Journal of Information Technology Research
Volume 15 • Issue 1

A Novel Ensemble Learning Model Combined XGBoost With Deep Neural Network for Credit Scoring

Xiaowei He, Northwest University of Information Science and Technology, China
Siqi Li, Northwest University of Information Science and Technology, China
Xin Tian He, Northwest University of Information Science and Technology, China
Wenqiang Wang, Northwest University of Information Science and Technology, China
Xiang Zhang, Northwest University of Information Science and Technology, China
Bin Wang, Northwest University of Information Science and Technology, China*
https://orcid.org/0000-0002-1589-8939

ABSTRACT

Credit scoring, aiming to distinguish potential loan defaulter, has played an important role in the financial industry. To further improve the accuracy and efficiency of classification, this paper develops an ensemble model combined extreme gradient boosting (XGBoost) and deep neural network (DNN). In the method, training set is divided into different subsets by bagging sampling at first. Then, each subset is trained as a feature extractor by DNN and the extracted features is taken as the input of XGBoost to construct the base classifier. At last, the prediction result is the average of outputs of different base classifiers. In the training verification process, three credit datasets from the UCI machine learning repository are used to evaluate the proposed model. The outcome shows that this model is superior with a significant improvement.

KEYWORDS

Credit Scoring, Deep Neural Network, Ensemble Learning, Extreme Gradient Boosting, Machine Learning

INTRODUCTION

Credit risk has always been one of the most important issues faced by financial institutions (Lai, Yu, Wang, & Zhou, 2006; Lai, Yu, Zhou, & Wang, 2006; Yu, Wang, & Lai, 2008). With the change of the concept of mass consumption and the development of the financial industry, the credit business has developed rapidly, and the financial institutions are facing more and more severe challenges. In this process, Credit scoring plays an important role. It can model the potential risks of loan applicants and classify them into "good credit" or "bad credit", which is a binary classification technology (He,

DOI: 10.4018/JITR.299924 *Corresponding Author

Zhang, & Zhang, 2018; Xia, Liu, Li, & Liu, 2017). For banks, financial institutions or other Internet finance companies, the cost of misclassifying "bad credit applicants" as "good credit applicants" is much higher than that of misclassifying "good credit applicants" as "bad credit applicants" (Qian, Liang, Li, Feng, & Shi, 2014). Therefore, how to build a robust and reliable credit scoring model is getting wider attention from both academia and business circles.

There are two mainstream classification techniques for credit risk assessment, namely statistical analysis and machine learning (He et al., 2018; Saberi et al., 2013). In statistical analysis, Linear discriminant analysis (LDA) and logistic regression (LR) are the two most commonly used approaches (Eisenbeis, 1978; Henley & Edward, 1995). However, both LDA and LR have difficulty in modeling complex financial systems due to the use of ideal statistical assumptions. Machine learning techniques are also widely used in credit scoring, including k-nearest neighbor (KNN) (W. E. Henley & Hand, 1996), support vector machine (SVM) (Huang, Chen, Hsu, Chen, & Wu, 2004), decision tree (DT) (Xiu, Weiyun, Jianyong, Bing, & Wenhuang, 2004), mathematical programming (Peng, Kou, Shi, & Chen, 2008; SHI, PENG, XU, & TANG, 2002), and Multi-layer perceptron (MLP) with a single hidden layer (Alejo, García, Marqués, Sánchez, & Antonio-Velázquez, 2013). Apart from single classifiers, researches have also shown that ensemble classification tends to be an effective way in improving the accuracy and stability of a single classifier for credit scoring (Ko, Sabourin, & Britto, 2008; Tsymbal, Pechenizkiy, & Cunningham, 2005).

Ensemble learning is a method that integrating several classifiers derived from different algorithms, features and training subsets to predict the class label of unknown samples. Ensemble classification can take advantage of the diversity of classifiers to avoid the weaknesses of single one. Moreover, it has been shown theoretically and experimentally that classification based on ensemble learning performs better than a single classifier in terms of credit scoring (Nanni & Lumini, 2009; Xia et al., 2017; Xiao, Xiao, & Wang, 2016). In recent years, deep neural networks (DNN) has also been widely applied in classification problems. Such deep architecture improves the ability of feature extraction and help get more information of hidden layers, and that's why its performance is better compared to shallow architectures in credit risk assessment. To the best of our knowledge, there were few studies on credit risk assessment by using DNN.

For that reason, this paper propose an ensemble classification approach which combines extreme gradient boosting (XGBoost) with DNN for credit scoring, as DNN is more capable of modeling or abstracting data with more hierarchies which makes it have the ability to mine the potentially valuable information of data and provide support for the classification of base classifiers. In the first step, we use bagging algorithm to form several variable training subsets to get enough data for training. Then DNN is applied to train the original data of each training subset to increase the ability of feature extraction, and the last hidden layer of this model is taken as feature extractor. The testing set and the training subset are respectively extracted, and the features extracted from the training subset are further trained by XGBoost. Finally, the final class label of the unknown sample is obtained by simply averaging the prediction probabilities of different base classifiers.

The rest of this paper is organized as follows. Section 2 reviews the related work in credit scoring. Section 3 describes and explains the proposed ensemble model in detail. Section 4 reports the experimental setup. Section 5 analyzes and compares the results of the experiments, and Section 6 provides the conclusion and directions for future work.

RELATED WORK

In recent years, credit scoring has received extensive attention from academia and business circles. Consequently, many scholars have done research in different aspects on credit scoring and related technology.

Credit Scoring Based on Single Classifier

Some scholars have compared neural networks, genetic algorithms and their extensions with traditional statistical approaches (Chang & Yeh, 2012; Crook, Edelman, & Thomas, 2007; Ravi Kumar & Ravi, 2007). D. Wang, Zhang, Bai, and Mao (2018) proposed a two-phase hybrid approach based on filter approach and multiple population genetic algorithm (HMPGA), which is effective in feature selection. Zhang, He, and Zhang (2018) presented a new method of selecting classifiers based on genetic algorithm. Unsupervised clustering is integrated with a fuzzy assignment procedure in the model, to make more use of the data patterns and improve performance. Oreski, Oreski, and Oreski (2012) proposed genetic algorithm combined with neural network to select the optimal feature subset, which improves the accuracy of credit scoring. The experiment found that the genetic algorithm is very advantageous in searching for the most significant features of default risk. Hsieh (2005) proposed a hybrid model based on k-means and neural network. It first pre-processed input samples through k-means and put unrepresentative samples into isolated clusters. Then the model used neural networks to construct a credit scoring model. Although the single classifier is relatively easy to implement and can achieve satisfactory results in simple scenarios, for complex scenarios, single classifier cannot capture the subtle differences between individuals.

Credit Scoring Based on Ensemble Classification

Because of the limitations of a single classifier, it is impossible to solve all problems effectively. Many scholars have proposed different ensemble approaches. There are many typical ensemble strategies in ensemble learning approaches, such as majority voting, weighted average and ranking (Yang & Browne, 2004). G. Wang and Ma (2012) proposed a new hybrid ensemble approach based on SVM and two ensemble strategies (bagging and random subspace). He et al. (2018) constructed a new three-stage ensemble model, generated adjustable balance subsets by extended the Balance Cascade approach. Random forest and extreme gradient boosting were used as the base classifiers of the three-stage ensemble model. The stacking was used for ensemble and the parameters of the base classifiers are optimized by a particle swarm optimization algorithm. The results showed that the average performance of the model is superior to other comparative algorithms. Xiao et al. (2016) investigated an ensemble classification approach based on supervised clustering for credit scoring to take both the accuracy of classification and the diversity of the classifiers into consideration. Yu et al. (2008) proposed a multistage reliability-based neural network ensemble learning approach for credit risk assessment, in which the neural network is used to fuse the final result by means of reliability measurement. Yu, Yue, Wang, and Lai (2010) also proposed an SVM -based ensemble learning system for credit risk, in which ANN model was introduced as the ensemble strategy.

Some scholars have also introduced the deep architectures training algorithms into the credit scoring. Hinton and Salakhutdinov (2006) first successfully introduced deep architectures training algorithm. The deep belief network (DBN) with sufficient hidden layers have been developed as a powerful ensemble technique to obtain the rich information in the confidence degree. Yu, Yang, Tang, and Journal (2016) proposed a novel multistage deep belief network based extreme learning machine ensemble learning paradigm for credit risk assessment, and DBN model as a new ensemble strategy shows great potential in improving accuracy. Yu, Zhou, Tang, and Chen (2018) further proposed a DBN-based resampling support vector machine (SVM) ensemble learning paradigm to solve imbalanced data problem in credit classification. Zhao et al. (2015) also presented an improved multi-layer perceptron neural network based on back propagation to improve the performance of credit scoring, and demonstrated that the performance of the model has been improved.

As can be seen from the above review, for single classifier credit scoring model, feature selection is the main concern, while for ensemble classification credit scoring model, the main focus is on ensemble strategy. The above approaches with DBN is also used as an ensemble technique to capture the information hidden in the results of the base classifiers. Unfortunately, little attention has been paid to the valuable information hidden in the original data. The method in this paper just makes up

for this shortcoming. We can use the deep neural network to increase the ability of feature extraction and extract all the valuable hidden information. Before training the base classifiers, we first use the deep neural network to train the original data, and then extracting the potential features train as the input of the base classifiers. Different subsets of data have different relations, so the information mined is more abundant, which is helpful to base classifier classification.

METHODOLOGY

In this section, we propose an ensemble classification model based on deep neural network for credit scoring. The original data is divided into training set and testing set. Bagging algorithm is used to generate variable training subset in the training set. The framework of the proposed model is shown in Figure 1. The process can be divided into two parts: (1) For each training subset, deep neural network (DNN) model is trained, and then the model with the last hidden layer is selected to extract the training subset and the testing set to obtain more valuable information. (2) The training set features obtained in the first step are trained by XGBoost to predict the extracted test set features. Finally, the predicted probability results of different base classifiers are averaged simply to obtain the final classification result.

Generating Training Subsets

To ensure sufficient data for model training, we use the bagging algorithm to generate different training subsets (Breiman, 1996). Given a training dataset , the size of the data is. We use bagging algorithm to retrieve training subsets with random sampling . The data size of each training subset is . The newly generated training subset is used for the next stage of DNN training. The number of training subsets *N* is optimized experimentally to obtain the optimal number.

Feature Engineering

Feature engineering is the most important part of machine learning, even deep learning. A good feature engineering can make the algorithm model work well. Shallow structure algorithms mainly focus on the output of the classifier at the abstract level (Luo, Wu, & Wu, 2017). The limitation of the algorithm

Figure 1. The flowchart of the proposed ensemble model

lies in its limited ability to represent complex functions with finite samples and computational units. For complex classification problems, its generalization ability is restricted to a certain extent. In this study, the deep neural network was used to extract the features of the original data.

Deep Neural Network (DNN) is an artificial neural network with more than three layers, sometimes referred to as deep MLP. Through the deep non-linear network structure, it can realize the approximation of complex functions, represent the distributed representation of input data, and demonstrate the powerful ability to learn the essential features of datasets from a small sample set (Hinton & Salakhutdinov, 2006). At present, the deep learning architecture has been widely used in various classification problems, and has produced excellent classification performance.

The neural network layers within DNN can be divided into three types: the input layer, the hidden layer and the output layer. A 3-layer fully connected DNN framework is shown in Figure 2. The layers are fully connected, that is, any neuron of the i -th layer must be connected to any one of the $(i+1)$ -th layer.

Each layer of DNN model is a linear relationship plus an activation function $\sigma(\mathrm{z})$. The linear relationship is as follows:

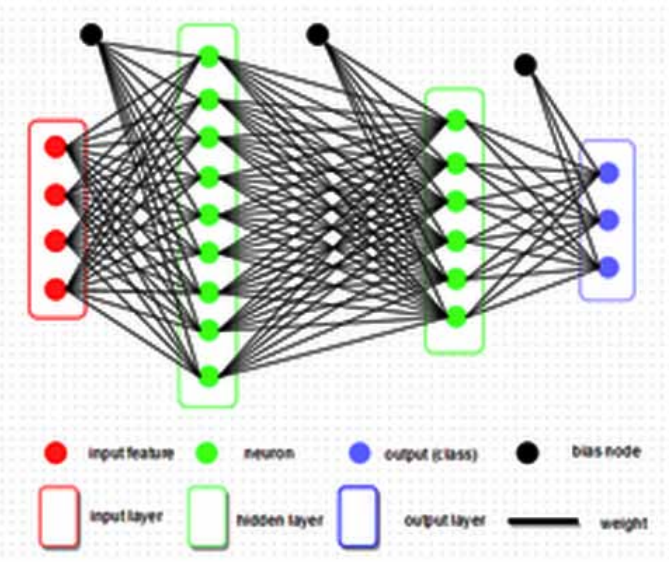
$$z = \sum \omega_i x_i + b \tag{1}$$

Where x_i represents the value of the i -th neuron, ω_i is the weight of the i -th neuron, b is the bias. There are no ω and b parameters in the input layer. Here we use the ReLU activation function:

$$\sigma(\mathrm{z}) = \max(0, \mathrm{z}) \tag{2}$$

If it is greater than or equal to 0, it will remain unchanged, and if it is less than 0, it will be activated to 0. Assuming that there are neurons in the $(l-1)$ -th layer and n neurons in the l -th

Figure 2. A 3-layer fully connected DNN framework



layer, the linear coefficient ω of the l-th layer constitutes an $n \times m$ matrix W^l, and the bias b of the l-th layer constitutes an $n \times 1$ vector b^l. The $(l-1)$-th layer output a constitutes an $m \times 1$ vector a^{l-1}. Then the output of the l-th layer is:

$$a^l = \sigma\left(z^l\right) = \sigma\left(\mathrm{W}^l \mathrm{a}^{l-1} + \mathrm{b}^l\right) \tag{3}$$

The categorical cross-entropy loss function is used to perform iterative optimization by gradient descent method to obtain the minimum value. After a certain number of iterations, the output of the last hidden layer is obtained, which is the extracted hidden feature. These features are further used as the input of XGBoost to construct base classifier.

Constructing Base Classifiers

The boosting method is a very effective machine learning method. Its basic idea is to combine a series of weak classifiers to form a strong classifier. Boosting tree is a boosting method based on decision tree. The learning model is optimized by addition model and forward distribution algorithm. When the loss function is a square loss function or an exponential loss function, the boosting tree is very effective. But for general loss function, it is not so easy to optimize each step. Therefore, Friedman (2001) proposed a gradient boosting algorithm. Its characteristic is that it constructs a new model in the gradient direction of residuals to minimize the loss function produced by each iteration. XGBoost (Chen & Guestrin, 2016) is an improvement based on the gradient boosting algorithm by Chen Tianqi. It not only has the advantage of high precision of traditional boosting algorithms, but also can flexibly implement distributed and parallel computing (Gumus & Kiran, 2017). In various international machine learning contests (Adam-Bourdarios et al., 2015), XGBoost is one of the algorithms adopted by many winners. Therefore, in this paper, XGBoost is used as the base classifier for the ensemble model. (4)

Given a training set feature vector $x_i \in \Re^n$, the corresponding class label is $y_i \in \{-1, +1\}, i = 1, \cdots, n$. The prediction model of XGBoost can be expressed as:

$$\hat{y}_i = F\left(x_i\right) = \sum_{k=1}^{K} f_k\left(x_i\right) \tag{4}$$

Where $f_k\left(x_i\right)$ represents the k-th tree, K is the total number of trees, $\hat{y}_i$ is the prediction result of the sample x_i.

The function f_k is learned by minimizing the following objective functions:

$$Obj\left(\theta\right) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right) \tag{5}$$

Where $l\left(y_i, \hat{y}_i\right)$ is the training error of the sample x_i, $\Omega\left(f_k\right)$ represent the regular term of the k-th tree. For the regular term part of the objective function, we consider it from a single tree. For each regression tree, the model can be written as:

$$f_t(x) = w_{q(x)}, w \in R^T, q: R^d \to \{1, 2, \cdots, T\} \tag{6}$$

Where w is the score of the leaf node, $q(x)$ is the leaf node corresponding to the sample x. T is the number of leaf nodes of the tree. Therefore, we write the complexity of the tree as follows:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{7}$$

Where γ is the complexity parameter, and λ is the fixed coefficient.

XGBoost uses Taylor expansion to approximate the original objective function, then the original objective function can be approximated as equation (8):

$$Obj^{(t)} = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j + \lambda) w_j^2 \right] + \gamma T \tag{8}$$

The objective function is rewritten as T independent quadratic functions of single variables. Therefore, the optimal score w_j^* for each leaf node in XGBoost is equation (9) and the solution equation for the objective function is equation (10):

$$w_j^* = -\frac{G_j}{H_j + \lambda} \tag{9}$$

$$Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T \tag{10}$$

is only related to the structure $q(x)$ of the tree, and is independent of the score of the leaf node. Therefore, the corresponding objective function can be calculated as long as the structure of the tree is determined.

XGBoost uses exact greedy algorithm and defines gain formula to find the optimal tree structure heuristically. If the current tree structure I can be split into I_L and I_R, $I = I_L \cup I_R$, the gain formula can be expressed as equation (11):

$$Gain = -\frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \tag{11}$$

Where γ represents the complexity cost of introducing additional leaf nodes. The exact greedy algorithm is shown in Algorithm 1.

EXPERIMENTAL SETUP

In this study, ten classifiers DT, LR, NB, SVM, RF, GBDT, LDA, KNN, Adaboost, and XGBoost, which are widely used in credit scoring, are used as base classifiers. Besides XGBoost, each base classifier is combined with DNN to validate the performance of the proposed model.

Dataset Description

In this experiment, three credit datasets from Australian, German and Japanese in the UCI machine learning repository (Dua, 2017) are used to validate the proposed model. They are widely used in credit scoring research. The detailed description of the datasets is shown in Table 2.

The Australian dataset contains 690 samples with 307 positive samples and 383 negative samples. Each sample has 14 feature factors and 1 class label. 6 of the 14 feature factors are numerical features and the other 8 are categorical features. Similarly, there are 690 samples in the Japanese dataset, including 383 positive samples and 307 negative samples. Each sample has 15 feature factors and 1 class label. Among the 15 feature factors, there are 6 numerical features and 9 categorical features. The German dataset consists of 1000 samples, 700 of which belong to positive samples and the remaining 300 belong to negative samples. Each sample has 20 feature factors, including 7 numerical features and 13 categorical features, and 1 class label.

The class labels of the above three datasets are "1" or "0". "1" indicates good applicants, that is, the applicant has good credit. On the contrary, "0" means bad applicants, that is, the applicant has bad credit.

Data Preprocessing

Before building the model, feature engineering is applied to preprocess the dataset. For missing data, if the missing value of an attribute is more than 2% of the total number of samples, we use the mean value to fill in, otherwise we use 0 to fill in. If the attribute is a categorical attribute, it is populated as a new category. For categorical features, dummy variables are used instead.

Table 1. Algorithm 1: Exact greedy algorithm for split finding

Algorithm 1: Exact Greedy Algorithm for Split Finding

Input: I **, instance set of current node**
Input: d **, feature dimension**
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i, \mathrm{H} \leftarrow \sum_{i \in I} h_i$
for $k = 1$ **to** m **do**
$G_L \leftarrow 0, H_L \leftarrow 0$
for j **in sorted(**I **,by** x_{jk} **) do**
$G_L \leftarrow G_L + g_i, H_L \leftarrow H_L + h_j$
$G_R \leftarrow G - G_L, H_R \leftarrow H - H_L$
$$score \leftarrow \max\left(score, \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G^2}{H + \lambda}\right)$$
end for
end for
Output:Split with max score

Table 2. Description of three credit datasets

Dataset	Number of instances	Good/bad credit	Total features	Numerical features	Categorical features
Australian	690	307/383	14	6	8
German	1000	700/300	20	7	13
Japanese	690	383/307	15	6	9

Evaluation Measures

In order to validate the performance of the model, seven commonly used evaluation measures are adopted in this experiment, namely, Area Under ROC Curve (AUC), Accuracy, Precision, Recall, F-score, Type I error and Type II error. They are all based on the confusion matrix shown in Table 3. Credit scoring is a two-class problem. Each sample can be classified into two classes: good credit and bad credit. There are four basic elements in the confusion matrix: True Positive (TP) indicates that the prediction result of the sample is good credit, and the real one is good credit; False Negative (FN) indicates that the prediction result of the sample is bad credit, but its real class is good credit. Similarly, False Positive (FP) is a bad credit sample, but good credit is predicted; True Negative (TN) is the same type of prediction as the real one, and all of them are bad credit.

- Accuracy: It is defined as the correct prediction sample size divided by the total testing sample size, as shown in formula (12), which reflects the overall prediction accuracy of the dataset.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{12}$$

- Precision: Precision denotes the accuracy of sample prediction for good credit, as shown in Formula (13).

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

- Recall: The recall rate, also known as sensitivity, describes the sensitivity of the classifier to positive samples, as shown in formula (14).

$$Recall = TPR = \frac{TP}{TP + FN} \tag{14}$$

Table 3. Confusion matrix

		Predicted	
		Positive	**Negative**
Real	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

- F-score: It is a comprehensive evaluation measure based on accuracy and recall rate, and its expression definition is shown in formula (15).

$$F_{score} = \frac{2 * Recall * Precision}{Recall + Precision} \tag{15}$$

- AUC: It is an extensively used evaluation measure, which is calculated by probability. It is the area under the ROC (Receiver Operating Characteristic) curve (Fawcett, 2004). The x-axis of ROC curve indicated false positive rate (FPR), and the y-axis indicated true positive rate (TPR).
- Type I error: It refers to the rate of misclassifying bad credit applicants into good credit, which is defined as formula (16). The cost of such errors is often higher. Therefore, the lower the value, the lower the cost.

$$error\ \mathrm{I} = FPR = \frac{FP}{FP + TN} \tag{16}$$

- Type II error: It refers to the rate of misclassifying good credit applicants as bad credit, which is defined as formula (17). Such errors will also have an impact on financial institutions, which loss out on a lot of profits. Therefore, the lower the value, the lower the cost. But compared with Type I error, the cost is lower.

$$error\ \mathrm{II} = \frac{FN}{TP + FN} \tag{17}$$

Experiment Parameters Setting

According to the distribution of datasets, 20% of good/bad applicants are used as testing sets respectively, and 90% of the remaining 80% are extracted by bagging algorithm to form several training subsets. After parameter tuning, the Australian and German datasets use 10 training subsets and the Japanese dataset have 15 training subsets. In each experiment, the deep neural network model is constructed by modifying the parameters of DNN, such as the number of hidden layers, the number of neurons in each hidden layer, the number of iterations, and the activation function. The loss function of the output layer is categorical cross-entropy. After parameter tuning, Table 4 describes the necessary parameters for DNN model. Default parameters are used for DT, LR, NB, SVM, RF, GBDT, LDA, KNN and Adaboost. The learning rate of XGBoost is 0.03, the maximum depth of trees is 5, the sampling ratio of attributes is 0.8, and the number of iterations is 100.

EXPERIMENTAL RESULTS ANALYSIS

In this section, we compare the proposed model with the base classifiers of three credit datasets, compare the different base classifiers combined with the deep neural network, and analyze the influence of the layers of the deep neural network on the performance of the model. Finally, we compare the proposed model with other reference methods.

Table 4. The necessary parameters for DNN model

Model used	Parameter value
Activation function (Input)	ReLU
Activation function (Output)	Softmax
The loss function	Categorical cross-entropy
Optimizer	Rmsprop
Number of iterations	500
Batch_size	30

Benchmarking Results

Firstly, the performance of the ten base classifiers described in Section 4 on three credit data sets is evaluated, as shown in Table 5. AUC indicates the area under the ROC curve. ACC is the accuracy. PREC denotes the precision. And REC indicates the recall. The representations in all the tables below are the same, and the bold indicates the best performance. According to the results of Table 4, XGBoost, SVM, GBDT, LDA and LR can show better performance on three datasets, while the performance of DT and NB is relatively poor. The type I error of SVM is relatively higher in the three datasets up to 0.71 in the German dataset. This is not a good situation for credit scoring. It means that more bad credit applicants will be predicted for good credit, which will cause huge losses to financial institutions. In contrast, the performance of the proposed model is significantly better than that of the base classifier. Compared with the single performance of the best base classifier, the AUC of Australian, German and Japanese increased by 2.7%, 8.6% and 1.6%, respectively, while the Type I error decreased by 23.6%, 9.5%, and 54.2%, respectively.

Performance Comparisons of Different Base Classifiers Combined With DNN

In order to validate the effect of our model and the effect of combining the deep neural network, we compare the ten base classifiers described in Section 4 with DNN respectively. The experimental results of combining different base classifiers with DNN on three datasets are shown in Table 6. The experimental results show that the performance of the model combined with DNN is significantly improved compared with the previous single classifier. On the Australian dataset, the AUC of DNN+KNN model is as high as 0.9601, and the accuracy is about 4% higher than before. Type I error and type II error of DNN+NB model also decreased. In contrast, the proposed DNN+XGBoost model performs better, with 0.9653 AUC reaching the highest value and 0.9343 accuracy being the only one that achieves more than 90%. And the type I error and type II error are the smallest in all the models.

In the German dataset, the accuracy of Adaboost and XGBoost models combined with DNN can reach more than 80%. The DNN+LR model has the lowest type II error. The proposed model performs best in all experiments, with 0.8858 F score and 0.8564 AUC.

In the Japanese dataset, the accuracy of DNN+KNN model and DNN+Adaboost model has not been significantly improved, but compared with the previous base classifier, it has not decreased. This may be due to the missing data in Japanese dataset,

which does not improve the performance of the model in processing missing data. The AUC and accuracy of the proposed model are relatively higher. The type I error is as low as 3.95%, but the type II error is relatively higher. Compared with type II error, the lower type I error has less loss for financial institutions.

Table 5. Results of three datasets in different base classifiers

Dataset	Models	AUC	ACC	PREC	REC	Type I error	Type II error	F-score
Australian	DT	0.8183	0.8205	0.8067	0.7982	0.1615	0.2018	0.7995
	LR	0.9268	0.8581	0.8112	0.9024	0.1771	0.0976	0.8514
	NB	0.8988	0.7896	0.8528	0.6341	0.0861	0.3659	0.7228
	SVM	0.9288	0.8552	0.7890	0.9248	0.2007	0.0752	0.8507
	RF	0.9221	0.8549	0.8634	0.8008	0.1018	0.1992	0.8296
	LDA	0.9342	0.8610	0.7988	0.9219	0.1879	0.0781	0.8554
	KNN	0.9141	0.8492	0.8436	0.8180	0.1252	0.1820	0.8270
	GBDT	0.9250	0.8582	0.8391	0.8472	0.1332	0.1528	0.8414
	Adaboost	0.9339	0.8547	0.7915	0.9247	0.2017	0.0753	0.8514
	XGBoost	0.9393	0.8667	0.8465	0.8602	0.1281	0.1398	0.8519
	DNN-XGBoost	**0.9653**	**0.9343**	**0.9194**	**0.9344**	**0.0658**	**0.0656**	**0.9268**
German	DT	0.6329	0.6860	0.7810	0.7657	0.5000	0.2343	0.7725
	LR	0.7808	0.7560	0.7884	0.8914	0.5600	0.1086	0.8363
	NB	0.7615	0.7300	0.8252	0.7800	0.3867	0.2200	0.8009
	SVM	0.7885	0.7440	0.7547	**0.9400**	0.7133	**0.0600**	0.8372
	RF	0.7569	0.7320	0.7953	0.8329	0.5033	0.1671	0.8126
	LDA	0.7811	0.7590	0.7974	0.8814	0.5267	0.1186	0.8366
	KNN	0.7482	0.7430	0.7903	0.8629	0.5367	0.1371	0.8246
	GBDT	0.7853	0.7760	0.8131	0.8857	0.4800	0.1142	0.8468
	Adaboost	0.7410	0.7170	0.7410	0.9171	0.7500	0.0829	0.8193
	XGBoost	0.7872	0.7500	0.7732	0.9114	0.6267	0.0886	0.8364
	DNN-XGBoost	**0.8564**	**0.8350**	**0.8591**	0.9143	**0.3500**	0.0857	**0.8858**
Japanese	DT	0.8026	0.8056	0.7866	0.7749	0.1698	0.2251	0.7796
	LR	0.9166	0.8568	0.8054	0.8961	0.1748	0.1039	0.8480
	NB	0.8827	0.7884	0.8632	0.6314	0.0863	0.3686	0.7228
	SVM	0.9267	0.8553	0.7920	0.9250	0.2007	0.0749	0.8514
	RF	0.9153	0.8579	0.8573	0.8173	0.1096	0.1827	0.8358
	LDA	0.9216	0.8538	0.7861	0.9255	0.2037	0.0745	0.8490
	KNN	0.9101	0.8609	0.8387	0.8539	0.1334	0.1461	0.8441
	GBDT	0.9405	0.8696	0.8522	0.8602	0.1228	0.1398	0.8545
	Adaboost	0.9317	0.8536	0.7851	**0.9282**	0.2064	**0.0718**	0.8495
	XGBoost	0.9415	0.8681	0.8387	0.8729	0.1359	0.1271	0.8548
	DNN-XGBoost	**0.9566**	**0.8978**	**0.9434**	0.8197	**0.0395**	0.1803	**0.8772**

The Influence of the Hidden Layer Of DNN on the Model

We analyze the influence of the number of hidden layers on the performance of the proposed model. Figure 3 illustrates the influence of the number of different DNN hidden layers on the performance

of the model in three datasets respectively. For the Australian dataset, when the number of hidden layers is 3 or 6, the performance of the model is better. However, the performance of the number of the hidden layer below 3 decreases obviously. The rates of type I and type II error with 3 hidden layers are 0.0658 and 0.0656, respectively, the AUC is 0.9653 and the accuracy is 0.9343. When the number of hidden layers is 6, the model achieves the AUC with 0.9592, the accuracy with 0.9270, the F score with 0.9167, 0.0526 type I error and 0.0984 type II error. But more hidden layers mean more time costs. Similarly, on German and Japanese datasets, the performance of hiding layers with 3 is better than that of higher layers, and type I error is the lowest. Except for the obvious difference of type I error, the fluctuation range of other metrics is small. In conclusion, when the number of hidden layers is 3, the performance of the proposed model performs well.

(a) Australian
(b) German
(c) Japanese

Comparison With Other Reference Methods

Table 7 shows the comparison of the proposed DNN-XGBoost model with other reference methods, i.e. EBCA-three-stage ensemble model by He et al. (2018), heterogeneous ensemble model based on Bstacking by Xia, Liu, Da, and Xie (2018), CF-GA-Ens by Zhang et al. (2018) and XGBoost-TPE model by Xia et al. (2017). The results show that the proposed model in this paper outperforms other models by using the same datasets and the same performance measures.

CONCLUSION

In this study, we present an ensemble model with a combination of XGBoost and deep neural network for credit scoring. As the following steps show, we previously separate the original data into training set and testing set, and use bagging sampling method to break the training set into several subsets, then extract features by performing DNN and take them as the input of XGBoost to form base classifiers. Based on the constructed classifiers, the label prediction of testing set can be acquired by calculating the probability average of its output. Comparative experiments acted on the three UCI

Figure 3. Influence of the number of hidden layers on the model

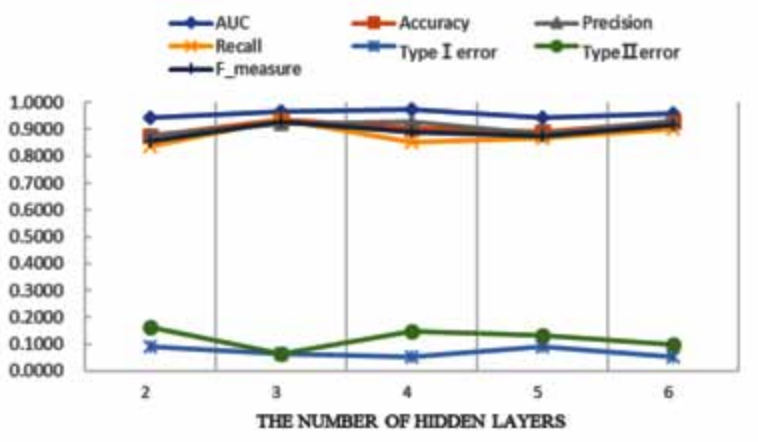


Table 6. Different base classifiers combined with DNN in three datasets

Dataset	Models	AUC	ACC	PREC	REC	Type I error	Type II error	F-score
Australian	DNN-DT	0.9204	0.8686	0.8772	0.8197	0.0921	0.1803	0.8475
	DNN-LR	0.9340	0.8686	0.8525	0.8525	0.1184	0.1475	0.8525
	DNN-NB	0.9116	0.8394	0.898	0.7213	**0.0658**	0.2787	0.8000
	DNN-SVM	0.9299	0.8613	0.8182	0.8852	0.1579	0.1148	0.8504
	DNN-RF	0.9461	0.8905	0.8594	0.9016	0.1184	0.0984	0.8800
	DNN-LDA	0.9474	0.8759	0.8929	0.8197	0.0789	0.1803	0.8547
	DNN-KNN	0.9601	0.8832	0.8358	0.9180	0.1447	0.0820	0.8750
	DNN-GBDT	0.9290	0.8759	0.8793	0.8361	0.0921	0.1639	0.8571
	DNN-Adaboost	0.9446	0.8686	0.8028	**0.9344**	0.1842	**0.0656**	0.8636
	DNN-XGBoost	**0.9653**	**0.9343**	**0.9194**	**0.9344**	**0.0658**	**0.0656**	**0.9268**
German	DNN-DT	0.7385	0.7350	0.7605	0.9071	0.6667	0.0929	0.8274
	DNN-LR	0.8043	0.7750	0.7746	**0.9571**	0.6500	**0.0429**	0.8562
	DNN-NB	0.7890	0.7850	0.7870	0.9500	0.6000	0.0500	0.8608
	DNN-SVM	0.8040	0.7500	0.7778	0.9000	0.6000	0.1000	0.8397
	DNN-RF	0.8136	0.7650	0.7853	0.9143	0.5833	0.0857	0.8449
	DNN-LDA	0.8289	0.7950	0.8037	0.9357	0.5333	0.0643	0.8647
	DNN-KNN	0.7899	0.7800	0.8038	0.9071	0.5167	0.0929	0.8523
	DNN-GBDT	0.8186	0.7900	0.8182	0.9000	0.4667	0.1000	0.8571
	DNN-Adaboost	0.7936	0.8050	0.8258	0.9143	0.4500	0.0857	0.8678
	DNN-XGBoost	**0.8564**	**0.8350**	**0.8591**	0.9143	**0.3500**	0.0857	**0.8858**
Japanese	DNN-DT	0.8852	0.8613	0.8281	**0.8689**	0.1447	**0.1311**	0.8480
	DNN-LR	0.9258	0.8613	0.8500	0.8361	0.1184	0.1639	0.8430
	DNN-NB	0.9044	0.8321	0.8800	0.7213	0.0789	0.2787	0.7928
	DNN-SVM	0.9394	0.8686	0.8772	0.8197	0.0921	0.1803	0.8475
	DNN-RF	0.9421	0.8905	0.8833	**0.8689**	0.0921	**0.1311**	0.8760
	DNN-LDA	0.9258	0.8686	0.8772	0.8197	0.0921	0.1803	0.8475
	DNN-KNN	0.9185	0.8759	0.8548	**0.8689**	0.1184	**0.1311**	0.8618
	DNN-GBDT	0.9474	0.8905	0.8966	0.8525	0.0789	0.1475	0.8739
	DNN-Adaboost	0.9197	0.8686	0.8525	0.8525	0.1184	0.1475	0.8525
	DNN-XGBoost	**0.9566**	**0.8978**	**0.9434**	0.8197	**0.0395**	0.1803	**0.8772**

datasets proved that the proposed method behaves better both in precision and efficiency. We also consider the influence of hidden layers of deep neural network on the performance of the model in this paper. For a dataset in small sample size, the experiments show that the performance of the model is the best when the number of hidden layers is 3.

Future work will focus on bring up an adaptive model for unbalanced datasets on the basis of this paper and determining the optimal number of ensemble models and the best choice of base classifiers. Studying different pre-processing techniques such as anomaly detection, more efficient missing value

Table 7. Comparison of the proposed model with other reference methods

Models	Australian		German		Japanese	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
EBCA-Three-stage model	/	0.9340	/	0.8002	/	0.9306
Bstacking Heterogeneous Ensemble Model	0.8629	0.9273	0.7832	0.7997	/	/
CF-GA-Ens	0.8761	0.9337	0.7725	0.8034	0.8746	0.9418
XGBoost-TPE	0.8792	/	0.7734	/	/	/
Proposed DNN-XGBoost Model	**0.9343**	**0.9653**	**0.8350**	**0.8564**	**0.8978**	**0.9566**

processing is also one of the future research directions. In addition, we will try to extend the model to deal with multi-classification problems.

FUNDING AGENCY

Acknowledgement

The publisher has waived the Open Access Processing fee for this article.

REFERENCES

Adam-Bourdarios, C., Cowan, G., Germain-Renaud, C., Guyon, I., Kégl, B., & Rousseau, D. (2015). The Higgs Machine Learning Challenge. *Journal of Physics: Conference Series*, *664*(7), 072015. doi:10.1088/1742-6596/664/7/072015

Alejo, R., García, V., Marqués, A. I., Sánchez, J. S., & Antonio-Velázquez, J. A. (2013). Making Accurate Credit Risk Predictions with Cost-Sensitive MLP. *Neural Networks*.

Breiman, L. J. M. L. (1996). *Bagging predictors*. 10.1007/BF00058655

Chang, S.-Y., & Yeh, T.-Y. (2012). An artificial immune classifier for credit scoring analysis. *Applied Soft Computing*, *12*(2), 611–618. doi:10.1016/j.asoc.2011.11.002

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. doi:10.1145/2939672.2939785

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research*, *183*(3), 1447–1465. doi:10.1016/j.ejor.2006.09.100

Dua, D. G. (2017). *UCI Machine Learning Repository*. Available from University of California, Irvine, School of Information and Computer Sciences http://archive.ics.uci.edu/ml

Eisenbeis, R. A. (1978). Problems in applying discriminant analysis in credit scoring models. *Journal of Banking & Finance*, *2*(3), 205–219. doi:10.1016/0378-4266(78)90012-2

Fawcett, T. J. P. R. L. (2004). ROC Graphs. *Notes and Practical Considerations for Researchers.*, *31*(8), 1–38.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, *29*(5), 1189–1232. doi:10.1214/aos/1013203451

Gumus, M., & Kiran, M. S. (2017). *Crude oil price forecasting using XGBoost.* Paper presented at the 2017 International Conference on Computer Science and Engineering (UBMK). doi:10.1109/UBMK.2017.8093500

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. doi:10.1016/j.eswa.2018.01.012

Henley, W. E., & Hand, D. J. (1996). A k-Nearest-Neighbour Classifier for Assessing Consumer Credit Risk. *The Statistician*, *45*(1), 77–95. doi:10.2307/2348414

Henley. (1995). *Statistical aspects of credit scoring*. Academic Press.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Reducing the Dimensionality of Data with Neural Networks.*, *313*(5786), 504–507. doi:10.1126/science.1127647 PMID:16873662

Hsieh, N.-C. (2005). Hybrid mining approach in the design of credit scoring models. *Expert Systems with Applications*, *28*(4), 655–665. doi:10.1016/j.eswa.2004.12.022

Huang, Z., Chen, H., Hsu, C.-J., Chen, W.-H., & Wu, S. (2004). Credit rating analysis with support vector machines and neural networks: A market comparative study. *Decision Support Systems*, *37*(4), 543–558. doi:10.1016/S0167-9236(03)00086-1

Ko, A. H. R., Sabourin, R., & Britto, J. A. S. Jr. (2008). From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, *41*(5), 1718–1731. doi:10.1016/j.patcog.2007.10.015

Lai, K. K., Yu, L., Wang, S., & Zhou, L. (2006). *Credit Risk Analysis Using a Reliability-Based Neural Network Ensemble Model*. Academic Press.

Lai, K. K., Yu, L., Zhou, L., & Wang, S. (2006). *Credit Risk Evaluation with Least Square Support Vector Machine*. Academic Press.

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. doi:10.1016/j.engappai.2016.12.002

Nanni, L., & Lumini, A. (2009). An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, *36*(2, Part 2), 3028–3033. doi:10.1016/j.eswa.2008.01.018

Oreski, S., Oreski, D., & Oreski, G. (2012). Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. *Expert Systems with Applications*, *39*(16), 12605–12617. doi:10.1016/j.eswa.2012.05.023

Peng, Y., Kou, G., Shi, Y., & Chen, Z. (2008). A Multi-criteria Convex Quadratic Programming model for credit data analysis. *Decision Support Systems*, *44*(4), 1016–1030. doi:10.1016/j.dss.2007.12.001

Qian, Y., Liang, Y., Li, M., Feng, G., & Shi, X. (2014). A resampling ensemble algorithm for classification of imbalance problems. *Neurocomputing*, *143*, 57–67. doi:10.1016/j.neucom.2014.06.021

Ravi Kumar, P., & Ravi, V. (2007). Bankruptcy prediction in banks and firms via statistical and intelligent techniques – A review. *European Journal of Operational Research*, *180*(1), 1–28. doi:10.1016/j.ejor.2006.08.043

Saberi, M., Mirtalaie, M. S., Hussain, F. K., Azadeh, A., Hussain, O. K., & Ashjari, B. (2013). A granular computing-based approach to credit scoring modeling. *Neurocomputing*, *122*, 100–115. doi:10.1016/j.neucom.2013.05.020

Shi, Y., Peng, Y., Xu, W., & Tang, X. (2002). *Data mining via multiple criteria linear programming: applications in credit card portfolio management*. 10.1142/S0219622002000038

Tsymbal, A., Pechenizkiy, M., & Cunningham, P. (2005). Diversity in search strategies for ensemble feature selection. *Information Fusion*, *6*(1), 83–98. doi:10.1016/j.inffus.2004.04.003

Wang, D., Zhang, Z., Bai, R., & Mao, Y. (2018). A hybrid system with filter approach and multiple population genetic algorithm for feature selection in credit scoring. *Journal of Computational and Applied Mathematics*, *329*, 307–321. doi:10.1016/j.cam.2017.04.036

Wang, G., & Ma, J. (2012). A hybrid ensemble approach for enterprise credit risk assessment based on Support Vector Machine. *Expert Systems with Applications*, *39*(5), 5325–5331. doi:10.1016/j.eswa.2011.11.003

Xia, Y., Liu, C., Da, B., & Xie, F. (2018). A novel heterogeneous ensemble credit scoring model based on bstacking approach. *Expert Systems with Applications*, *93*, 182–199. doi:10.1016/j.eswa.2017.10.022

Xia, Y., Liu, C., Li, Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, *78*, 225–241. doi:10.1016/j.eswa.2017.02.017

Xiao, H., Xiao, Z., & Wang, Y. (2016). Ensemble classification based on supervised clustering for credit scoring. *Applied Soft Computing*, *43*, 73–86. doi:10.1016/j.asoc.2016.02.022

Xiu, L., Weiyun, Y., Jianyong, T., Bing, L., & Wenhuang, L. (2004). *Applications of classification trees to consumer credit scoring methods in commercial banks.* Paper presented at the 2004 IEEE International Conference on Systems, Man and Cybernetics.

Yang, S., & Browne, A. (2004). *Neural network ensembles: combining multiple models for enhanced performance using a multistage approach*. 10.1111/j.1468-0394.2004.00285.x

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. doi:10.1016/j.eswa.2007.01.009

Yu, L., Yang, Z., Tang, L. J. F. S., & Journal, M. (2016). *A novel multistage deep belief network based extreme learning machine ensemble learning paradigm for credit risk assessment*. 10.1007/s10696-015-9226-2

Yu, L., Yue, W., Wang, S., & Lai, K. K. (2010). Support vector machine based multiagent ensemble learning for credit risk evaluation. *Expert Systems with Applications*, *37*(2), 1351–1360. doi:10.1016/j.eswa.2009.06.083

Yu, L., Zhou, R., Tang, L., & Chen, R. (2018). A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data. *Applied Soft Computing*, *69*, 192–202. doi:10.1016/j.asoc.2018.04.049

Zhang, H., He, H., & Zhang, W. (2018). Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring. *Neurocomputing*, *316*, 210–221. doi:10.1016/j.neucom.2018.07.070

Zhao, Z., Xu, S., Kang, B. H., Kabir, M. M. J., Liu, Y., & Wasinger, R. (2015). Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Systems with Applications*, *42*(7), 3508–3516. doi:10.1016/j.eswa.2014.12.006

Xiaowei He received the M.S. degree from the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China, in 2005, and the Ph.D. degree in pattern recognition and intelligent system from the School of Life Sciences and Technology, Xidian University, Xi'an, in 2011. Since 2016, he has been a Professor with the School of Information Sciences and Technology, Northwest University, Xi'an. He is currently the Director of the Xi'an Key Laboratory of Radiomics and Intelligent Perception. His current research interests include medical imaging processing, 3D molecular imaging, and artificial intelligence.

Siqi Li was born in 1994. She received the bachelor's degree in Software Engineering from Northwest University in China in 2017. She is currently the master of Computer Applied Technology of Northwest University in China. Her main research interests are credit scoring and machine learning.

Xintian He was born in 1995. Bachelor of Electronic Information Engineering of Northwest University of school of Information Science & Technology. Master of Electronics and Communication Engineering of Northwest University of school of Information Science & Technology. Main research fields are credit scoring, machine learning.

Wenqiang Wang was born in 1993, bachelor's degree at the School of Computer Shenyang Aerospace University, Main research fields of network engineering and network optimization, he is studying for a master's degree in the School of Information Science and Technology of Northwest University. His major is computer technology. Main research field are Internet Finance, credit scoring, big data credit technology.

Zhang Xiang was born in 1994. He received the B.A. degree in English from school of foreign languages at Xidian University, Xi'an, China, in 2017. He is currently working toward the Master Degree in Software Engineering in Network and Data Center at Northwest University in China. His current research interests include the credit scoring, machine learning and machine translation for extended reading in different culture background.

Bin Wang, born in 1979, received his MS in circuits and systems from the Northwest University in China in 2004. He is a lecturer at the Northwest University, a member of the Chinese Computer Federation and a member of the Chinese Graphics Society. His major research interests are big-data analytics, fintech and business intelligence.