SAMPLE SELECTION AND NEURAL NETWORK RULE EXTRACTION FOR CREDIT SCORING

Rudy Setiono
School of Computing, National University of Singapore
Computing 1, Computing Drive Singapore 117590
Republic of Singapore
rudys@comp.nus.edu.sg

Abstract

We present an approach for sample selection using an ensemble of neural networks (NNs) for credit scoring. The ensemble determines samples that are outliers by checking the NN prediction accuracy on the original training data samples. Samples that are consistently misclassified by NNs in the ensemble are removed from the training data set. The remaining data samples are used to train another NN for rule extraction. Our experimental results show that by eliminating the outliers, NNs can be trained to achieve better predictive accuracy. The rule set extracted from one of these networks is more accurate than the rule set extracted from NNs trained with the original data.

Keywords: Neural networks, ensemble, sample selection, classification, credit scoring

1. INTRODUCTION

In credit scoring, improving the predictive accuracy of the model which has been built to differentiate between good and bad credits even by a small margin can translate into a huge financial gain. Various machine learning approaches such as decision trees, neural networks and support vector machines have been applied to obtain better predictive accuracy rates over those obtained by human experts or traditional statistical approaches. Instead of building a more sophisticated models for credit scoring, the proposed method presented in this paper focuses on how we can achieve improvement in accuracy of the models by selecting relevant training data samples.

In a supervised learning scheme, classification models such as neural networks are trained using historical data set where each data sample has been labeled as either good credit risk or bad credit risk. It is possible that some of these class labels have been incorrectly assigned.

There is also the possible presence of irregular data samples, these are samples having similar attribute values to the majority of samples in one class, but actually belong to the other class. The presence of mislabeled and/or irregular data samples in the training data set is likely to affect the performance of the neural networks. We attempt to remove them before building the eventual model that distinguishes between good and bad credit risks. An ensemble of neural networks are trained to identify potential mislabeled and irregular data samples. Training data samples that are consistently misclassified by the majority of neural networks in the ensemble will be removed. A neural network is then trained and pruned using the rest of the training data samples. Finally, a neural network rule extraction algorithm is applied to obtain a set of classification rules that explains the classification process of the neural network in a more comprehensible way to the user.

Our experimental results show that the accuracy of the networks trained using the selected data samples is higher than the accuracy of the networks that have been trained using the entire original training data samples. The rules extracted from one of these neural networks also achieve better predictive accuracy than the rules obtained from neural networks built using the original training data samples.

Neural network ensembles have been long known to improve the overall predictive accuracy over individual classifiers (Hansen & Salamon, 1990). In this work, the neural networks in the ensemble were trained on the same set of data samples and the classification of a data sample was obtained by voting. The superior performance of the ensemble was attributed to the ability of individual neural networks to search for good weights in the weight space having many local minima of the error function. Thus the ensemble enabled the neural networks to generalize differently, that is, to predict cross-validation or test data samples in different ways.

An application of neural network ensembles for time series forecasting shows improved accuracy from the ensembles over individual neural networks on a number of time series tested (Landassuri-Moreno & Bullinaria, 2009). The networks in the ensembles were trained using an evolutionary computation approach called the EPNet Algorithm by Landassuri-Moreno and Bullinaria. It was found that using the fittest half of the population in the ensemble and discarding the worst performing networks in the population was better than keeping all the networks. Further improvements in the predictive accuracy was also obtained by combining the outputs of the ensemble with a Rank-Based Linear Combination method instead of simply computing the average outputs.

Clustering was applied to group back-propagation neural networks into homogeneous clusters by Xiang and Yang (2001). The networks were trained prior to clustering using three ensemble strategies: bagging, Ada-boost and Random Space Method (RSM). The final ensemble was created by selecting neural networks having the highest classification accuracy from each cluster. Experimental results on three credit scoring data showed improved prediction accuracy when the networks were trained in conjunction with bagging and RSM.

An ensemble of Radial Basis Function (RBF) neural networks was proposed for the prediction of financial time series forecasting by Wang and Li (2010). The method for building and training the ensemble consists of four stages. In the first stage, data samples for neural network training were selected via bagging and boosting. In the second stage, candidates RBF neural networks were trained. The number of RBF neural networks to be included in the ensemble was determined in the third stage via Partial Least Square. Finally in the final stage, ν-Support Vector Machine for regression was used for predicting the time-series. Experiments on S&P500 and Nikkei225 time series showed that the proposed approach outperformed predictions from single RBF neural networks and other ensemble methods such as simple averaging.

West et al. (2005) presented the application of three strategies for forming neural network ensembles, namely cross-validation, bagging and boosting. These strategies were tested on three real world financial decision applications. The cross-validation ensemble strategy trained 100 neural networks using the training data samples and predicted the class labels of the test data samples by simple majority voting. The training data samples for the neural networks in the bagging ensemble were selected randomly with replacement from the available data. In the boosting ensemble, the samples for training were also selected randomly with replacement. However, samples that were misclassified by a neural network in the ensemble were given higher probability to be selected for training the subsequent neural networks. The neural network ensemble strategies of cross-validation and bagging produced better performance in terms of predictive accuracy than the single best model. The poor performance of the boosting strategy was attributed to the likely presence of noise, outliers and mislabeled training data samples in two of the three data sets.

Tsai and Wu (2008) compared the performance of individual neural network classifiers with the performance of ensembles of neural networks on three benchmark credit scoring data sets. The final classification of an ensemble was simply determined as the output which received the largest number of votes from the ensemble members. Only on one of the three data sets, the neural networks ensembles were found to be more accurate than the individual classifiers. When Type I and Type II errors were measured, there was no single approach that was superior to the rest. Type I error occurred when a good credit case is predicted as a bad one, and Type II error occurred when a bad credit case is predicted as a good one.

A new boosting algorithm called ET Boost was introduced by Finlay (2011). Error Trimmed Boosting iteratively removed well classified samples from the training data. At each iteration, a classifier was constructed, the prediction errors of all the training data samples were calculated and the data samples were sorted in decreasing order according to their prediction error values. For the next iteration, the training data set included only the top 97.5% of the sorted current data. A total of 50 classifiers were constructed, and the final classification was determined by simple majority voting. On two large credit scoring data sets, it was shown that ET Boost consistently outperformed other ensemble methods for classification. The classifiers that were tested included logistic regression, linear discriminant analysis, decision tree method CART, neural networks and k-Nearest Neighbor.

In this paper, we present how a neural network ensemble is applied for selection of training data samples with application to credit scoring. By training another neural network on the selected data samples, more accurate neural networks are obtained. A set of classification rules is then extracted from one of the trained networks. Such classification rules would provide a more comprehensible explanation to general users on how the samples are classified as bad credits or good credits compared to the decision made by the neural network. The outline of this paper as follows. In Section 2 we present our proposed approach to credit scoring using a neural network ensemble. We describe the following steps in our approach: (1) neural network ensemble creation, (2) sample selection with the neural network ensemble, (3) neural network training and pruning with the selected samples, and (4) rule extraction from a pruned network. The data set that we employed to test the effectiveness of the proposed approach is the German credit data set, which is publicly available (Asuncion & Newman, 2007). and has been used in many data mining and business analytics experiments. We report the results from our experiments in Section 3. Finally, we conclude the paper in Section 4.

2. CREDIT SCORING USING A NEURAL NETWORK ENSEMBLE

Our proposed approach to credit scoring using a neural network ensemble can be summarized in the following steps:

1. **Ensemble creation:** Using the available training data samples, train an ensemble of N feedforward neural networks.
2. **Sample selection:** Select training data samples based on the predictions provided by the ensemble.
3. **Model generation:** Train a neural network with the selected samples.
4. **Rule extraction:** Apply a neural network rule extraction algorithm to generate a comprehensible set of classification rules that distinguishes between good and bad credits.

We create an ensemble of neural networks by training a number of feedforward neural networks with all the available training data. As noted by Hansen and Salamon (Hansen & Salamon, 1990), this cross-validation strategy when applied in conjunction with neural networks provides better classification results due to the nature of the neural network training methods. Most methods for training neural networks such as the backprogation method, conjugate-gradient method (Battiti, 1992) or the quasi-Newton method (Dennis Jr. & Schnabel, 1983) are local optimization methods. When the neural networks have been initialized with different sets of random weight values, it is very likely that the network training will stop at different local minima. The predictions of these networks are therefore expected not to be identical and their collective decision to be more accurate than the individual network's predictions.

We employ the ensemble to identify outliers in the training data set. It has been shown that removing outliers and noise prior to learning improves the predictive accuracy of a large number of learning methods. Smith and Martinez (Smith & Martinez, 2011) employ a number of heuristics to identify outliers, such as k-Disagreeing Neighbors, Disjunct Size and Class Likelihood Difference. We label a data sample as an outlier if a sufficient number of neural networks in the ensemble incorrectly classify it. Those samples that are not identified as outliers form a new data set for training and pruning a new neural network that would be used as the final model for credit scoring. Finally, we apply a rule extraction algorithm Re-RX (Setiono et al, 2008; Setiono et al, 2009) to obtain a set of classification rules that allows the classification process of the network to be explained in a more comprehensible way. It also allows us to compare the rule set in terms of accuracy and complexity with the rule set that is generated by the same algorithm on the same training data set but with the outliers still included when training the neural network.

3. EXPERIMENTAL RESULTS

The German credit data set contains 1000 samples, each described by 7 continuous attributes and 13 discrete attributes. The number of good credit cases is 700, and bad credit cases 300. Prior to training the neural networks, we normalized the continuous attributes to values in $[0, 1]$, while the discrete attributes were recoded as binary attributes. As a result, there were a total of 63 inputs. The binary inputs were denoted as $D_1, D_2, \ldots D_{56}$, and the normalized continuous attributes $C_{57}, C_{58}, \ldots C_{63}$. We divided the data set into a training data set consisting of 666 randomly selected samples and a test data set consisting the remaining 334 samples (Baesens et al, 2003).

Each network was trained to minimize an error function that combined the cross-entropy error function and a penalty function of the network's weights. The penalty function was included in the error function to encourage weight decay (Hertz et al, 1991). Network connections with small weights will be removed by a network pruning algorithm. In addition to obtaining better generalization, pruning normally leaves only a small number of relevant network connections in the network that makes the rule extraction process simpler. The details of the error function as well as the pruning algorithm are described in our earlier paper (Setiono et al, 2011). Network connections were removed by pruning as long as the classification accuracy of the network was at least 78%. This threshold was selected based on our previous experiments on this data set (Setiono et al, 2011). The number of networks in the ensemble was set to 30.

A training data sample would be discarded if it was misclassified by at least 90% of the networks in the ensemble, that is, 27 of the 30 networks. Since the networks have been pruned to achieve a classification accuracy rate of 78%, relatively large number of training data samples were misclassified. In total, 135 out of the original 666 training data samples were removed because 27 or more pruned networks misclassified them.

We trained another set of 30 neural networks to check the average performance of the net-

	Accuracy rate		Area under ROC	
# Training samples	Training set	Test set	Training set	Test set
666	78.08 ± 0.14	76.58 ± 0.35	68.78 ± 0.35	66.23 ± 0.63
531	78.70 ± 0.17	77.83 ± 0.10	69.86 ± 0.60	69.34 ± 0.39

Table 1: The average accuracy of 30 neural networks trained with the full training set and selected samples.

works trained with the remaining 531 data samples. The average classification and prediction accuracy rates of 30 neural networks trained with all 666 training data samples, and of the 30 neural networks trained with 531 samples are shown in Table 1. We also show the average Areas Under the Receiver Operating Curve (AUC) of these networks on both the training and test data samples. The Receiver Operating Characteristic (ROC) curve is a plot of the true positive (tp) rate against the false positive (fp) rate obtained by a classifier. The two rates are computed as follows

$$\begin{aligned} fp &= \frac{\text{\# Class 0 samples classified as Class 1}}{\text{\# Class 0}} \\ tp &= \frac{\text{\# Class 1 samples classified as Class 1}}{\text{\# Class 1}} \end{aligned}$$

The accuracy rates and AUC had been computed by classifying each sample as follows: If the network output is at least equal to θ, then predict Class 1 (Good risk/Positive), otherwise predict Class 0 (Bad risk/Negative). The value of the threshold θ was selected for each network so as to maximize its classification accuracy on the training data samples. Given a fixed threshold θ as the cut-off value for classification, the true-positive and false-positive rates of the pruned network were calculated and the AUC_d of this classifier was computed as the area of the trapezoid with corner points $(0,0), (fp, tp), (1,1)$ and $(1,0)$:

$$AUC_d = \frac{1 - fp + tp}{2}$$

It has been suggested that for two-class real-world problems, the ROC curve provides a more meaningful performance measure than accuracy rate specially when the proportion of the two classes is not balanced (Yan et al, 2003).

The figures in Table 1 show that both the accuracy and AUC of the neural networks increase when the networks have been trained and pruned using only the selected samples on the training data set and the test data set. Student t-tests for mean comparison indicate that the results obtained from training the networks using only the selected samples are significantly different from the results from the networks trained with the original data samples at 95% confidence level.

One of the neural networks was selected for rule extraction using Re-RX (Figure 1). This network has only 9 inputs remained after pruning, these are shown in Table 2 along with

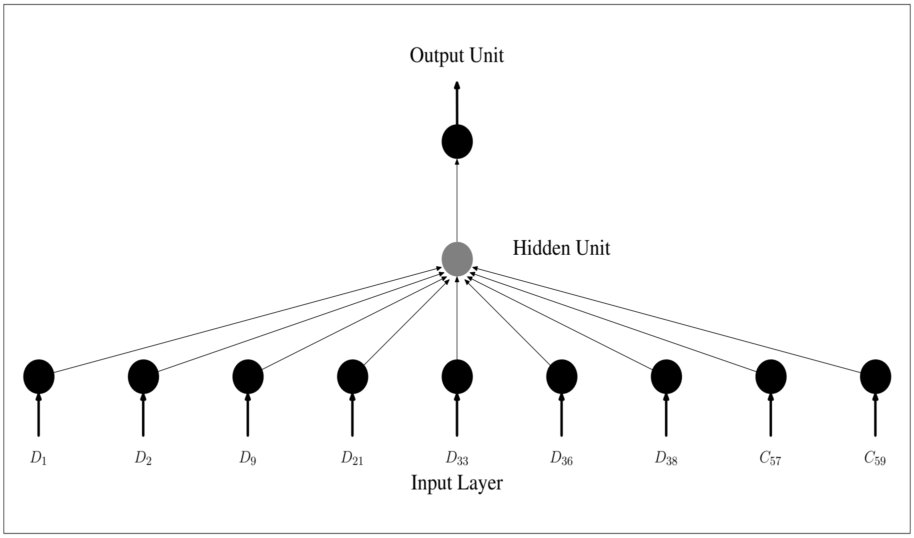

Figure 1: A feedforward neural network with nine input units and 1 hidden unit selected for rule extraction.

their corresponding original attribute information. The 9 network inputs correspond to 7 of the original 20 data attributes.

The extracted rules are shown below. The accuracy of these rules is shown and compared with the results from the widely-used decision tree method C4.5 (Quinlan, 1993) and four other rule extraction methods (Baesens et al, 2007) in Table 3. Note that all the accuracy rates in this table have been obtained from the same division of the data samples for training and testing each of the methods. The highest predicted accuracy is achieved by the rules extracted by Re-RX from a network trained with selected samples. These rules involve one fewer attributes (only 7 binary attributes and 2 continuous attributes) than the rules extracted by Re-RX from a network that had been trained using the complete training data set.

Rule set extracted from pruned neural network in Figure 1.

Rule $\mathcal{R}_1$: if $D_1 = 0$ and $D_2 = 0$, then predict Class 1,

Rule $\mathcal{R}_2$: else if $D_{38} = 1$, then predict Class 1,

Rule $\mathcal{R}_3$: else if $D_9 = 1$ and $D_{36} = 1$, then

Rule $\mathcal{R}_{3a}$: if $D_{21} = 0$, then

Rule $\mathcal{R}_{3a-i}$: if $C_{57} \geq 1.17$, then predict Class 0,

Rule $\mathcal{R}_{3a-ii}$: else predict Class 1,

Rule $\mathcal{R}_{3b}$: else

Input	Original attribute
$D_1 = 1$	iff Status of checking account less than 0 DM
$D_2 = 1$	iff Status of checking account between 0 DM and 200 DM
$D_9 = 1$	iff Credit history: critical account/other credits existing (not at this bank)
$D_{21} = 1$	iff Saving accounts/bonds: less than 100 DM
$D_{33} = 1$	iff Personal status and sex: male and single
$D_{36} = 1$	iff Other debtors/guarantors: none
$D_{38} = 1$	iff Other debtors/guarantors: guarantor
C_{57}	Duration in months
C_{59}	Installment rate in percentage of disposable income

Table 2: The relevant inputs for the German data set found by network pruning and their corresponding original attribute information.

Rule $\mathcal{R}_{3b-i}$**:** if $C_{57} \geq 0.35$, then predict Class 0,

Rule $\mathcal{R}_{3b-ii}$**:** else predict Class 1,

Rule $\mathcal{R}_4$**:** else if $D_1 = 1$ and $D_9 = 0$ and $D_{21} = 1$ and $D_{38} = 0$, then

Rule $\mathcal{R}_{4a}$**:** if $D_{33} = 0$, then

Rule $\mathcal{R}_{4a-i}$**:** if $C_{57} + 1.28\, C_{59} \geq 0.20$, then predict Class 0,

Rule $\mathcal{R}_{4a-ii}$**:** else predict Class 1,

Rule $\mathcal{R}_{4b}$**:** else

Rule $\mathcal{R}_{4b-i}$**:** if $C_{57} + 1.28\, C_{59} \geq 0.56$, then predict Class 0,

Rule $\mathcal{R}_{4b-ii}$**:** else predict Class 1,

Rule $\mathcal{R}_5$**:** else if $D_1 = 1$ and $D_{21} = 1$ and $D_{36} = 0$ and $D_{38} = 0$, then predict Class 0,

Rule $\mathcal{R}_6$**:** else if $D_2 = 1$ and $D_{36} = 0$ and $D_{38} = 0$, then predict Class 0,

Rule $\mathcal{R}_7$**:** else if $D_2 = 1$ and $D_9 = 0$ and $D_{21} = 1$ and $D_{38} = 0$, then

Rule $\mathcal{R}_{7a}$**:** if $D_{33} = 0$, then

Rule $\mathcal{R}_{7a-i}$**:** if $C_{57} + 0.19\, C_{59} \geq -0.01$, then predict Class 0,

Rule $\mathcal{R}_{7a-ii}$**:** else predict Class 1,

Rule $\mathcal{R}_{7b}$**:** else

Rule $\mathcal{R}_{7b-i}$**:** if $C_{57} + 0.19\, C_{59} \geq 0.62$, then predict Class 0,

Rule $\mathcal{R}_{7b-ii}$**:** else predict Class 1,

Rule $\mathcal{R}_8$**:** else predict Class 1.

Method	Accuracy (train)	Accuracy (test)
C4.5	80.63 %	71.56 %
C4.5rules	81.38 %	74.25 %
Neurorule	75.83 %	77.84 %
Trepan	75.37 %	73.95 %
Nefclass	73.57 %	73.65 %
Re-RX (all training samples)	77.93 %	78.74 %
Re-RX (selected training samples)	75.98 %	79.64 %

Table 3: Accuracy comparison of rules from decision tree method C4.5, various neural network rule extraction algorithms and the Re-RX algorithm using all training data samples and selected data samples for the German credit data set.

As the algorithm Re-RX generate hierarchical rules with the rules involving binary inputs appear higher in the hierarchy and those involving continuous inputs appear in the lowest hierarchy, the extracted rules could give better insight as to how the distinction between good and bad credit risks is made. The first two rules, $\mathcal{R}_1$ and $\mathcal{R}_2$ are straightforward, a sample is considered to be good credit risk if the status of checking account is more than 0 DM ($D_1 = 0$) and the status of checking account is greater than 200 DM ($D_2 = 0$) for the former, and if there is a guarantor ($D_{38} = 1$) for the latter. When the values of the continuous input attributes are required, the rule condition that involves only the discrete attribute is checked first. For example, $\mathcal{R}_3$ indicates that if credit history shows critical account or other credits existing not at this bank ($D_9 = 1$) and there is no other debtor/guarantor ($D_{26} = 1$) and saving account/bonds is more than 100 DM ($D_{21} = 0$), then the duration in months (C_{57}) is checked. If the (normalized) duration is at least 1.17, then the sample is predicted as bad credit risk. Otherwise, it is predicted as good credit risk. We believe that by checking the rule conditions involving the discrete and continuous attributes separately, it would prove easier to interpret and analyze the decision made by the pruned neural network from which these rules have been extracted.

4. CONCLUSION

Machine learning strategies involving ensembles of models have been demonstrated by many researchers to lead to more accurate classifiers. A more common approach to learning using an ensemble is to build a group of models and to make prediction by aggregating the outputs of the models in the ensemble. Diversity in the ensemble is introduced via varied training data samples, different learning tools, or different parameters used during learning.

In this paper, we have used a neural network ensemble to identify potential outliers in the data. Those samples incorrectly classified by a majority of the networks are considered to be potential outliers and they are removed from the training data set. When another group of neural networks are trained with the selected training data samples, we are able to obtain significant improvements in terms of accuracy and Area under the ROC curve.

By applying a rule extraction algorithm to generate a comprehensible set of rules from one of these networks, we obtain rules with better predictive accuracy compared to other rules from other rule generating methods such as C4.5, Neurorule, Trepan and Nefclass. Using the same rule extraction algorithm Re-RX, the rules generated from a pruned network that was trained with selected samples achieves higher predictive accuracy than the rules extracted by the same algorithm on a network trained with the original data set. The rule conditions of Re-RX rules separate the discrete attributes and the continuous attributes. The discrete attributes appear first in the rule conditions. When it is not possible to make a decision by just knowing these values, the relevant values of the continuous attributes are then checked. We believe this separation of the two groups of attributes in the rule conditions could enhance the interpretability of the neural network generated rules. Rule interpretability and high accuracy rates would make neural network rule extraction algorithms a viable alternative tool for a wider range of data analytics and business intelligence applications.

References

Asuncion, A and Newman, D.J. (2007). UCI Repository of machine learning databases. Irvine, CA: School of of Information and Computer Sciences, University of California. Available from http://www.ics.uci.edu/∼mlearn/MLRepository.html.

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

Battiti, R. First- and second-order methods for learning: Between steepest descent and Newton's method. *Neural Computation*, **4** (1992), 141–166

Hansen, L.K. and Salamon, P. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12(10)** (1990), 993–1001.

Dennis Jr. J.E. and Schnabel, R.E. Numerical methods for unconstrained optimization and nonlinear equations. 1983. Prentice Halls: Englewood Cliffs, NJ.

Finlay, S. Multiple classifier architectures and their application to credit assessment. *European Journal of Operational Research*, **210** (2011), 368–378.

Hertz J., Krogh A. and Palmer, R.G. Introduction to the theory of neural computation. 1991. Addison Wesley: Redwood City, CA.

Landassuri-Moreno, V. and Bullinaria, J.A. Neural network ensembles for time series forecasting. In *Proc. GECCO'09, Genetic and Evolutionary Computation Conference 2009*, 1235-1242.

Quinlan, R. C4.5: Programs for Machine Learning. 1993. Morgan Kaufmann: San Mateo, CA.

Setiono R., Baesens, B. and Mues, C. Recursive neural network rule extraction for data with mixed attributes. *IEEE Transactions on Neural Networks*, **19(2)** (2008), 299-307.

Setiono R., Baesens, B. and Mues, C. A note on knowledge discovery using neural networks and its application to credit screening. *European Journal of Operational Research*, **192(1)** (2009), 1009–1018.

Setiono, R., Baesens, B. and Mues, C. Rule extraction from minimal neural networks for credit card screening. *International Journal on Neural Systems*, **21(4)** (2011), 265–276.

Smith, M.R. and Martinez, T. Improving classification accuracy by identifying and removing instances that should be misclassified. In *Proc. of the 2011 International Joint Conference on Neural Networks*, 2690–2697.

Tsai, C-F. and Wu, J-W. Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications*, **34** (2008), 2639–2649.

Wang, D. and Li, Y. A novel nonlinear RBF neural network ensemble model for financial time series forecasting. In *Proc. 3rd International Workshop on Advanced Computational Intelligence 2010*, 86–90.

West, D., Dellana, S. and Qian, J. Neural network ensemble strategies for financial decision applications. *Computers & Operations Research*, **32** (2005), 2543–2559.

Xiang, H. and Yang, S.G. Credit scoring model based on selective neural network ensemble. In *Proc. 7th International Conference on Natural Computation 2011*, 513–516.

Yan L., Dodier R., Mozer M.C. and Wolniewicz, R. Optimizating classifier performance via the Wilcoxon-Mann-Whitney statistic, In *Proc. International Conference on Machine Learning 2003*, 848-855.