112*, 310–321.

Raviv, Y., & Intrator, N. (1996). Bootstrapping with noise: an effective regularization technique. *Connection Science, 8*, 355–372.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*, 197–227.

Sharkey, A. J. C. (1996). On combining artificial neural nets. *Connection Science, 8*, 299–314.

Smalz, R., & Conrad, M. (1994). Combining evolution with credit apportionment: a new learning algorithm for neural nets. *Neural Networks, 7*, 341–351.

Suen, C. Y., & Lam, L. (2000). Multiple classifier combination methodologies for different output levels. *Lecture Notes in Computer Science, 1857*, 52–66.

Thomas, L. C. (2002). A survey of credit and behavioral scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting, 16*, 149–172.

Thomas, L. C., Oliver, R. W., & Hand, D. J. (2005). A survey of the issues in consumer credit modelling research. *Journal of the Operational Research Society, 56*, 1006–1015.

Tumer, K., & Ghosh, J. (1996). Error correlations and error reduction in ensemble classifiers. *Connection Science, 8*, 385–404.

Van Gestel, T., Baesens, B., Garcia, J., & Van Dijcke, P. (2003). A support vector machine approach to credit scoring. *Bank en Financiewezen, 2*, 73–82.

Varetto, F. (1998). Genetic algorithms applications in the analysis of insolvency risk. *Journal of Banking and Finance, 22*, 1421–1439.

Wang, Y. Q., Wang, S. Y., & Lai, K. K. (2005). A new fuzzy support vector machine to evaluate credit risk. *IEEE Transactions on Fuzzy Systems, 13*, 820–831.

White, H. (1990). Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings. *Neural Networks, 3*, 535–549.

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behaviour. *Journal of Financial Quantitative Analysis, 15*, 757–770.

Wolpert, D. (1992). Stacked generalization. *Neural Networks, 5*, 241–259.

Xu, L., Krzyzak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics, 22*, 418–435.

Yang, S., & Browne, A. (2004). Neural network ensembles: combining multiple models for enhanced performance using a multistage approach. *Expert Systems, 21*, 279–288.

Yu, L., Lai, K. K., Wang, S. Y., & Huang, W. (2006). A bias-variance-complexity trade-off framework for complex system modeling. *Lecture Notes in Computer Science, 3980*, 518–527.

Yu, L., Wang, S. Y., & Lai, K. K. (2005). A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. *Computers and Operations Research, 32*, 2523–2541.

Zavgren, C. V. (1985). Assessing the vulnerability to failure of American industrial firms: a logistic analysis. *Journal of Business Finance and Accounting, 12*, 19–45.