Expert Systems With Applications 165 (2021) 113925

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Utilizing historical data for corporate credit rating assessment

Mingfu Wang, Hyejin Ku [*]

Department of Mathematics and Statistics, York University, 4700 Keele St, Toronto, ON M3J 1P3, Canada

ARTICLE INFO

Keywords:
Credit rating
Ensemble learning
Artificial Neural Networks
Artificial Intelligence

ABSTRACT

Corporate credit rating assessment is one of the crucial problems of credit risk management; it will help the financial institutions and government decide whether to issue debts. Recent studies focusing on the prediction of credit rating by using artificial intelligence (AI) techniques have shown impressive results compared to traditional statistical methods. Although the AI techniques can be used to assess credit risk, the prediction accuracy is still worth improving further, as even a small improvement in credit rating prediction accuracy leads to significant loss reduction in the industry. In this paper, we propose new learning analytic methods to enhance the prediction accuracy of credit rating. First, we devise the metrics based on the credit rating history of the firms, and expand the feature space with new input variables. This approach can be applied to any conventional AI methods for improvement of prediction accuracy. Second, we develop a novel learning algorithm that is designed to take into account historical financial data. We propose the parallel artificial neural networks (PANNs) ensemble model that creates several independent artificial neural networks (ANNs); each ANN deals with financial performance of the firms for each year, and the final output of PANNs is aggregated by ensemble learning. In our experiment, three different real-world datasets are used to validate the performance of our proposed approach. Consequently, the experimental results show that our proposed approach achieved competitive results compared to conventional AI techniques.

1. Introduction

Credit risk assessment is one of the most critical management problems in the field of financial risk, especially after the global financial crisis, the banks forced on corporate credit rating to reduce the default risk and systematic risk (Baesens, Setiono, Mues, & Vanthienen, 2003; Kim & Ahn, 2012; Tsai & Chen, 2010; Yu, Wang, & Lai, 2008). Credit rating is an independent evaluation process whose aim is to find out how a debtor is capable and willing to meet its payable obligations, specifically based on the complex analysis of all the known risk factors of the assessed object (Hájek, 2011). Credit rating is not only used for bank's financial instrument, but also a risk management tool, which is used by bond investors, debts issuers, and even government officers. Debt issuers use credit rating as a measure of the company risk, it presents the level of the credit risk of debtors, predicting their ability to pay back the debts, allowing firms issuing debt to estimate the likely return investors require. Moreover, bankers and regulators rely on credit rating to make a decision; many regulatory requirements for the financial decisions are based on the current credit rating. In general, credit rating is assigned to market participants by credit rating agencies; in the marketplace, the largest of credit agencies are Standard & Poor's (S&P), Moody's, and Fitch Ratings. However, these agencies charge large fees for their services, because they invest large amounts of time and human resources in performing the credit rating process. Also, these agencies periodically provide ratings that do not reflect the real situation due to time lag. Therefore, there is a much larger effort the researchers made to simulate the credit rating process. The prediction accuracy of credit rating has a significant impact on financial institutions' profitability and government regulations. Even a 1% improvement on the prediction accuracy of the corporate credit rating will decrease a significant loss and risk for both financial institutions and government (Tsai & Chen, 2010). Thus, developing an appropriate model for credit rating is the most important and difficult task for managers and researchers in both industry and academia.

Many statistical methods have been used for prediction of credit ratings, such as logistic regression (Laitinen, 1999; Reichert, Cho, & Wagner, 1983; Thomas, 2000), linear discriminant analysis (LDA) (Thomas, 2000), and multivariate adaptive regression splines (MARS) (Friedman et al., 1991). Although these methods have wide applications in the financial area, the statistical models have difficulty in modeling the complex financial systems due to the limitations of the model and the statistical assumptions. By contrast, the artificial intelligence (AI) techniques are affected by these restrictions to a much lesser degree. Recently, AI techniques have been widely applied in financial areas,

[*] Corresponding author.
E-mail addresses: morgan08@yorku.ca (M. Wang), hku@mathstat.yorku.ca (H. Ku).

https://doi.org/10.1016/j.eswa.2020.113925
Received 28 January 2020; Received in revised form 25 August 2020; Accepted 26 August 2020
Available online 2 September 2020

especially in credit rating assessment. AI methods can automatically extract useful information from the dataset to develop different prediction models. The main difference between statistical methods and AI techniques is that statistical methods usually need the researchers to propose a model to fit the dataset; thus, the limitations of the statistical methods are the assumption about the distribution of the data and linearity of the classification model definitions. Many empirical studies have demonstrated that the performance of AI techniques is superior to traditional statistical methods, especially for non-linear credit rating classification. Among these studies, the most popular methods are Support vector machines (SVM) (Baesens, Van Gestel et al., 2003; Cao, Guan, & Jingqing, 2006; Kim & Ahn, 2012; Krebel, 1999; Wang, Wang, & Lai, 2005) and artificial neural networks (ANN) (Desai, Crook, & Overstreet, 1996; Lai, Yu, Wang, & Zhou, 2006; Thulasiram, Rahman, & Thulasiraman, 2003). Also, many scholars put much effort on the hybrid models and ensemble methods, which have more prediction power compared to conventional AI methods. Donate, Cortez, Sánchez, and De Miguel (2013) and Yu et al. (2008) proposed several novel neural network ensemble models and verified that they have better forecasting performance. Due to the success of ensemble ANN model, it has been widely applied in many areas. However, training the ensemble ANN has a huge time consumption due to large training subsets. In order to accelerate the training process, Zhang, Xu, Zhang, and Root (2016) proposed a powerful and general parallel artificial neural network to accelerate the training process. Large numbers of training samples are distributed to multiple cores on a cluster system to achieve a high speed-up for training.

Motivated by Zhang et al. (2016), we propose a novel parallel ANNs model, which takes into account the past financial performance to improve the accuracy of credit rating prediction. Similar to multi-agent ANNs (Yang & Browne, 2004) and the parallel ANN training technique (Zhang et al., 2016), our proposed model creates several independent traditional ANNs. However, the main difference is that our proposed model takes into consideration historical financial data, e.g., each ANN deals with each year's financial performance, and the more powerful classifier is aggregated by extracting useful information from historical data. Overall, the parallel ANNs model consists of n ANNs, and each ANNs' outputs are combined for the final output by the weighted average algorithm. On the other hand, we adopt a new approach to build metrics and characterize past movements in credit rating, inspired by Kim (2003), Masoud (2014) and Saia, Carta, and Fenu (2018). We consider the historical data of credit rating as a time series of financial feature, and construct the metrics based on credit rating history. We propose five metrics, which include Momentum, Disparity, Disparity ratio, First-order variation, and Second-order variation. Momentum measures the change in credit rating over a year; Disparity measures the distance between its current rating and moving average over the past $T (T \geq 1)$ years; Disparity ratio represents the ratio of the difference between its current rating and moving average to the difference between the highest credit rating and the lowest credit rating over the past T years; First-order variation measures the cumulative change in credit rating and its volatility (or mobility) for the past T years; Second-order variation measures the volatility of credit rating and focuses more on big movements.

The main goal of this paper is to utilize historical financial data and credit rating history for improving prediction accuracy. In pursuing this goal, we make the following contributions. First, a novel ensemble method (called PANNs) is proposed based on the parallel computing technique that takes into account historical financial data. Our approach is validated by two different real-world cases from the U.S and Japan, and compared to established classifiers. The four different comprehensive performance metrics are employed to assess predictive performance for different perspectives such as the precision, recall, F_1 score, and AUC. Second, new metrics are built to extract the characteristics from credit rating history, and applied to conventional AI methods, e.g., SVM, Stacking, Random Forest, ECOC, and ANN. In order to validate the effectiveness of our approach, three different real-world datasets from the U.S. and one dataset from Japan are adopted in our experiment. Via the experimental study, it is shown that utilizing historical financial data and credit rating history is an effective way to enhance the prediction accuracy for the credit rating classification problem, and our approaches yield promising results in credit rating assessment.

The rest of this paper is structured as follows. Section 2 reviews the related work in the area of credit rating assessment. Section 3 briefly introduces the conventional learning algorithms, which include multi-class SVM (One-vs-One, One-against-All), Random Forest, ECOC, OMSVM (Forward, Backward), Stacking, and ANN. Section 4 presents the data preprocessing and feature selection, then reveals the experimental results of conventional methods. Section 5 proposes new metrics based on credit rating history and a novel structure of ensemble learning ANNs, named parallel ANNs (PANNs). Section 6 analyzes the obtained experimental results. Section 7 provides the conclusion.

2. Related work

A large number of credit rating classification methods have been proposed in the literature. Support Vector Machines (SVM) is one of the most powerful methods for the multi-label credit rating classification problem. SVM is originally devised for binary classification, but it is not naturally geared for multi-class classification. Cao et al. (2006) extended the standard SVM to multi-class SVM by constructing several binary classifiers, and applied to the multi-class classification of credit rating. Kim and Ahn (2012) devised a novel ordinal pairwise partitioning multi-class SVM (OMSVM), which extended the binary SVM by using the ordinal pairwise partitioning strategy, and it can efficiently and effectively deal with multiple ordinal classes through constructing fewer classifiers. The benefit of multi-class SVM is that the solution of SVM may be globally optimal because the goal of SVM seeks to minimize structural risk. More recently, Maldonado, Pérez, and Bravo (2017) proposed two novel cost-based Mixed-integer programming approaches for SVM used in credit rating classification, which demonstrated the effectiveness of these methods in terms of predictive performance at a low cost in the real-world data from Chilean banks.

ANN is another popular machine learning technique that has been more frequently adopted in the application of corporate credit rating assessment. Baesens, Setiono et al. (2003) and West (2000) investigated the performance of the ANN model in credit rating, and showed the ANN model performs better than the traditional statistical methods. ANN benefits from strong learning ability and facilitates risk modeling without assumptions about the relationship between the variables (Li & Zhong, 2012). However, ANN is challenging to explain how input variables in network prediction relate to each other, and also ANN has much computational complexity and running time requirements. Extreme learning machines (ELM) have developed to overcome this problem. Bequé and Lessmann (2017) examined the performance of ELM for credit scoring in three different aspects, (i) ease of use, (ii) computational complexity, (iii) discriminative accuracy, and also assessed the ELM in conjunction with different ensemble frameworks.

Recent works showed that hybrid models and ensemble methods can achieve a better result for credit rating assessment. Among them, Yu et al. (2008) developed a multistage neural network ensemble model to evaluate credit rating. The new selective ensemble strategy consists of two critical steps, the first step is scaling, which transforms decision values to degrees of reliability, the second step is fusion, which aggregate degrees of reliability to generate final classification results. Donate et al. (2013) proposed an evolutionary artificial neural network (EANN) to evolve a fitness weighted n-fold cross-validation ANN ensemble scheme for time series forecasting, which demonstrated the EANN model has the capability of forecasting for the future based on time-series data. More recently, Abellán and Castellano (2017) compared Credal Decision Tree (CDT) with several base classifiers applied in different ensemble schemes for credit rating tasks. He, Zhang, and Zhang (2018) extended the BalanceCascade approach to generate adjustable balanced subsets based on the imbalance ratios of training data for obtaining superior predictive performance. In addition, Chornous and Nikolskyi (2018) proposed an ensemble-based technique combining selected base classification models with business-specific feature selection add-on to increase the classification accuracy of the real-life case of credit scoring.

Despite the tremendous amount of studies and high quality results for credit rating assessment, there is very little literature studied on utilizing historical financial data and credit rating history for an improvement of prediction accuracy. Several studies emphasize the ensemble frameworks that combine multiple classifiers are superior to a single classifier in isolation. However, they have not considered the effects of the previous financial performance and credit rating history on the current credit rating. In this work, we develop an ensemble method to examine the effects of previous financial performance on the current credit score, which takes advantage of the parallel computing structure to take into account historical financial data. Moreover, the credit rating history contains vital information on the firms' financial position. Thus, constructing the credit rating metrics to measure the movement of credit ratings is well-suited to improve prediction accuracy for any AI methods.

3. Conventional learning algorithms

3.1. Multi-class SVM

In general, the conventional SVM is devised for binary classification. It is not naturally geared for multi-class classifications of credit ratings. However, there exists a variety of techniques to extend the conventional SVM to multi-class SVM. The main idea of these techniques is to decompose the multi-class problem into several binary-class problems, and then combines several binary classifiers. In this subsection, we will present three main methods, which include multi-class SVM (One-vs-one, One-against-all), ECOC, and OMSVMs (Forward, Backward).

Constructing several binary classifiers: One-vs-one

The method of One-vs-one involves decomposition of the multi-class SVMs to binary classifiers for each two classes. Assume there are k classes, the One-vs-one model constructs $\frac{k(k-1)}{2}$ pairs of binary SVMs classifiers for all pairs of classes. For each pair of classes, a binary SVMs classifier is constructed by maximizing the margin between the two classes. The decision-function assigns an instance to a class that has the largest number of votes that so-called vote count strategy, which was introduced by Krebel (1999). For training data from the ith and the jth classes, we solve the following binary classification problem:

$$\begin{aligned}&\text{Min}\,\frac{1}{2}(\mathbf{w}^{ij})^{\mathbf{T}}\mathbf{w}^{ij} + C\sum \delta_t^{ij},\\ &\text{subject to}:\ \ y_i(\mathbf{w}^{ij})^{\mathbf{T}}K(\boldsymbol{x}_i) + b^{ij} + \delta_t^{ij} - 1 \geq 0,\ \delta_t^{ij} \geq 0,\end{aligned} \tag{1}$$

where $K(\cdot)$ is the kernel function that maps the attributes to a high dimensional space.

Constructing several binary classifiers: One-against-all

This method constructs k binary SVM classifiers for a k-class classification, e.g. class 1 vs all other classes; class 2 vs all other classes; ...; until class k vs all other classes. The method of multi-class One-against-all has been published by Statnikov, Aliferis, Tsamardinos, Hardin, and Levy (2004). The mth class SVM is trained with all of the examples in the mth class with positive labels and all other examples with negative labels (Statnikov et al., 2004). Thus, the mth SVMs solves the following problem:

$$\begin{aligned}&\text{Min}\,\frac{1}{2}(\mathbf{w}^{m})^{\mathbf{T}}\mathbf{w}^{m} + C\sum_{i=1}^{n} \delta_i^{m},\\ &\text{subject to}:\ y_i[(\mathbf{w}^{m})^{\mathbf{T}}K(\boldsymbol{x}_i) + b^{m}] + \delta_i^{m} - 1 \geq 0,\ \delta_i^{m} \geq 0,\end{aligned} \tag{2}$$

where $K(\cdot)$ is the kernel function that maps the attributes into a high-dimensional space. The combined One-against-all decision function chooses the class for a sample that corresponds to the maximum of k binary classification functions that are specified by the farthest positive hyperplane (Kim & Ahn, 2012).

Constructing several binary classifiers: Error-Correcting Output Codes (ECOC)

The ECOC approach, adopted from the digital communication theory, fuses the decisions that were generated by individual SVM classifier, which was introduced in Dietterich and Bakiri (1994). This method constructs a code matrix, where row i represents the code-vector of class i, and column j represents a classifier assignment. Then, to determine the class, ECOC compares the error-correcting codes with each row of the matrix. Assuming there are Q classes and S is the number of binary classifiers, during the process a new input x is classified by computing the vector formed by the outputs of the classifiers, $f(x) = (f_1(x), f_2(x), \ldots, f_S(x))$ and choosing the class whose corresponding row is closest to $f(x)$ (Klautau, Jevtić, & Orlitsky, 2003). Thus, the classification can be seen as a decoding operation and the class of input x is computed as:

$$\underset{q=\{1,\ldots,Q\}}{\arg\min}\ d(m_q, f(x)), \tag{3}$$

where m_q is the coding matrix of qth row, Q is the number of classes, d is generally the Hamming distance, which is computed as follows:

$$d(m_q, f) = \sum_{s=1}^{S} \frac{|m_{qs} - sign(f_s)|}{2}. \tag{4}$$

Klautau et al. (2003) showed an example of the error-correcting codes for four-class classification where a classifier (p vs q) responds with +1 when the output class is p and −1 when the output class is q.

Constructing several binary classifiers: OMSVM (Forward and Backward)

The methods of multi-class SVM are of interest to researchers, who investigate how to achieve effective and efficient classifiers for credit ratings. A new type of multi-class SVM technique has been developed by Kim and Ahn (2012), and it is called ordinal pairwise multi-class SVM (OMSVM). The OMSVM is a hybrid algorithm that applies the ordinal pairwise partitioning technique to multi-class SVM. The partitioning One-against-followers method is similar to One-against-all in multi-class SVM, but OMSVM builds fewer classifiers and works more efficient. The binary classifiers are constructed for the pairs $\{(i,j) : i = 1, 2, \ldots, k-1, j = \bigcup_{m=i+1}^{k} m\}$, where k is the total number of classes. Consequently, One-against-followers constructs $k-1$ binary classifiers if there are k classes. Regarding the methods of fusion, there are forward and backward methods. The forward method fuses the binary classifiers from the lowest level of classes to the highest level of classes. In contrast, the backward method combines the binary classifiers in reverse direction, i.e., it constructs the binary classifiers from the highest classes to the lowest classes. Also, Kim and Ahn (2012) graphically showed that the binary classifiers and order of their application for the four-class classification problem and the mechanism of each type of OMSVM.

3.2. Random forest

Random Forests (RF) is a bagging ensemble learning algorithm, which is widely used in different areas such as classification, regression, and feature selection. That is because there are many advantages, e.g., it has a higher prediction accuracy than other methods, and the randomness of RF can effectively avoid over-fitting. At the beginning of the RF method, the bootstrap algorithm is employed to generate different subsets from the original training set, and the base learners of each subset (i.e., Decision Trees) are trained on these subsets.

In general, RF is a dual diversity Decision Tree (DT) that combines bagging and random subspace feature selection to merge individual decision trees (Breiman, 2001). Randomness is explicitly introduced as the following two steps. First of all, T subsets are generated, where each subset is randomly selecting N (sample size) data from the original sample. In addition, each subset is independent of others. Secondly, an un-pruned tree is built from each subset using random subspace feature selection to generate splits, it reduces the correlation between trees in the forest. Each tree casts a unit vote for the most popular class at the point $t \in T$, with the final class is obtained by majority rule (Breiman, 2001). RF differs from simple tree bagging in the sense that the former selects a random subset of features at each candidate splitting point when growing DTs. After n DTs have been created, majority voting is performed to determine the label of prediction.

3.3. Artificial neural network (ANN)

Artificial neural network (ANN) is originally proposed to simulate the way of biological neural network. With the capability to learn complex relationships between inputs and outputs, ANN has been widely used in many applications. Many researchers have developed the artificial neural networks as useful high-performance analysis tools for credit rating (Abdou, Pointion, & EI-Masry, 2008; Baesens, Setiono et al., 2003; Hájek, 2011; West, 2000; Zhou & Da Xu, 2001). An ANN is composed of a group of neural nodes that link with weighted nodes. Every node can simulate a neuron of creatures, and the connection among the neurons (Hájek, 2011; Zhong, Miao, Shen, & Feng, 2014). The most general type of neural network consists of three layers of units: input layer, hidden layer, and output layer. A layer of input units is connected to a layer of hidden units, which is connected to a layer of output units. Multi-layer Perceptron (MLP) is a simple feed-forward neural network, which is frequently used with excellent approximation capabilities, and it performs a linear combination of input variables (Abdou et al., 2008). More formally, assuming that a simple MLP consists of one hidden layer and two output neurons, the hidden layer has N nodes and denote h_i as the output of hidden neuron i. The output of hidden layer is computed by processing the weighted inputs and its bias term $b_i^{(1)}$ as follows.

$$h_i = g^{(1)}(b_i^{(1)} + \sum_{j=1}^{N} w_{ij}^{(1)} x_j), \tag{5}$$

where $w_{ij}^{(1)}$ denotes the weight connecting input j to hidden unit i. Then, the final output of MLP O_i can be expressed in the following:

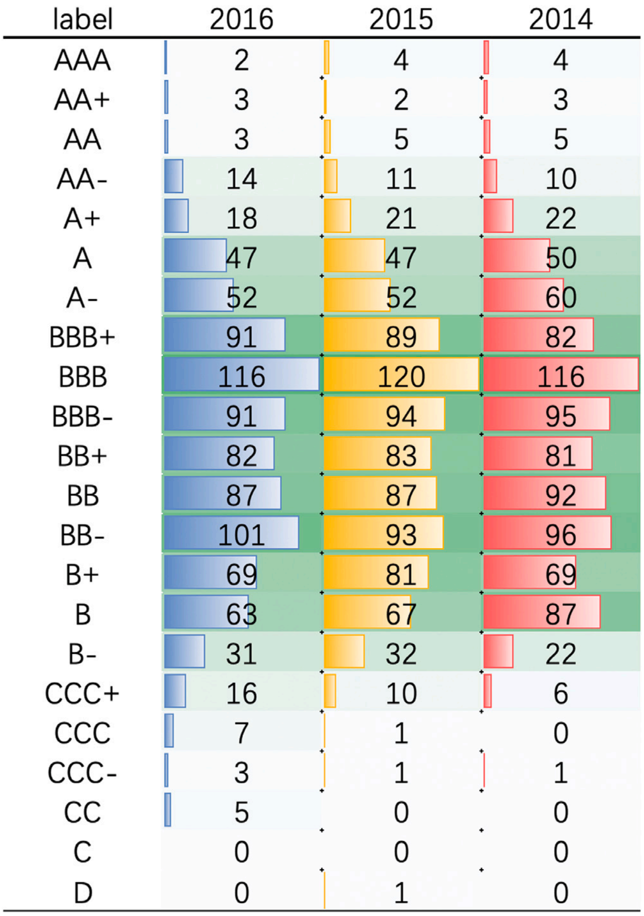
$$O_i = g^{(2)}(b_i^{(2)} + \sum_{j=1}^{N} w_{ij}^{(2)} h_j), \tag{6}$$

where $w_{ij}^{(2)}$ is the connection weight from the jth hidden node to the ith output node, and g is the activation function. The activation function allows the network to model nonlinear relationship in the data (Baesens, Setiono et al., 2003). There are different types of activation function for each neuron such as Tanh, ReLu, and Sigmoid. A MLP can apply different activation functions depending on the problems. The parameters of weights w_{ij} need to be updated during a training process, which is usually based on gradient descent or stochastic gradient descent learning algorithm to minimize some kind of loss function over a set of training observations. Starting from initial random weights, MLP minimizes the loss function by repeatedly updating these weights. If the loss function is the mean squared error and ϵ is the distance between the anticipated output and output of MLP, then ϵ is computed by

$$\epsilon = \frac{1}{2} \sum_{i=1}^{N} \|O_i - y_i\|^2, \tag{7}$$

where y_i is the label of output. The weights w_{ij} will be adjusted iteratively by the partial derivative of the distance ϵ and the learning rate γ;

$$\Delta w_{ij} = -\gamma \times \frac{\partial \epsilon}{\partial w_{ij}}. \tag{8}$$



label	2016	2015	2014
AAA	2	4	4
AA+	3	2	3
AA	3	5	5
AA-	14	11	10
A+	18	21	22
A	47	47	50
A-	52	52	60
BBB+	91	89	82
BBB	116	120	116
BBB-	91	94	95
BB+	82	83	81
BB	87	87	92
BB-	101	93	96
B+	69	81	69
B	63	67	87
B-	31	32	22
CCC+	16	10	6
CCC	7	1	0
CCC-	3	1	1
CC	5	0	0
C	0	0	0
D	0	1	0

Fig. 1. Credit ratings from 2014 to 2016 for the U.S. data.

3.4. Stacking

Ensemble learning method is another approach to improve the prediction accuracy for credit ratings; they are composed of several base learners, then the ensemble learning algorithm is trained to make a final prediction using all the outputs of the base learners. Ensemble learning method typically yields better performance than any single learning models. It also has been successfully used on both supervised learning tasks and unsupervised learning tasks. Wolpert (1992) proposed the Stacking ensemble learning method, which involves training a learning algorithm to combine the predictions of several other learning algorithms. Compared to bagging and boosting ensemble methods, Stacking is normally used to combine different types of the base learning algorithm, it can avoid the over-fitting effectively (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). The first step of stacking is to predict training set and testing set with some base level classifiers, and then use these predictions as features for high-end learner. The original data and the base learners construct the new datasets by ordinary cross-validation. The second step is that the high-end learning algorithm is employed to aggregate the base classifiers. Although an arbitrary high-end learning algorithm is used, Stacking can theoretically represent any of the ensemble techniques. In general, logistic regression is often used as the high-end learner.

4. Data and experimental results

4.1. Real world dataset

To validate the performance of the model, we first apply three different real-world datasets from the U.S., each of which consists of

Table 1
28 financial features.

Assets - Total	Sales/Turnover (Net)
Cash	Stockholders Equity - Total
Debt in Current Liabilities - Total	Interest and Related Expense - Total
Long-Term Debt - Total	Market Value - Total Fiscal
Earnings Before Interest	Book Value Per Share
Gross Profit (Loss)	Common Equity - Liquidation Value
Liabilities - Total	Comprehensive Income - Parent
Retained Earnings	Employees
Total Debt/Total Asset	Inventories - Total
Total Asset/Total Liabilities	Earnings Per Share from Operations
EBTI/Total Assets	Revenue - Total
Gross Profit/REV	Operating Activities - Net Cash Flow
EBTI/REV	Financing Activities - Net Cash Flow
Dividends per Share - Pay Date - Calendar	Net Cash Flow

Table 2
Credit ratings mapping.

New rating classes	Old ratings
Class A:	AAA, AA+, AA, AA-, A+, A, A-
Class B:	BBB+, BBB, BBB-
Class C:	BB+, BB, BB-
Class D:	B+, B, B-, CCC+, CCC, CCC-, CC, C, D

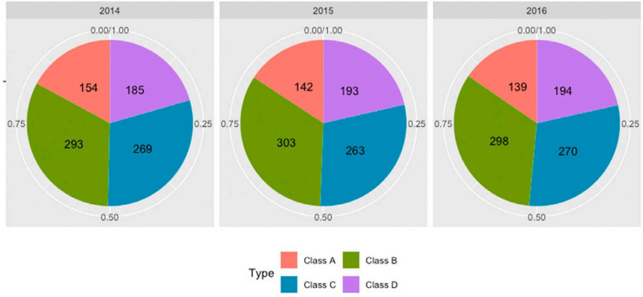


Fig. 2. Distribution of new rating classes from 2014 to 2016 for the U.S. data.

901 publicly-traded firms with 28 financial features that come from Quarterly reports[1] of the firms. The financial features were collected from Bloomberg, a major financial institution which provides the financial data service in North America. The credit ratings of the firms were obtained from Standard and Poor's (S&P). Corporate credit rating is a process in which a grade $w \in \Omega$ from a predefined rating scale Ω is assigned to a company. The rating scale of Standard and Poor's (S&P) is Ω = {AAA, AA+, AA, AA-, A+, A, A-, BBB+, BBB, BBB-, BB+, BB, BB-, B+, B, B-, CCC+, CCC, CCC-, CC, C, D}; a total of 22 grades that are ordered from AAA rating, the most promising for investors, to D rating, the most risky one. Fig. 1 illustrates the distribution of the credit ratings from 2014 to 2016 for U.S data. The original financial features include firm's size, financial structure, detailed financial records, ability to paying a debt, etc., which are known to affect the prediction of credit rating. These financial features are listed in Table 1.

In each dataset, the respective numbers of firms for CCC+ or below are very small. In fact, CCC+ or below ratings are usually treated similarly in the market due to high risk meaning that the debtor is currently highly vulnerable to no payment. To avoid an imbalance dataset, we group the ratings into four classes as listed in Table 2. After grouping them, the distribution of new rating classes is shown in Fig. 2.

4.2. Data preprocessing

The data preprocessing usually has a significant impact on the prediction performance of AI algorithm. It operates a linear transformation on the research data so that it can be used as the inputs for AI algorithms through the following two steps. The first step is that we convert the categorical credit ratings to numerical data, e.g., class A is converted to "1", ..., and class D is converted to "4". Some researchers use other labels to classify credit ratings; for instance, Zhong et al. (2014) used real numbers in $[-1, 1]$ for labels of credit rating. In order to mitigate the size effect, we apply min–max normalization to all input variables. Min–max normalization performs a linear transformation on the original data. It is often applied to improve the model performance, because it ensures that the larger value of an input variable does not overwhelm the smaller value of features. Suppose X_{min} and X_{max} are the minimum and maximum values of feature X. The mapping of Min–max normalization is computed in the following:

$$X_{\text{new}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}(X_{\text{new-max}} - X_{\text{new-min}}) + X_{\text{new-min}}, \tag{9}$$

where $X_{\text{new-max}}$ and $X_{\text{new-min}}$ represent the maximum and minimum values of the range, respectively. In our study, we put all variables in the range $[0, 1]$.

4.3. Feature selection

Feature selection is one of the core techniques in machine learning, which has significantly impacted the model performance. The benefit of feature selection is to reduce the dimensionality of the features. Performing this step not only makes the calculations faster and the parameters easier to be tuned, but also may improve classification accuracy. There exist many methods of feature selection in machine learning such as Mutual information, Information gain, Correlation-based filter, PCA, and Chi-squared test. The varieties of feature selection methods are usually divided into Filters and Wrappers. Filters perform feature selection based on the characteristics of data itself, and they operate independently of any learning algorithms by estimating the usefulness of features using an evaluation function. Features that are not expected to provide valuable information for classification are filtered out of the dataset before classification starts. Generally, mutual information, information gain, and correlation-based filter methods belong to the Filters. Filters are usually less computationally intensive than wrappers, but Filters provide a feature set which does not depend on a specific type of predictive model, which means a feature set from a filter is more general than the set from wrappers (Hajek & Michalak, 2013). Filters have also been used as a preprocessing step for wrappers methods, allowing a wrapper to be used on more significant problems. Many Filters methods provide a feature ranking rather than an explicit best feature subset, and the cut-off point in the ranking is usually chosen by cross-validation. Wrappers use some types of enumeration algorithms to explore the space of feature subsets since the number of all possible feature subsets is large. Thus, it is necessary to employ a search procedure that only iterates over a portion of all of the possible subsets. Overall, Wrappers feature selection is much slower than Filters due to large subsets, but Wrappers may produce better results on the specific type of prediction method. Hua, Tembe, and Dougherty (2009) proved that Wrappers performed better than Filters on a large sample size. The search process may be methodical such as a best first search, and it also may be stochastic such as a random hill-climbing algorithm.

In our study, we employ the Correlation-based filter method, because it can generate a general feature subset, which is compatible with different AI methods. Correlation-based feature selection is a filter algorithm that ranks feature subsets according to a correlation based

[1] Quarterly reports are issued by companies every three months, and include key accounting and financial data for a company. The Securities and Exchange Commission (SEC) requires issuers of publicly traded shares to file Form 10-K annually and Form 10-Q quarterly within 60 days of the end of applicable period.

Table 3
Correlation-based feature selection.

Ranked attributes:			
0.2015	Total Debt/Total Asset[a]	0.1307	Retained Earnings[a]
0.1767	Market Value - Total - Fiscal[a]	0.1205	Interest and Related Expense - Total[a]
0.1690	Earnings Before Interest[a]	0.1171	Book Value Per Share[a]
0.1673	Common Equity - Liquidation Value[a]	0.1128	Assets - Total[a]
0.1672	Revenue - Total[a]	0.1037	Cash[a]
0.1672	Sales/Turnover (Net)[a]	0.0986	Liabilities - Total[a]
0.1665	Stockholders Equity - Total[a]	0.0958	EBTI/Total Asset[a]
0.1636	Gross Profit (Loss)[a]	0.0931	EBTI/REV[a]
0.1531	Earnings Per Share from Operations[a]	0.0913	Net Cash Flow
0.1517	Dividends per Share - Pay Date - Calendar[a]	0.0692	Debt in Current Liabilities - Total
0.1473	Operating Activities - Net Cash Flow[a]	0.0676	Gross Profit/REV
0.1415	Comprehensive Income - Parent[a]	0.0660	Financing Activities - Net Cash Flow
0.1393	Long-Term Debt - Total[a]	0.0639	Inventories - Total
0.1376	Employees[a]	0.0503	Total Asset/Total Liabilities

[a] Indicates the selected features.

heuristic evaluation function. The bias of the evaluation function is toward subsets that contain features that are highly correlated with the class. Here, the stepwise subset method can be applied to find the optimal stopping point; we start with an empty set of features, and the feature set is expanded based on the ranking of the correlation-based method. The procedure is stopped when the addition of more variables provides no improvement in classification accuracy. Finally, the 22 features have been selected out of 28 features, which is operated by WEKA. The rank of each feature is listed in Table 3.

4.4. Model performance

In order to validate the effectiveness of our model, we conduct the experiments using the following eight learning algorithms, namely; (1) One-vs-one, (2) One-against-all, (3) Random Forest, (4) ECOC, (5) OMSVM Forward, (6) OMSVM Backward, (7) Stacking, and (8) ANN. For each dataset, twenty percent of the data is used for validation, and the remaining eighty percent is used for training. We adopt the five-fold stratification cross-validation for each dataset. Stratification is the process of rearranging the data to ensure each fold has the same class distribution as the original dataset, and it is a better scheme both in terms of bias and variance, compared to conventional cross-validation. More precisely, stratification is designed in a supervised way for classification and aims to ensure each class is approximately equally represented across each training and testing folds. As a result, we produce the experimental results under the eight learning algorithms for each dataset. The parameters for each method are described as follows.

4.4.1. Model parameters

In the case of multi-class SVM, Gaussian radial basis is used as the kernel function. There are two parameters, the upper bound C and the kernel parameter γ, that have very important roles in determining the performance of multi-class SVM; selecting improper parameters may cause the over-fitting or under-fitting. In our study, the optimal parameters C and γ of multi-class SVM are obtained by Grid Search. The RF model is one of the bagging ensemble learning algorithms, there are many parameters such as "n estimators", "max depth", "max feature", etc. However, the most important parameters of RF are "n estimators", "max depth", and "min samples split", that have a huge impact on the model performance. The best value of them can be obtained by Grid Search.

Table 4
Model parameters.

Model	Parameters	Parameter values
One-vs-one	C, gamma, kernel function	C=10000; gamma=0.1; RBF kernel
One-against-all	C, gamma, kernel function	C=10000; gamma=0.01; RBF kernel

Random Forest	n_estimators,
	max_depth,
	min_samples_split
		n_estimators=20;
		max_depth=20;
		min_samples_split=8

ECOC	C, gamma	C=10000; gamma=0.01;
OMSVM Forward	C, gamma, kernel function	C=11356; gamma=0.1; RBF kernel
OMSVM Backward	C, gamma, kernel function	C=11356; gamma=0.1; RBF kernel

ANN	learning rate;
	hidden layer size;
	activation function
		learning rate=0.0001;
		number of hidden layer =7;
		ReLu

The parameters of ANN model depend on the structure of ANN. The number of hidden layers is a crucial parameter, too many layers may cause vanishing gradient or over-fitting, too less may lead to bad prediction accuracy. In our experiment, we tested with the varying number of hidden layers on the range of $[3, 30]$; the model with 7 hidden layers has the highest prediction accuracy in five-fold cross-validation. The initial learning rate is another critical parameter in ANN, which determines the efficiency of the ANN model. If the learning rate is too small, the ANN model needs more time to be trained; if it is too large, the ANN model may fail to converge. Thus, we tested with the initial learning rate from the range $[0.0001, 0.1]$; the initial learning rate of 0.0001 has a good performance in five-fold cross-validation. In order to avoid the vanishing gradient, the ReLU is employed as our activation function. In addition, Luo, Wu, and Wu (2017) and Zhong et al. (2014) pointed out that increasing the nodes of each hidden layer will not improve performance when the size of each hidden layer is large enough. The optimal parameters values for each model are shown in Table 4.

4.4.2. Experimental results

Table 5 summarizes our experimental results in credit rating prediction on the U.S. data for the year 2014, 2015, and 2016. For each model, the average prediction accuracy of five-fold cross-validation is displayed. Accuracy is the percentage of correctly classified instances and provides a measure for the capability to make an accurate prediction. As shown in Table 5, RF provides the highest prediction accuracies 68.26%, 67.82%, and 68.92% for the year 2014, 2015 and 2016, respectively; ECOC provides the worst prediction accuracies 63.89% and 65.38% for the year 2014 and 2016, respectively; and SVM (One-vs-one) provides the worst prediction accuracy for the year 2015. In addition, the ANN model provides the second highest prediction accuracies 67.74% and 69.81% for the year 2015 and 2016. For OMSVM Forward and Backward, the results highlight that OMSVM (Forward and Backward) perform better than SVM (One-vs-one and One-against-all) for each dataset.

Table 5
Model performance.

	One-vs-one	One-against-all	RF	ECOC	OMSVM (Forward)	OMSVM (Backward)	Stacking	ANN
Data 2016	67.81	66.26	68.92	65.38	67.92	68.48	68.39	68.81
Data 2015	64.93	65.75	67.82	65.04	65.81	66.59	66.3	67.74
Data 2014	65.19	64.55	68.26	63.89	66.64	66.12	67.11	67.07

Table 6
Metrics based on credit rating history.

Feature name	Formula	Description
Momentum	$C_t - C_{t-4}$	It measures the change in credit rating over a year.
Disparity	$C_t - MA$	It measures the distance between its current rating and moving average over the past T years.
Disparity ratio	$\frac{C_t - MA}{H_{t,T} - L_{t,T}}$	It measures the ratio of the difference between its current rating and moving average to the difference between the highest and lowest credit ratings.
First-order variation	$\sum_{k=0}^{4T-1} \left\lvert C_{t-k} - C_{t-k-1} \right\rvert$	It measures the volatility (or mobility) of credit rating for the past T years.
Second-order variation	$\sum_{k=0}^{4T-1} (C_{t-k} - C_{t-k-1})^2$	It measures the volatility of credit rating and focuses more on big movements.

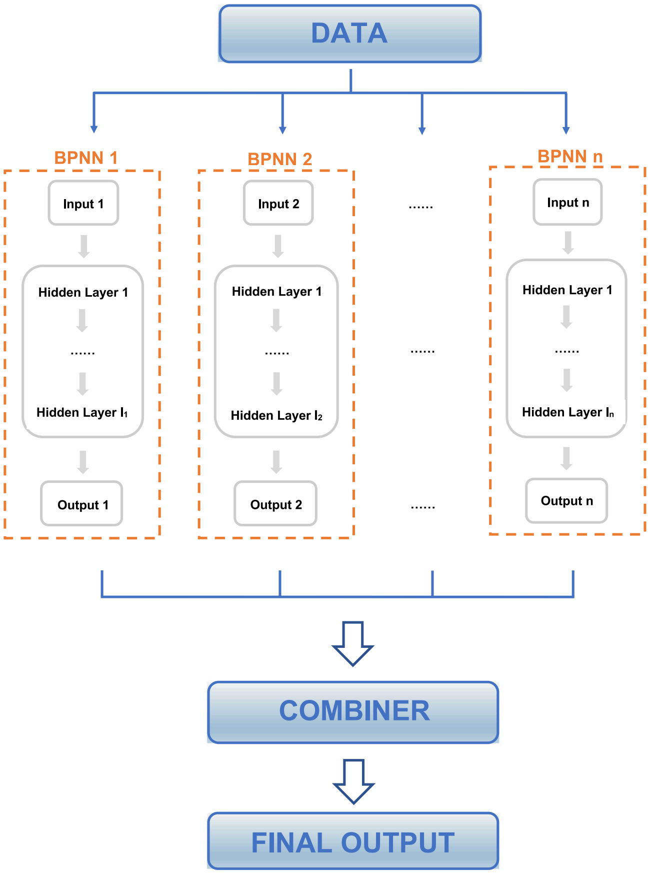


Fig. 3. The general formulation process of Parallel ANNs.

5. Proposed approaches

5.1. Metrics based on credit rating history

Inspired by Kim (2003) and Masoud (2014), we extract the features from the history of a firm's credit rating and construct the metrics which characterize past changes in the credit rating of the firm. The motivation behind this is that the current credit rating may be highly correlated with past changes in the firm's credit rating, and the prediction performance may be enhanced by using the information on rating history. In fact, credit rating is an indicator of the financial situation of a company, and there may be a meaningful relationship between the current credit rating and the company's recent financial performance. Before we build the metrics, we first convert the categorical data to numerical data, that is, AAA is converted to "22", AA+ is converted to "21", and so on; and we consider historical data for credit rating on a quarterly basis for the past $T (T \geq 1)$ years. For convenience purposes, we define some notations as follows. Let C_t be the credit rating at current time t, and MA be the moving average of credit ratings for the past T years, that is, $MA = \frac{1}{4T} \sum_{k=0}^{4T-1} C_{t-k}$ where the time unit is quarter. We denote by $H_{t,T}$ and $L_{t,T}$ the highest and the lowest credit ratings in the period, respectively, i.e., $H_{t,T}$ and $L_{t,T}$ are the maximum value and minimum value in the set $\{C_{t-k}, 0 \leq k \leq 4T-1\}$. We propose five metrics that include Momentum,[2] Disparity, Disparity ratio,[3] First-order variation,[4] and Second-order variation. The descriptions and formulations are shown in Table 6.

5.2. Parallel ANNs

As mentioned in Section 2, several articles have validated the capability of ANN and ensemble learning methods for credit rating assessment, which have been widely applied in the financial industry. However, the intensive investigation of neural networks ensemble for credit risk evaluation has not formulated a convincing theoretical foundation and overall process model yet (Yu et al., 2008). Recently, the parallel ANN training techniques (Zhang et al., 2016) have been implemented on Shared Memory Architecture (Araiijo, Teixeira, Camargo, & Almeida, 2003) and on Distributed Memory Architecture (Thulasiram et al., 2003). The parallel ANN training technique can accelerate the training process by dividing the dataset into multiple subsets. Inspired by Zhang et al. (2016), we apply the parallel training technique to deal with historical financial data. The motivation of our proposed PANNs model is that the current credit rating may be highly correlated with the recent financial performance of the firm in addition to the current financial report, so it would contribute to the improvement of credit rating prediction if useful information can be extracted from historical financial data. Based on this intuition, we apply the general parallel training technique to fit historical financial data by constructing several independent ANNs to achieve the goal of improving the prediction accuracy. Each independent ANN model we construct deals with each year's financial data; the final output is aggregated from the base classifiers by the weighted average algorithm.

The proposed Parallel ANNs model consists of three stages; the first stage is to create the neural network classifiers; the second stage is to

[2] It indicates whether the firm was upgraded (positive value) or downgraded (negative value) into its current rating.

[3] Disparity ratio is defined as 0 in case $H_{t,T} - L_{t,T} = 0$.

[4] It represents the cumulative sum of the changes in credit rating for a given period of time.

integrate multiple classifiers into an ensemble output; the third stage is to perform the learning process of the PANNs model. The general architecture of the Parallel ANNs ensemble learning model is illustrated in Fig. 3.

5.2.1. Neural network classifier

According to Hansen and Salamon (1990), an ensemble classifier performs more accurate than any of its individual members if the base classifiers are accurate and diverse. An effective ensemble method consisting of diverse models with much disagreement is more likely to have a good generalization performance (Wang et al., 2005). For the ensemble neural network model, several methods have been investigated for the generation of ensemble members making different errors (Sharkey, 1996). These methods include using different initial conditions, using different network architecture, using different training data, and using different training algorithms. Such methods basically rely on varying the parameters related to design and to the training of neural networks. In our study, the neural network is employed as our classifier because of two main reasons. First, the neural network with an identity activity function in the output unit and activity function in the middle-layer units can approximate any continuous function arbitrarily well given a sufficient amount of middle layer units. Second, the different input vectors can generate different architectures of neural network by changing the number of hidden layers and the number of nodes in each layer, etc.; the different architectures of neural networks can contribute to having an effective ensemble model that produces an excellent generalization performance.

Suppose we have n years financial data $\boldsymbol{X}$ for a firm, it can be represented as follows:

$$\boldsymbol{X} = \begin{bmatrix} X_{11}, & X_{12}, & \cdots, & X_{1m} \\ X_{21}, & X_{22}, & \cdots, & X_{2m} \\ \vdots & \ddots & & \vdots \\ X_{n1}, & X_{n2}, & \cdots, & X_{nm} \end{bmatrix} \tag{10}$$

where m is the number of inputs for each year. Each row of the matrix represents yearly financial data, i.e., $\boldsymbol{X}_i$ represents the financial features for the ith year and is written as $\boldsymbol{X}_i = (X_{i1}, X_{i2}, \ldots, X_{im})$, where $i = 1, 2, \ldots, n$. We divide the whole dataset into n subsets, each subset generates a BPNN model, so the PANNs model consists of n BPNN models. Let $\boldsymbol{Y}_l$ be the label of each firm for $l = 1, 2, \ldots, q$, where q is the number of firms in the dataset. Assume each $\boldsymbol{Y}_l$ has s neurons output and is written as $\boldsymbol{Y}_l = (y_{l1}, y_{l2}, \ldots, y_{ls})^{\boldsymbol{T}}$, where $\boldsymbol{T}$ is the transpose of a row vector. Let $\widetilde{\boldsymbol{Y}}^{(i)}$ be the prediction of the ith BPNN model and $\widetilde{\boldsymbol{Y}}^{(i)} = (\hat{y}_1^{(i)}, \hat{y}_2^{(i)}, \ldots, \hat{y}_s^{(i)})^{\boldsymbol{T}}$. Then, the output vector $\widetilde{\boldsymbol{Y}}^{(i)}$ can be represented as:

$$\widetilde{\boldsymbol{Y}}^{(i)} = \begin{bmatrix} \hat{y}_1^{(i)} \\ \hat{y}_2^{(i)} \\ \vdots \\ \hat{y}_s^{(i)} \end{bmatrix} = \begin{bmatrix} f_1(\boldsymbol{X}_i) \\ f_2(\boldsymbol{X}_i) \\ \vdots \\ f_s(\boldsymbol{X}_i) \end{bmatrix} = \begin{bmatrix} b_0^{(1)} + \sum_{k=1}^{h_i} w_k^{(1)} g(\sum_{j=1}^{m} w_{ij}^{(1)} X_{ij} + b_k^{(1)}) \\ b_0^{(2)} + \sum_{k=1}^{h_i} w_k^{(2)} g(\sum_{j=1}^{m} w_{ij}^{(2)} X_{ij} + b_k^{(2)}) \\ \vdots \\ b_0^{(s)} + \sum_{k=1}^{h_i} w_k^{(s)} g(\sum_{j=1}^{m} w_{ij}^{(s)} X_{ij} + b_k^{(s)}) \end{bmatrix} \tag{11}$$

where b_k is the bias on the kth unit, w_{ij} is the connection weight between layers of the ith BPNN model, $g(\cdot)$ is the transfer function of hidden layers, and h_i is the number of hidden nodes for the ith BPNN.

5.2.2. Ensemble method

In general, there are various methods to aggregate base learners such as ranking, simple averaging, and majority voting. Majority voting is the most widely used ensemble strategy for classification problems. The final decision is determined by the ensemble members' voting. Typically, it takes over half the ensemble to agree a result for it to be accepted as the final output of the ensemble method. However, majority voting ignores the fact some neural network that lies in a minority sometimes does produce the correct results (Yang & Browne, 2004). For the simple averaging method, the final output can be obtained by averaging the sum of each output of the ensemble members. It is more useful when the variances of ensemble members are different. Nevertheless, this method treats each ensemble member equally, it does not stress those ensemble members who can make more contribution to the output generalization. Ranking is where the members of an ensemble learning 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 (Yu et al., 2008). In our model, we adopt the weighted average ensemble algorithm where the contribution of each member to the final prediction is weighted by the performance of the model. Then, the combiner combines each BPNN$_i$'s outputs, and it can be formulated as

$$\text{PANNs} = \sum_{i=1}^{n} w_i^f \times \text{BPNN}_i, \tag{12}$$

where w_i^f is the aggregate level weight of BPNN$_i$. For the convenience of notations to express the final output of PANNs, we denote by the $\widetilde{\boldsymbol{Y}}_l^f$ the final output of the lth firm. From Eq. (11), we have

$$\widetilde{\boldsymbol{Y}}_l^f = \begin{bmatrix} \hat{y}_{l1}^f \\ \hat{y}_{l2}^f \\ \vdots \\ \hat{y}_{ls}^f \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} w_{i1}^f \hat{y}_{l1}^{(i)} \\ \sum_{i=1}^{n} w_{i2}^f \hat{y}_{l2}^{(i)} \\ \vdots \\ \sum_{i=1}^{n} w_{is}^f \hat{y}_{ls}^{(i)} \end{bmatrix}$$
$$= \begin{bmatrix} \sum_{i=1}^{n} w_{i1}^f [b_0^{(1)} + \sum_{k=1}^{h_i} w_k^{(1)} g(\sum_{j=1}^{m} w_{ij}^{(1)} X_{ij}^{(l)} + b_k^{(1)})] \\ \sum_{i=1}^{n} w_{i2}^f [b_0^{(2)} + \sum_{k=1}^{h_i} w_k^{(2)} g(\sum_{j=1}^{m} w_{ij}^{(2)} X_{ij}^{(l)} + b_k^{(2)})] \\ \vdots \\ \sum_{i=1}^{n} w_{is}^f [b_0^{(s)} + \sum_{k=1}^{h_i} w_k^{(s)} g(\sum_{j=1}^{m} w_{ij}^{(s)} X_{ij}^{(l)} + b_k^{(s)})] \end{bmatrix}, \tag{13}$$

where w_{ij}^f is the aggregate level weight for the jth node of BPNN i, $\hat{y}_{lj}^f$ is the final output for the jth node of firm l, and $X_{ij}^{(l)}$ represents n years financial data for the lth firm.

5.2.3. Learning process

For the classification task, PANNs have to be trained by the BP algorithm. In our proposed model, the objective is to minimize the cross-entropy loss function, which is defined as

$$L = -\sum_{l=1}^{q} \sum_{j=1}^{s} y_{lj} \log(\sum_{i=1}^{n} w_{ij}^f \hat{y}_{lj}^{(i)}), \tag{14}$$

where y_{lj} is the label of the jth node for firm l. The model parameters are to be updated iteratively by a process of minimizing the cross-entropy loss function L (14). By adjusting weights w_{ij}, all input vectors are correctly mapped to their corresponding output vectors. The optimal weights are to be obtained by stochastic gradient descent, layer by layer. The direction and magnitude change Δw_{ij} is computed as

$$\Delta w_{ij} = -\frac{\partial L}{\partial w_{ij}} \gamma, \tag{15}$$

where $0 < \gamma < 1$ is the learning rate that controls the algorithm's convergence rate. The total error is propagated back, layer by layer, from the output units to the input unit. The process is executed during each iteration of the Back-propagation algorithm until the total error converges.

Table 7
Summary of the ensemble learning methods for credit rating prediction.

Method	Dataset	Ensemble method	Prior studies
Parallel ANNs	Historical financial data, divided into disjoint data sets for each year	Weighted average	
Multi-agent ANNs	One year financial data, boosting or bagging algorithm applied to generate training data sets	Mean & majority voting	Yeung et al. (2007) Yang and Browne (2004)
Multi-stage ANNs	One year financial data, bagging algorithm applied to generate different training subsets	Decorrelated maximization algorithm used to select ANNs and five different learning strategies for the aggregate output	Yu et al. (2008)
Evolutionary ANNs	Time series dataset	Rank-based and softmax combination	Donate et al. (2013)

Table 8
Model performance with 3-year metrics.

	One-vs-one	One-against-all	RF	ECOC	OMSVM (Forward)	OMSVM (Backward)	Stacking	ANN
Data 2016	68.59[a]	67.37[a]	69.36[a]	66.26[a]	69.25[a]	69.37[a]	69.03[a]	70.03[a]
Data 2015	67.56[a]	67.49[a]	69.32[a]	66.48[a]	68.03[a]	70.48[a]	68.48[a]	68.81[a]
Data 2014	66.59[a]	66.59[a]	69.26[a]	66.93[a]	68.15[a]	66.37[a]	68.00[a]	67.82[a]

[a] *Indicates that indicator has been improved.*

5.2.4. Comparison with previous studies

The main studies of ensemble learning in credit rating assessment are summarized in Table 7. These ensemble studies have employed ANN as the base learner since ANN has an excellent generalization performance, and the different architectures of ANN model can contribute to generating an effective ensemble classifier. Yang and Browne (2004) and Yeung et al. (2007) applied the general bagging and boosting ensemble methods to generate different training datasets. The primary ensemble learning methods (mean and majority voting) are applied to the multi-agent ANNs model. Yu et al. (2008) split the original dataset into a training set and a testing set, then applied the bagging algorithm to generate different training subsets. Compared to primary ensemble methods, Yu et al. (2008) proposed a reliability-based ensemble strategy to make the final decision of the ensemble at the measurement level, in which Maximum strategy, Minimum strategy, Median strategy, Mean strategy, and Product strategy can be used to integrate the individual ensemble members. Donate et al. (2013) considered the single feature with time series data $y_{t-1}, y_{t-2}, \ldots, y_{t-k}$, where $t-1, \ldots, t-k$ is a set of time lags used, and t is the current time, then applied two different methods for integrating the ensemble learning, namely Rank-based and softmax combination.

6. Experimental results

6.1. Results for new metrics on credit rating history with US data

To thoroughly validate the superior performance of our model, we apply three different real-world datasets from the U.S. (2014, 2015, and 2016), which were described in Section 4.1. The experiments are conducted by using the past three years (12 quarters) of credit ratings data to extract the features for each dataset. To ensure that the proposed metrics provide valuable information for the multi-class classification of credit rating, we operate the stepwise method on feature selection. Starting with the 22 selected features, we test whether adding each of the five metrics improves the classification accuracy or not by using five-fold cross-validation. The procedure is stopped until adding more metrics provides no improvement in classification accuracy. Finally, Disparity and First-order variation are selected and added in the feature space in this experiment.

Table 8 presents the experimental results of the conventional AI models with the new metrics constructed in Section 5.1, and the average result of five-fold cross-validation is displayed for each dataset. As shown in this table, the average prediction accuracies of SVM (One-vs-one) with the three-year metrics are 68.59%, 67.59%, and 66.59% for the year 2016, 2015 and 2014, which have been improved by 1.1%, 2.7%, and 1.4% respectively, in comparison to those of SVM (One-vs-one) in Table 5. For the OMSVM (Forward) method, the average prediction accuracies with the three-year metrics have been increased by 1.3%, 2.2%, and 1.5% for the year 2016, 2015 and 2014, respectively. From the experimental results, we found that the expansion of the feature space by extracting the features from credit rating history leads to better predictive performance for corporate credit rating assessment. It is impressive that for each dataset the prediction accuracies have been improved for all conventional methods. Consequently, our approach provides an effective and efficient solution for the classification problem of credit rating. In addition, we conducted experiments by using the past five years of credit ratings; the performance has been improved compared to the models in Table 5, but not as good as the one using the past three years of data.

6.2. Results for PANNs with US data

In our experiments, we validate the performance of PANNs by applying two different scenarios; one is to predict the 2016 credit ratings by using the previous two years (2015, 2016) and three years (2014, 2015, and 2016) of financial data, and the other is to predict the 2015 ratings by using the past two years (2014, 2015) and three years (2013, 2014, and 2015) of financial data. The financial data (2013, 2014, 2015, and 2016) were collected from Bloomberg. After obtaining the raw data, min–max normalization is applied to mitigate the size of the effect as described in Section 4.2. All historical financial data have been put in the range $[0,1]$ by Eq. (9).

To evaluate the model efficiency, we apply five indicators to measure the model performance, which includes Accuracy, Precision, Recall, F_1 Score, and AUC. Precision is the fraction of relevant instances among the retrieved instances, which is calculated by dividing the true positive by the sum of the true positive and false positive. Recall (also called sensitivity) is the fraction of the total amount of relevant

Table 9
Model performance of PANNs.

		Accuracy	Precision	Recall	F_1 Score	AUC
2016	ANN	68.51%	69.99%	68.40%	68.93%	88.92%
2016	PANNs(2)	71.32%	72.56%	70.85%	71.30%	90.72%
2016	PANNs(3)	71.14%	73.66%	68.68%	70.11%	90.49%
2015	ANN	67.75%	69.04%	67.48%	68.11%	88.90%
2015	PANNs(2)	69.95%	70.63%	69.47%	69.77%	90.49%
2015	PANNs(3)	70.03%	70.86%	68.52%	69.25%	90.46%

instances that were actually retrieved, which is calculated by dividing the true positive by the sum of the true positive and false negative. These two performance metrics measure the true positive rate and positive predictive rate. F_1 score is a measure that combines precision and recall, i.e., the harmonic mean of precision and recall. The Receiver Operating Characteristic (ROC) curve is a graphical representation of the true positive rate against the false negative rate, and the Area Under the Curve (AUC) is simply the area under the ROC curve. AUC values of 0.5 and 1 correspond to random and perfect prediction, respectively.

In the first experiment, financial data of the firms for the year 2015 is used as input 1 and financial data for the year 2016 as input 2 for the two-year case; financial data of the firms for the year 2014 is used as input 1, financial data for the year 2015 as input 2, and financial data for the year 2016 as input 3 for the three-year case. The credit ratings for the year 2016 are assessed by using past two-year and three-year historical financial data. In the second experiment, we set the financial data for the year 2014 and 2015 as input 1 and input 2 for the two-year case; financial data for the year 2013 as input 1, financial data for the year 2014 as input 2, and financial data for the year 2015 as input 3 for the three-year case. The credit ratings for the year 2015 are assessed by using past two-year and three-year historical financial data. For each experiment, twenty percent of the data are used for validation, and the remaining eighty percent are used for training. In our experiment, each year's financial data generates an individual BPNN. The output of each BPNN is the probability that behavior happens, and each BPNN is trained by the traditional BP learning algorithm. The weights will be updated by minimizing the categorical cross-entropy layer by layer. The algorithm of PANNs is displayed in Algorithm 1.

Algorithm 1 Parallel ANNs Algorithm

```
Date set $D_1 := \{X_{11}, X_{12}, \dots, X_{1m}\}$
Data set $D_2 := \{X_{21}, X_{22}, \dots, X_{2m}\}$
$\vdots$
Data set $D_n := \{X_{n1}, X_{n2}, \dots, X_{nm}\}$
Label $Y := \{Y_1, Y_2, \dots, Y_n\}$
Process:
Network 1 depth: $I_1$
Network 1 weights vectors of model $W_1(i), i = 1, 2, \dots, I_1$.
The bias vectors of the network 1 $B_1(i), i = 1, 2, \dots, I_1$.
Network 2 depth: $I_2$
Network 2 weights vectors of model $W_2(i), i = 1, 2, \dots, I_2$.
The bias vectors of network 2 $B_2(i), i = 1, 2, \dots, I_2$.
$\vdots$
Network n depth: $I_n$
Network n weights vectors of model $W_n(i), i = 1, 2, \dots, I_n$.
The bias vectors of network n $B_n(i), i = 1, 2, \dots, I_n$.
$h_1(0) = D_1$;
For $K = 1, 2, 3, ..., I_1$;
$a_1(K) = B_1(K) + W_1(K)h_1(K-1)$;
$h_1(K) = f(a_1(K))$
end for
Output 1 $= h_1(I_1)$;
$h_2(0) = D_2$;
For $K = 1, 2, 3, ..., I_2$;
$a_2(K) = B_2(K) + W_2(K)h_2(K-1)$;
$h_2(K) = f(a_2(K))$
end for
Output 2 $= h_2(I_2)$;
$\vdots$
$h_n(0) = D_n$;
For $K = 1, 2, 4, ..., I_n$;
$a_n(K) = B_n(K) + W_n(K)h_n(K-1)$;
$h_n(K) = f(a_n(K))$
end for
Output n $= h_n(I_n)$;
Final output = g(Output 1, Output 2, ⋯, Output n)
```

Table 9 presents the experiment results of PANNs on the prediction of 2015 and 2016 credit ratings by using the past two years and three years of financial data. The first three rows of Table 9 are the experimental results of the year 2016, and last three rows are the prediction results of the year 2015. For convenience purposes, we denote by PANNs(2) the PANNs method with the past two years of financial data, and by PANNs(3) the PANNs method with the past three years of data.

As shown in Table 9, it is illustrated that the average prediction accuracy and AUC of PANNs(2) are slightly higher than those of PANNs(3) for the year 2016, which implies that PANNs with the past two years of data works better on the 2016 credit ratings prediction than PANNs with the past three years of data. Compared to established ANN, the prediction accuracies of PANNs(2) and PANNs(3) have been enhanced by 2.8% and 2.6%, respectively. On the other hand, it reveals that the average accuracy of PANNs(3) is slightly higher than that of PANNs(2) for the year 2015, while the AUC of PANNs(2) is slightly higher than that of PANNs(3). Compared to conventional ANN, the accuracies of PANNs(2) and PANNs(3) for the year 2015 have risen by 2.2% and 2.3%, respectively. Overall, the accuracy, precision, recall, F_1 score, and AUC of PANNs on the prediction of 2015 and 2016 credit ratings have been significantly improved, and the PANNs model has performed remarkably better than the conventional AI methods. Interestingly, the PANNs model with one more year of financial data may lead to lower prediction performance. For this phenomenon, Yu, Wang, and Lai (2005) pointed out that the neural network ensemble model does not follow the rule of "the more, the better". One reasonable interpretation is that the three years of financial data may have had more noise than the two years of data.

6.3. Model validation with additional data

In this subsection, we validate our approach with a Japanese dataset which is different in context, number and balance of data classes from the U.S. datasets described in Section 4.1.

We collected a Japanese corporate rating dataset from the Japan Credit Rating Agency,[5] and obtained comparable financial variables with those in the U.S. datasets from Bloomberg and WRDS.[6] The dataset covered financial variables and ratings from the year of 2015 to 2019. After filtering data with missing values, we obtained 292 companies in Japan with 27 financial features. We group the ratings into four classes; grouping AAA, AA+, AA, AA- as class A to represent highest credit quality with the lowest credit risk; grouping A+, A as class B to represent second best credit quality; grouping A- as class C; grouping BBB+, BBB, BBB-, BB+ as class D to represent the higher credit risk. The distributions of the credit ratings and the new rating classes are shown in Figs. 4 and 5, respectively.

[5] https://www.jcr.co.jp/en/ratinglist/corp.

[6] Wharton Research Data Service https://wrds-www.wharton.upenn.edu.

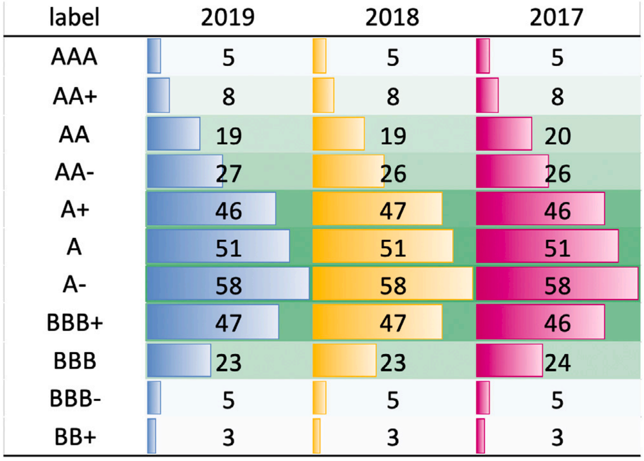
Table 10
Model performance on Japanese data.

	One-vs-one	One-against-all	RF	ECOC	OMSVM (Forward)	OMSVM (Backward)	Stacking	ANN
Data 2019	61.82	60.61	60.95	58.55	60.76	60.27	58.37	59.25
With new metrics	62.46[a]	61.98[a]	62.03[a]	59.59[a]	60.96[a]	61.30[a]	59.04[a]	59.38[a]

[a] *Indicates that indicator has been improved.*

Table 11
Model performance of PANNs on Japanese data.

		Accuracy	Precision	Recall	F_1 Score	AUC
2019	ANN	59.25%	56.8%	58.81%	58.05%	77.02%
2019	PANNs(2)	60.54%	57.77%	59.01%	58.93%	78.53%
2019	PANNs(3)	60.27%	56.71%	58.93%	57.75%	77.81%



label	2019	2018	2017
AAA	5	5	5
AA+	8	8	8
AA	19	19	20
AA-	27	26	26
A+	46	47	46
A	51	51	51
A-	58	58	58
BBB+	47	47	46
BBB	23	23	24
BBB-	5	5	5
BB+	3	3	3

Fig. 4. Credit ratings from 2017 to 2019 for Japanese data.

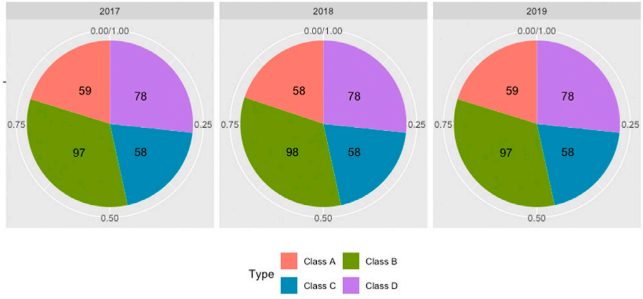


Fig. 5. Distribution of new rating classes from 2017 to 2019 for Japanese data.

The data preprocessing and feature selection are also implemented in this dataset, similarly as in Section 4.2. After applying the min–max normalization to eliminate the size effect, we operate the correlation-based feature selection. Twenty-one financial features have been selected from the total features we collected, which include all Twenty-two features in Table 3, except for Gross Profit (Loss).

In order to validate our approach of adopting new metrics on credit rating history introduced in Section 5.1, the experiments are conducted by using the past four years of annual credit ratings to extract the features for Japanese data, and operating the stepwise method on feature selection. Finally, Disparity, Momentum, and First-order variation are selected and added in the feature space in this experiment.

Table 10 presents the results of the conventional AI models without and with the new metrics, where the average result of five-fold cross-validation is displayed. As shown in this table, the average of prediction accuracies of all conventional AI models have been improved by adding new metrics. Among them, SVM (One-against-all) provides the largest improvement (1.37%), and Random Forest provides the second largest improvement (1.08%).

Table 11 presents the experiment results of PANNs on the prediction of 2019 credit ratings by using the past two years and three years of financial features. The prediction accuracies of PANNs(2) and PANNs(3) have been enhanced by 1.3% and 1%, respectively, and PANN(2) performs slightly better than PANN(3). Moreover, the precision, recall, F1 score, and the AUC of PANNs on the prediction of 2019 credit ratings are also improved.

7. Conclusion

Credit rating has become an efficient tool for financial institutions to discriminate the potential risky borrowers and manage credit risk. The predictive performance of the credit rating is critical to the profitability of financial institutions. A more accurate model can significantly reduce the cost of the credit industry and the loss of debt issuers.

In this paper, we propose two approaches to improve the prediction accuracy of credit rating. First, we construct new metrics to characterize past changes in credit rating and expanded the feature space with new input features. Second, we propose a novel ensemble structure of artificial neural networks, called parallel ANNs, which is designed to utilize historical financial data for credit rating assessment.

To validate the applicability of the proposed learning analytics methods, we apply two real-world cases of credit ratings from the U.S. and Japan. Our experiment results disclose that the metrics based on credit rating history can contribute to enhancing the prediction accuracy of credit rating. It is impressive that our approach has led to an improvement in prediction accuracy for all conventional learning methods. In addition, our experiment results illustrate that the PANNs model is superior to conventional AI techniques; the prediction accuracy has been roughly improved by 3% to 5%. Consequently, our two approaches provide an effective and efficient way to enhance the prediction accuracy in credit rating assessment.

Although we validated our proposed PANNs model has superior performance with real-world data, the limitation of our study is that we do not provide theoretical evidence to support our assertion that our approach is the most efficient among various conventional methods. Especially for PANNs, despite the model is intuitively designed to seek the relationship between the current credit rating and the previous financial performance of the firm, it does not follow the rule of “ the more, the better ”. In addition, it seems to be slightly slower than conventional methods with respect to time consumption. However, our experiment results reveal that the proposed PANNs ensemble learning model provides a promising solution in credit risk assessment, and furthermore, it has a great potential to other multi-class classification problems using historical data of features.

CRediT authorship contribution statement

Mingfu Wang: Investigation, Data curation, Writing - original draft, Visualization. **Hyejin Ku:** Conceptualization, Methodology, Writing - editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was partly supported by Natural Sciences and Engineering Research Council of Canada, Discovery grant 504316.

References

Abdou, H., Pointion, J., & EI-Masry, A. (2008). Neural nets versus conventional techniques in credit scoring in Egyptian banking. *Expert Systems with Applications*, *35*(3), 1275–1292.

Abellán, J., & Castellano, J. G. (2017). A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications*, *73*, 1–10.

Araiijo, M., Teixeira, E., Camargo, F., & Almeida, J. (2003). Parallel training for neural networks using PVM with shared memory. In *The 2003 congress on evolutionary computation, 2003, vol. 2* (pp. 1315–1322). IEEE.

Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science*, *49*(3), 312–329.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, *54*(6), 627–635.

Bequé, A., & Lessmann, S. (2017). Extreme learning machines for credit scoring: An empirical evaluation. *Expert Systems with Applications*, *86*, 42–53.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Cao, L., Guan, L. K., & Jingqing, Z. (2006). Bond rating using support vector machine. *Intelligent Data Analysis*, *10*(3), 285–296.

Chornous, G., & Nikolskyi, I. (2018). Business-oriented feature selection for hybrid classification model of credit scoring. In *2018 IEEE second international conference on data stream mining & processing (DSMP)* (pp. 397–401). IEEE.

Desai, V. S., Crook, J. N., & Overstreet, G. A., Jr. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research*, *95*(1), 24–37.

Dietterich, T. G., & Bakiri, G. (1994). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, *2*, 263–286.

Donate, J. P., Cortez, P., Sánchez, G. G., & De Miguel, A. S. (2013). Time series forecasting using a weighted cross-validation evolutionary artificial neural network ensemble. *Neurocomputing*, *109*, 27–32.

Friedman, J. H., et al. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, *19*(1), 1–67.

Hájek, P. (2011). Municipal credit rating modelling by neural networks. *Decision Support Systems*, *51*(1), 108–118.

Hajek, P., & Michalak, K. (2013). Feature selection in corporate credit rating prediction. *Knowledge-Based Systems*, *51*, 72–84.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (10), 993–1001.

He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications*, *98*, 105–117.

Hua, J., Tembe, W. D., & Dougherty, E. R. (2009). Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, *42*(3), 409–424.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1–2), 307–319.

Kim, K.-j., & Ahn, H. (2012). A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach. *Computers & Operations Research*, *39*(8), 1800–1811.

Klautau, A., Jevtić, N., & Orlitsky, A. (2003). On nearest-neighbor error-correcting output codes with application to all-pairs multiclass support vector machines. *Journal of Machine Learning Research (JMLR)*, *4*(Apr), 1–15.

Krebel, U.-G. (1999). Pairwise classification and support vector machines. *Advances in Kernel Methods: Support Vector Learning*, 255–268.

Lai, K. K., Yu, L., Wang, S., & Zhou, L. (2006). Credit risk analysis using a reliability-based neural network ensemble model. In *International conference on artificial neural networks* (pp. 682–690). Springer.

Laitinen, E. K. (1999). Predicting a corporate credit analyst's risk estimate by logistic and linear models. *International Review of Financial Analysis*, *8*(2), 97–121.

Li, X.-L., & Zhong, Y. (2012). *An overview of personal credit scoring: Techniques and future work*. Scientific Research Publishing.

Luo, C., Wu, D., & Wu, D. (2017). A deep learning approach for credit scoring using credit default swaps. *Engineering Applications of Artificial Intelligence*, *65*, 465–470.

Maldonado, S., Pérez, J., & Bravo, C. (2017). Cost-based feature selection for support vector machines: An application in credit scoring. *European Journal of Operational Research*, *261*(2), 656–665.

Masoud, N. (2014). Predicting direction of stock prices index movement using artificial neural networks: The case of libyan financial market. *British Journal of Economics, Management & Trade*, *4*(4), 597–619.

Reichert, A. K., Cho, C.-C., & Wagner, G. M. (1983). An examination of the conceptual issues involved in developing credit-scoring models. *Journal of Business & Economic Statistics*, *1*(2), 101–114.

Saia, R., Carta, S., & Fenu, G. (2018). A wavelet-based data analysis to credit scoring. In *Proceedings of the 2nd International conference on digital signal processing* (pp. 176–180).

Sharkey, A. J. C. (1996). On combining artificial neural nets. *Connection Science*, *8*(3–4), 299–314.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, *15*(1), 1929–1958.

Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., & Levy, S. (2004). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, *21*(5), 631–643.

Thomas, L. C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting*, *16*(2), 149–172.

Thulasiram, R. K., Rahman, R. M., & Thulasiraman, P. (2003). Neural network training algorithms on parallel architectures for finance applications. In *2003 International conference on parallel processing workshops, 2003. Proceedings* (pp. 236–243). IEEE.

Tsai, C.-F., & Chen, M.-L. (2010). Credit rating by hybrid machine learning techniques. *Applied Soft Computing*, *10*(2), 374–380.

Wang, Y., Wang, S., & Lai, K. K. (2005). A new fuzzy support vector machine to evaluate credit risk. *IEEE Transactions on Fuzzy Systems*, *13*(6), 820–831.

West, D. (2000). Neural network credit scoring models. *Computers and Operations Research.*

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–259.

Yang, S., & Browne, A. (2004). Neural network ensembles: combining multiple models for enhanced performance using a multistage approach. *Expert Systems*, *21*(5), 279–288.

Yeung, D. S., Ng, W. W., Chan, A. P., Chan, P. P., Firth, M., & Tsang, E. C. (2007). A multiple intelligent agent system for credit risk prediction via an optimization of localized generalization error with diversity. *Journal of Systems Science and Systems Engineering*, *16*(2), 166–180.

Yu, L., Wang, S., & Lai, K. K. (2005). A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates. *Computers & Operations Research*, *32*(10), 2523–2541.

Yu, L., Wang, S., & Lai, K. K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Systems with Applications*, *34*(2), 1434–1444.

Zhang, S., Xu, J., Zhang, Q.-J., & Root, D. E. (2016). Parallel matrix neural network training on cluster systems for dynamic FET modeling from large datasets. In *2016 IEEE MTT-S international microwave symposium (IMS)* (pp. 1–3). IEEE.

Zhong, H., Miao, C., Shen, Z., & Feng, Y. (2014). Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings. *Neurocomputing*, *128*, 285–295.

Zhou, S. M., & Da Xu, L. (2001). A new type of recurrent fuzzy neural network for modeling dynamic systems. *Knowledge-Based Systems*, *14*(5–6), 243–251.