

Credit Risk Assessment Model Based Using Principal component Analysis And Artificial Neural Network

Abeer Hamdy^{1,2}, Walid B. Hussein¹

¹Faculty of Informatics and Computer science, The British University in Egypt, Cairo, Egypt

² Department of Computers and Systems, Electronics Research Institute, Cairo, Egypt

Abstract. Credit risk assessment for bank customers has gained increasing attention in recent years. Several models for credit scoring have been proposed in the literature for this purpose. The accuracy of the model is crucial for any financial institution's profitability. This paper provided a high accuracy credit scoring model that could be utilized with small and large datasets utilizing a principal component analysis (PCA) based breakdown to the significance of the attributes commonly used in the credit scoring models. The proposed credit scoring model applied PCA to acquire the main attributes of the credit scoring data then an ANN classifier to determine the credit worthiness of an individual applicant. The performance of the proposed model was compared to other models in terms of accuracy and training time. Results, based on German dataset showed that the proposed model is superior to others and computationally cheaper. Thus it can be a potential candidate for future credit scoring systems.

Keywords. Credit scoring, ANN, PCA, credit risk, German data.

1 Introduction

Credit risk assessment has gained increasing attention in recent years. Banks and financial institutions have extensively started to consider the credit risk of their customers in order to make decisions to grant credit to new applicants, extend credit or increase credit limit for existing customers and under what terms. If the customer starts to fall behind in his repayments what actions should the financial institution take? Techniques that help with such decisions are called credit and behavioral scoring models [1]. The information that is available in making a credit scoring decision includes both of the applicant's application form details and the information held by a credit reference agency on the applicant. Credit scoring models analyze the applicant's information and classify him into either applicant with good credit or applicant with bad credit. Applicant with good credit has great possibility to repay financial obligation, while, applicant with bad credit has high possibility of failing. The accuracy of credit scoring model is critical for financial institution's

profitability. A one percent improvement on the accuracy of credit scoring of applicants may protect the financial institution from a great loss. Parametric statistical methods particularly Linear Discriminate Analysis (LDA) and Logistic Regression (LR) have been utilized in developing credit scoring models [2-6]. The weakness of the LDA and LR is due to the assumption of linear relationship between response and predictor variables, which is usually nonlinear and the sensitivity to the deviation from the multivariate normality assumption. Recently, artificial intelligence (AI) and data mining approaches have been utilized in developing credit scoring models, and numerous studies revealed that AI based models outperform the classical statistical models. Neural networks is one of the AI techniques that have been applied to credit scoring due to their ability to model both of linear and non-linear functions [8-16]. However, ANNs are often criticized as being a "black box", computationally expensive and may suffer over fitting problem. Support vector machines SVMs, firstly developed by Cortes and Vapnik (1995), is another modern classification

Corresponding author: abeer.hamdy@bue.edu.eg

technique applied to credit scoring [17- 23]. SVMs do not suffer from over fitting problem like ANN and consequently can generalize well. However they are still computationally expensive. Genetic algorithms was also tried in the field of credit scoring [24].

According to the literature review, it is easy to find that almost all classification methods can be used for credit risk assessment and they demonstrate comparable performance. However, some hybrid AI techniques, which integrate two or more single classification methods or cluster and classification methods have shown higher predictability than individual methods. Recent examples are neural discriminate technique [25], Neurofuzzy [26,27], neural network ensemble [28,29], evolving neural network [30], fuzzy SVM, that integrates the theory of fuzzy sets with SVM [31] and with LSSVM [32] classifiers to increase their sensitivity to outliers and generalization capabilities. Harris [33] utilized clustered SVM (CSVM) in order to reduce the computational complexity required to train the non linear SVM using large datasets.

As AI techniques are computationally expensive relative to traditional statistical methods like LDA and LR, especially when it comes to classifying massive data sets like the ones used in developing credit scoring systems, it is known that many financial institutions and banks have millions of customers and they hold the information of their credit history and application forms.. The contribution of the paper is proposing a hybrid model to discriminate good from bad customers in customer credit evaluation. The model consists of an ANN classifier fed by attributes determined by a feature extraction technique called principal component analysis PCA. The hybrid technique accomplishes the following:

1. A credit scoring system that outperforms previous systems in terms of accuracy.
2. Reduction in the computational complexity of the ANN by reducing the dimension of the feature space using PCA.
3. Parametric study to the impact of the number of Principal components (PCs) on the model accuracy and training time
4. An analysis to the significance of each of the credit scoring attributes commonly used in building credit scoring models.

We compared our achievements to Harris [33] for the following reasons: 1. He utilized the open source data set we utilized in this paper, 2. His motivation is same as ours (improve predictive accuracy and reduce training time) and 3. He trained several other classifiers and outlined their performance.

The remainder of the paper is organized as follows: Section 2 presents the model, dataset and the performance criteria. Section 3 discusses our findings. Section 4 concludes the paper and outlines possible directions for future research.

2 Background

2.1 Artificial neural networks

Artificial neural network (ANN) is an information processing paradigm that is inspired by human brain nerve cells, their communication and techniques in processing information. ANN is composed of a large number of highly interconnected processing elements (neurons) working together to solve specific problem. ANN, like people, learn by examples; they generalize from previous experiences to new ones, and can make decisions [34]. The most common type of ANN consists of three layers of units: input layer, hidden layer, and output layer. It is called multilayer perceptron (MLP). A layer of “input” units is connected to a layer of “hidden” units, which is connected to a layer of “output” units. The input nodes represent the raw information that is fed into the network. The input of each hidden node is determined by the outputs of the input units and the weights on the connections between the input and the hidden units. The output layer consists of nodes that represent the system’s classification decisions. The outputs of the output nodes are determined based on the outputs of the hidden layer and the weights between the hidden and output layers. Figure 1 shows an example of three-layer neural network including input, output, and one hidden layer. The weight adjustments process of the ANN is known as training the ANN. The training process consists of running input values over the network with predefined classification output nodes. The predicted outputs for each training sample are calculated, and compared to the corresponding target outputs. The weights are then adjusted so that the error rate will be reduced next time when the training samples are presented to the network. Thus, the training algorithm finds out the properties of the ANN that are most relevant to learning the classification model. Testing samples are used to verify the performance of the trained network.

2.2 Principal component analysis (PCA)

Due to the multivariate nature of the samples, the computational time of designing and operating a classifier is challenging. Therefore, a PCA is applied to the dataset to find the best low dimension space that conveys maximum useful information, and to

define the underlying attributes that influence the samples.

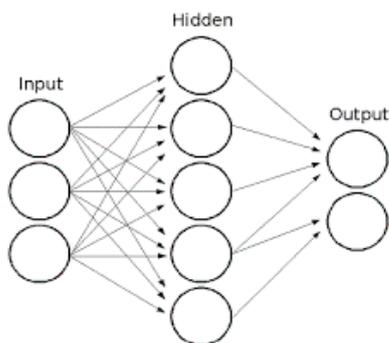


Figure 1. Three layers, two outputs an ANN architecture.

PCA is a mathematical procedure uses orthogonal decomposition and transformation to project a set of observations of possibly correlated attributes (i.e., high dimensional data) into a set of values of linearly uncorrelated attributes (i.e., low dimensional data) called principal components PCs. PCs are linear combinations of original attributes in which the weights allocated to the linear combinations of those original attributes are termed eigenvector. The number of PCs is less than the number of original attributes. The projection is performed in a least square sense, where the large (i.e. principal) variability in the data is captured, while the small variability is ignored. As a result, the first principal component (PC1) will be in the direction of maximum variance in the input space. The second principal component (PC2) is orthogonal to (PC1) and coincides the direction of the second maximum variance. Subsequent principal components are found according to the remaining maximum variance down to the last PC (i.e., PC24) which is the one represents the lowest variance of the data. PCA proved its effectiveness in various applications, for example [35-37] more and elaborated details about PCA algorithm are found in [38,39].

3 Methodology

3.1 Dataset

A German credit scoring dataset was taken from the UCI Machine Learning Repository. This dataset consists of 700 examples of creditworthy applicants and 300 examples of customers who should not have been granted credit. In addition, it presents twenty (20) categorical attributes for each credit applicant, which are:

- The status of the client’s existing checking account,
- The duration of the credit period in months,
- The client’s credit history,
- The purpose for the credit,
- The credit amount requested,
- The client’s savings account/bonds balance,
- The client’s present employment status,
- The client’s personal (marital) status and sex,
- Whether the client is a debtor or guarantor of credit granted by another institution,
- The number of years spent at present residence,
- The type of property possessed by client,
- The client’s age in years,
- Whether the client has other installment plans,
- The client’s housing arrangements (i.e. own their home, rent, or live for free)
- The number of existing credits the client has at the bank,
- The client’s job,
- The number of people for whom the client is liable to provide maintenance for,
- Whether the client has a telephone,
- Whether the client is a foreign worker.

3.2 Experimental Approach

Two types of models were trained for credit scoring. The first one of them is an ANN fed with the 24 attributes German data. The second type is a hybrid model consists of a PCA and ANN classifier. PCA is fed with the 24 attributes German dataset while ANN is fed with a set of PCs as shown in figure 2.

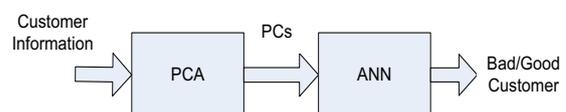


Figure 2. PCA-ANN model for credit scoring

For each attribute in the German dataset, the enclosed values were normalized in the range [-1,1] to prevent the ANN from being dominated by the input attributes with large values (in case of using ANN

only). If the attributes will be fed to PCA, normalization guarantees that PCs will be independent as PCA is sensitive to the relative scaling of the original attributes. Normalization is carried out by removing the feature's average and then dividing by its standard deviation using equation 1.

$$x' = \frac{x - \text{mean}}{\text{standard deviation}} \quad (1)$$

Where x and x' are the old and new value of each feature in the data set respectively.

Afterwards, the data were divided into 65% for training the model, 15% for validation, and 20% for testing the performance of the trained model.

3.3 Performance Metrics

A number of performance metrics could be used to report the performance of the credit scoring classifiers including predictive accuracy, confusion matrix and AUC metrics.

3.3.1 Predictive accuracy

Predictive accuracy metric measures how accurately the model classifies credit applicants on the dataset and is defined by equation 2.

$$\text{Predictive accuracy} = \frac{TP + TN}{P + N} \quad (2)$$

Where,

TP: true positive (the number of correct predictions that a customer is good);

TN: true negative (the number of correct predictions that a customer is bad);

P, N : are the number of positive and negative samples respectively.

Predictive accuracy is not a reliable metric for the real performance of the classifiers specially, in case of imbalanced data sets (this is the case of credit scoring data).

3.3.2 Confusion matrix

Confusion matrix is one of the methods that provides more detailed analysis than the accuracy metric to the results of the classifiers. Confusion matrix is explained in figure 3. Where, FP: false positive (the number of bad customers who are predicted as good customers); and FN: false negative (the number of good customers who are predicted as bad customers)

3.3.3 AUC

AUC, the Area under the Receiver Operating Characteristic (ROC) curve, metric has emerged as one of the most promising and widely accepted evaluation metric for binary classifiers in general and in the field of credit scoring [21]. AUC was first used by the American army after the attack on Pearl Harbour, to detect Japanese aircraft from radar signals. Its main advantages over other metrics is its insensitivity to unbalanced datasets. ROC curve is a two dimensional measure of classification performance where the sensitivity (the fraction of actual positives predicted as positive), as in equation 3 and the specificity (the fraction of actual negatives predicted as negative), as in equation 4 are plotted on the Y and X axis, respectively. The AUC measure is calculated as in equation 5, where, S1, represents the sum of the ranks of the creditworthy clients. Here, a score of 100% indicates that the classifier is able to perfectly discriminate between the classes, and a score of 50% indicates a classifier of insignificant discriminatory quality.

$$\text{Sensitivity} = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Specificity} = \frac{TN}{FP + TN} \quad (4)$$

$$\text{AUC} = \frac{S1 - \text{Sensitivity} * (\text{Sensitivity} + 1) * .5}{\text{Sensitivity} * \text{Specificity}} \quad (5)$$

		Actual classification	
		Positive (Good Customers)	Negative (Bad Customers)
Predicted Classification	Positive (Good Customers)	True Positive (TP)	False Positive (FP)
	Negative (Bad Customers)	False Negative (FN)	True Negative (TN)

Figure 3. Confusion matrix.

4 Results and Discussions

4.1 ANN model

The neural network was designed of three layers, input, hidden, and output layers with 24 neurons in the input layer for the 24 input credit scoring attributes. Fifteen neurons in the hidden layer with

Tansig activation function (equation 6) and one neuron in the output layer with linear transfer function to linearly scale the output values to the interval [0,1].

$$a = \frac{2}{1+e^{-2n}} - 1 \quad (6)$$

Where, n and a are the input and output of each neuron in the hidden layer; respectively.

The training algorithm is the back propagation algorithm [41]. Iterations were performed on the weights and bias of each neuron in the network until the mean squared error for the validation samples dropped to the possible minimum value. Figure 4 shows that the minimum mean square error was 0.14 at epoch number 2. At other epochs either the mean square error is high (epochs number 1 and 2). At epochs 4 and more, the mean square error of the model on the training data is decreased while it increases on the validation data which means that the generated models suffer over fitting problem. The performance of the best model (at epoch 2) was evaluated against the testing dataset. Figures 5 and table 6 show the ROC curve of the resulting classifier and its confusion matrix. It could be observed that the ROC curve of the trained classifier is above the diagonal line (line of no discrimination or random classifier) which implies that the ANN classifier can produce a good classification results.

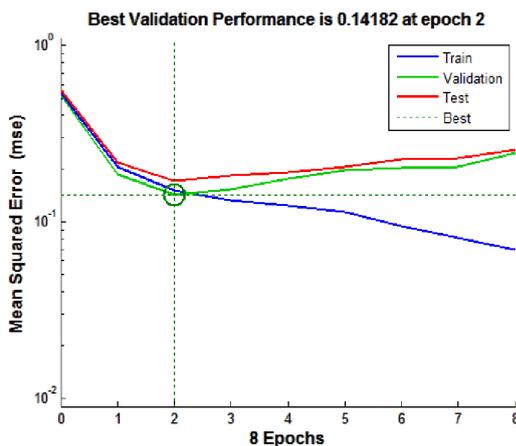


Figure 4. MSE of the trained ANN models at different epochs on each of the training, validation and testing datasets.

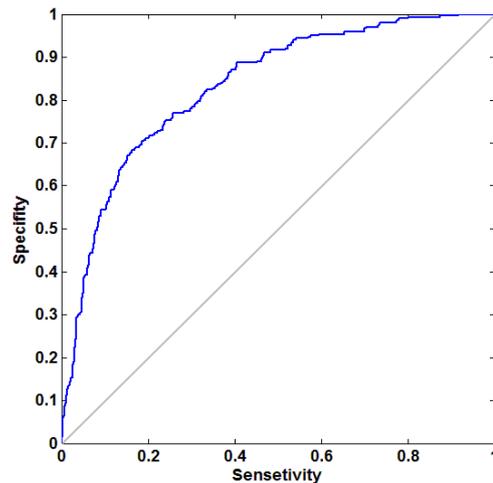


Figure 5. ROC curve of the trained ANN classifier

Table 1. Confusion matrix of the trained ANN classifier

		Actual classification	
		Positive (Good Customers)	Negative (Bad Customers)
Predicted Classification	Positive (Good Customers)	622 62.2%	104 10.4%
	Negative (Bad Customers)	78 7.8%	196 19.6%

4.2 PCA-ANN models

PCA was applied on the normalized data set. Figure 6 shows the significance of the resulting PCs with respect to data variation. The Y axis represents the normalized weight of each PC. It was found that the highest weight is equal to 861.75 and the lowest weight is equal to 0.015. It could be observed that the first three PCS account for as much of the variability in the data; their cumulative weights is equal to 1073.512.

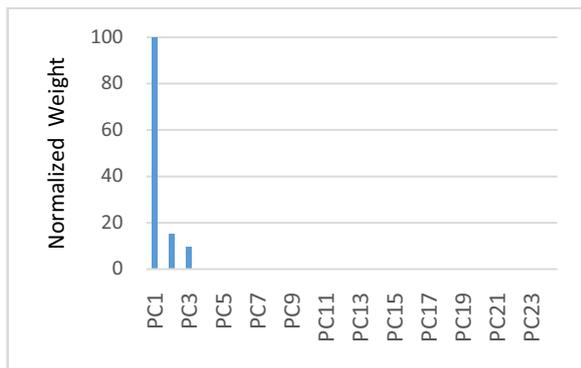


Figure 6. Significance of the principal components with respect to the data variation

Five PCA-ANN models were trained. The models differ from each other in the number of PCs that were fed to a three layer ANN. The structures of the five trained PCA-ANN models are listed in table 2. Figure 7 shows the AUC metric values for each of the ANN model and the five PCA-ANN models. It is clearly observed that the best model based on the AUC metric is the ANN model (AUC = .856). While the worst performance model is PCA-ANN#5 which uses 3 PCs only and results in AUC value equal to .625. PCA-ANN#4 which uses 6 PCS improves the performance dramatically (AUC = 0 .779), the more PCs included the closer the performance to the ANN model. Figure 8 shows the ROC curves of the models PCA-ANN#4 and PCA-ANN#5 and it is obvious the less the number of PCs included the closer the ROC curve to the diagonal which indicated less capability on diffrentitaing between good and bad clients.

Table 2. Description of PCA-ANN models.

Model	PCs number (Input neurons)	Hidden neurons
PCA-ANN #1	20	15
PCA-ANN# 2	10	5
PCA-ANN# 3	8	5
PCA-ANN# 4	6	2
PCA-ANN #5	3	2

Figure 9. Shows the training time (in seconds) of each model, as observed PCA-ANN#5 model is the most expensive model computationally as ANN was not able to generalize quickly with the few number of input features. However, all PCA-ANN models generalize faster than ANN only and the more PCs included the more training time required.

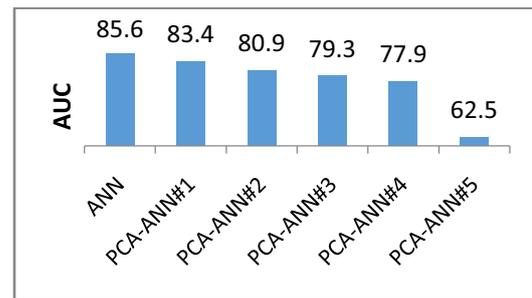


Figure 7. AUC metric values of the 6 trained models

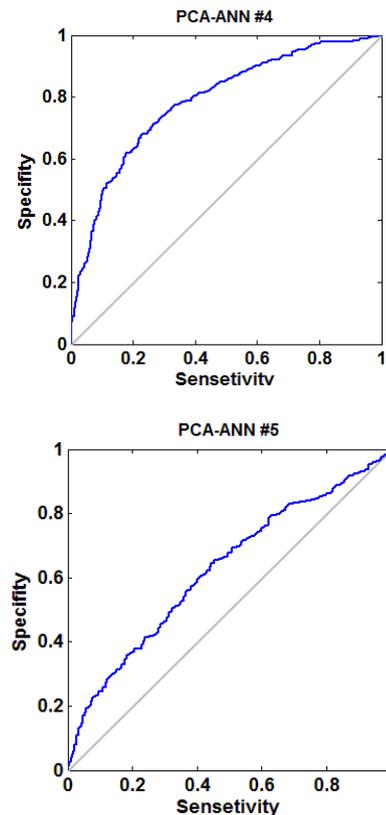


Figure 8. ROC curves for PCA-ANN#4 and PCA-ANN#5.

Figure 10 shows a comparison ,based on AUC, among the performance of our 6 trained models and 8 models of Harris [33]. The models of Harris are: logistic regression (LR), K-means plus logistic regression (K-means + LR), clustered support vector machine with a RBF kernel (CSVM-RBF), K-means plus support vector machine with a RBF kernel (K-means + SVM-RBF), support vector machine with a RBF kernel (SVM-RBF), linear clustered support vector machine (CSVM-linear), K-means plus support vector machine with a linear kernel (K-means + SVM-linear), and a linear support vector machine (SVMlinear). As observed all our models, except the model PCA-AN #5, outperform Harris models. We

could not compare the training time of our models to models in Harris [33] due to the difference between our platform and his.

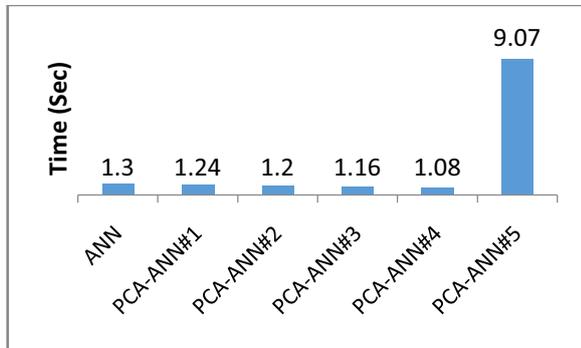


Figure 9. Training time (seconds) of the proposed models

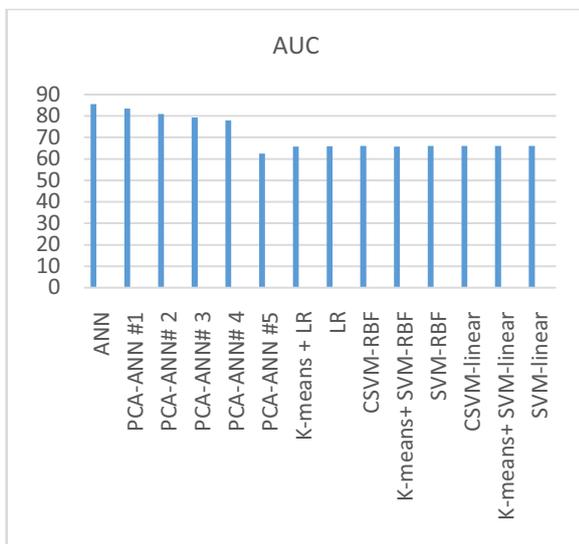


Figure 10. Performance of our trained credit scoring models compared to performance (in terms of AUC) of models provided in Harris [33].

4.3. Credit scoring attributes analysis

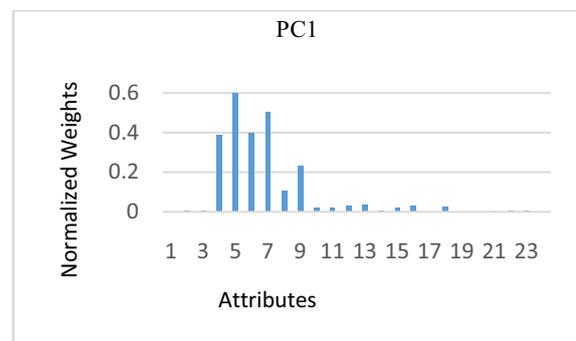
Although data mining techniques are useful in analyzing the credit worthiness, the selection of the best set of independent attributes to be included in the model building is essential to improve the performance of the model, speed up the learning process and save the storage space especially with the large amount of data. Same as PCA is a technique that can find a smaller size attribute space that represents a set of observations; it can also be helpful in analyzing an existing attribute space and determine the significance of each of these attributes. This significance is determined based on the dominant attributes in the dominant PCs, which are responsible for the most variability in the dataset. Figure 11

shows the significance, weights, of the different attributes in each of PC1, PC2 and PC3. It is obvious that some attributes contribute minimally to PC1, PC2 and PC3; consequently, their contribution to the credit-scoring model is minimal too and could be ignored.

It is noteworthy that when a NNET was trained using the attribute space of the German dataset (without pre-PCA), it gave a weight to each input attribute without any consideration to the actual impact of each attribute (as shown in figure 12), and this is in general a drawback of any ANN based model, where it develops a models that maps any nature of inputs to any nature of outputs. But with PCA analysis, significance of each input attribute is taken into consideration to develop a realistic model.

5 Conclusion

Credit scoring model is a mean to measure the risk related to a potential customer, by analyzing his data to determine the probability of him paying his debts back to the financial institution. The paper discussed the development of a credit-scoring model based on PCA and ANN to achieve better performance than existing models and reduce the computational complexity of the ANN through using PCA analysis. Numerous studies have proven that ANN perform remarkably better than any other statistical approach, such as logistic regression or discriminant analysis however they are computationally expensive. So a PCA The proposed model showed its ability to generalize well to the credit-scoring problem and its performance, in terms of AUC, outweighed many other previous models. Preceding ANN with PCA reduces the number of inputs to the ANN and reduces its training time. Moreover, PCA provides a mean for analyzing the importance of each attribute to the model which helps in reducing the input space size and save storage.



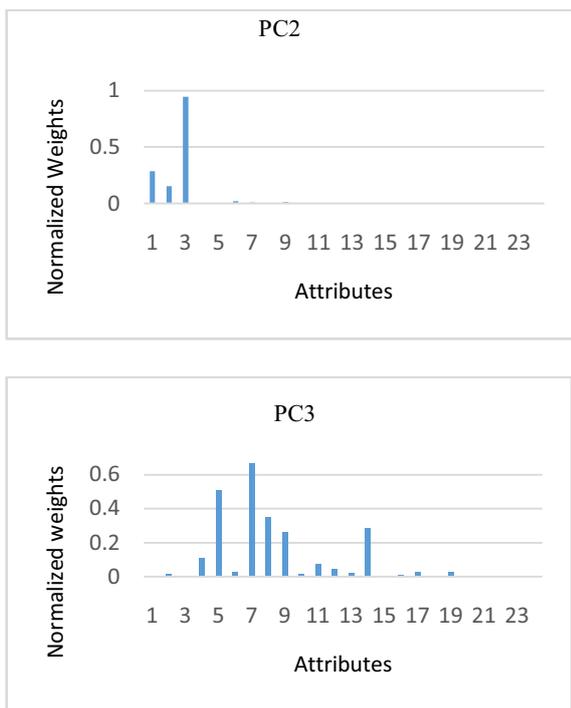


Figure 11. Relative significance of each German dataset attribute in the calculation of PC1 , PC2 and PC3.

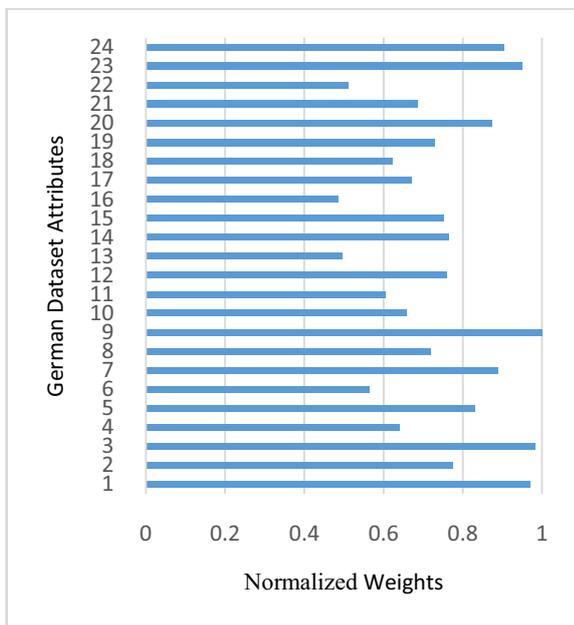


Figure 12. Normalized input weights of the trained ANN.

References

1. Lyn C. Thomas, A survey of credit and behavioral scoring: forecasting financial risk of lending to

consumers, *International Journal of Forecasting*, vol. 16, issue 2, pp. 149-172, April-June 2000.

2. E. I. Altman, "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *Journal of Finance*, vol. 23, pp. 89-609, 1968

3. Kleimeier, S., & Dinh, T. A. (2007). Credit scoring model for Vietnam's retail banking market. *International Review of Financial Analysis*, 16(5), 471-495.

4. Rayo, S., Lara, J., & Camino, D. (2010). A credit scoring model for institutions of microfinance under the Basel II normative. *Journal of Economics, Finance & Administrative Science*, 15(28), 89-124.

5. Viganò, L. A. (1993). Credit scoring model for development banks: An African case study. *Savings and Development*, 17(4), 441-482.

6. Vogelgesang, U. (2003). Microfinance in times of crisis: The effects of competition, rising indebtedness, and economic crisis on repayment behaviour. *World Development*, 31(12), 2085-2114.

7. Lee, T. S., & Chen, I. F. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications*, 28(4), 743-752.

8. Yu, L., Wang, S. A. and Lai, K. K. 2008. Credit risk assessment with a multistage neural network ensemble learning approach. *Expert systems with applications*. vol. 34. pp. 1434-1444.

9. Tsai, C.-f. and Wu, J.-w. 2008. Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert systems with applications*. vol. 34. pp. 2639-2649.

10. Angelini, E., Tollo, G. D. and Roil, A. 2008. A neural network approach for credit risk evaluation. *The quarterly review of economics and finance*. vol. 48. pp. 733-755.

11. Antonio Blanco, Rafael Pino-Mejías, Juan Lara, Salvador Rayo, Credit scoring models for the microfinance industry using neural networks: Evidence from Peru, *Expert Systems with Applications* 40 (2013) 356-364

12. Zongyuan Zhao a, Shuxiang Xu a, Byeong Ho Kang b, Mir Md Jahangir Kabir a, Yunling Liu, Rainer Wasinger a Investigation and improvement of multi-layer perceptron neural networks for credit scoring, *Expert Systems with Applications* 42 (2015) 3508-3516

13. Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu, "Credit rating analysis with support vector machines and neural networks: A Market Comparative Study," *Decision Support Systems*, vol. 37, no. 4, pp. 543-558, 2004.

14. K. K. Lai, L. Yu, S. Y. Wang, and L. G. Zhou, "Neural network meta-learning for credit scoring," in *Intelligent Computing*, vol. 4113 of *Lecture Notes in Computer Science*, pp. 403-408, 2006. View at Google Scholar

15. R. Malhotra and D. K. Malhotra, "Evaluating consumer loans using neural networks," *Omega*, vol. 31, no. 2, pp. 83-96, 2003. View at Publisher · View at Google Scholar · View at Scopus

16. Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, and S. Wu, "Credit rating analysis with support vector machines

- and neural networks: A Market Comparative Study,” *Decision Support Systems*, vol. 37, no. 4, pp. 543–558, 2004.
17. K. K. Lai, L. Yu, L. G. Zhou, and S. Y. Wang, “Credit risk evaluation with least square support vector machine,” in *Rough Sets and Knowledge Technology*, vol. 4062 of *Lecture Notes in Artificial Intelligence*, pp. 490–495, 2006.
 18. K. K. Lai, L. Yu, W. Huang, and S. Y. Wang, “A novel support vector machine metamodel for business risk identification,” in *PRICAI 2006: Trends in Artificial Intelligence*, vol. 4099 of *Lecture Notes in Artificial Intelligence*, pp. 480–484, 2006.
 19. R. Malhotra and D. K. Malhotra, “Differentiating between good credits and bad credits using neuro-fuzzy systems,” *European Journal of Operational Research*, vol. 136, no. 1, pp. 190–211, 2002.
 20. L. Zhou, K. K. Lai, and L. Yu, “Least squares support vector machines ensemble models for credit scoring,” *Expert Systems with Applications*, vol. 37, no. 1, pp. 127–133, 2010.
 21. L. Zhou, K. K. Lai, and L. Yu, “Credit scoring using support vector machines with direct search for parameters selection,” *Soft Computing*, vol. 13, no. 2, pp. 149–155, 2009.
 22. L. Yu and X. Yao, “A total least squares proximal support vector classifier for credit risk evaluation,” *Soft Computing*, vol. 17, no. 4, pp. 643–650, 2013.
 23. L. Yu, X. Yao, S. Wang, and K. K. Lai, “Credit risk evaluation using a weighted least squares SVM classifier with design of experiment for parameter selection,” *Expert Systems with Applications*, vol. 38, no. 12, pp. 15392–15399, 2011.
 24. Vaclav Kozeny, “Genetic algorithms for credit scoring: Alternative fitness function performance comparison” *Expert Systems with Applications*, vol. 42, pp. 2998–3004, 2015.
 25. T.-S. Lee, C.-C. Chiu, C.-J. Lu, and I.-F. Chen, “Credit scoring using the hybrid neural discriminant technique,” *Expert Systems with Applications*, vol. 23, no. 3, pp. 245–254, 2002.
 26. Malhotra, R., & Malhotra, D. K. (2002). Differentiating between good credits and bad credits using neuro-fuzzy systems. *European Journal of Operational Research*, 136(1), 190–211
 27. S. Piramuthu, “Financial credit-risk evaluation with neural and neurofuzzy systems,” *European Journal of Operational Research*, vol. 112(2), pp. 310–321, 1999
 28. K. K. Lai, L. Yu, S. Y. Wang, and L. G. Zhou, “Credit risk analysis using a reliability-based neural network ensemble model,” in *Proceedings of the International Conference on Artificial Neural Networks (ICANN '06)*, vol. 4132 of *Lecture Notes in Computer Science*, pp. 682–690, 2006.
 29. L. Yu, S. Wang, and K. K. Lai, “Credit risk assessment with a multistage neural network ensemble learning approach,” *Expert Systems with Applications*, vol. 34, no. 2, pp. 1434–1444, 2008.
 30. R. Smalz and M. Conrad, “Combining evolution with credit apportionment: a new learning algorithm for neural nets,” *Neural Networks*, vol. 7, no. 2, pp. 341–351, 1994.
 31. Y. Wang, S. Wang, and K. K. Lai, “A new fuzzy support vector machine to evaluate credit risk,” *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 6, pp. 820–831, 2005.
 32. Lean Yu, 2014, *Credit Risk Evaluation with a Least Squares Fuzzy Support Vector Machines Classifier*
 33. Terry Harris , *Credit scoring using the clustered support vector machine*, *Expert Systems with Applications*, 2015.
 34. S.Haykin, *Neural Networks: A comprehensive foundation*, Prentice Hall , 1998.
 35. L. Yu, J.-h. Zhu, L.-j. Chen *Parametric study on PCA-based algorithm for structural health Prognostics & System Health Management Conference*, Macao (2010), pp. 1–6.
 36. M.A. Rassam, A. Zainal, M.A. Maarof *An adaptive and efficient dimension reduction model for multivariate wireless sensor networks applications Applied Soft Computing*, 13 (2012), pp. 1978–1996
 37. Jie Wang, Jun Wang, *Forecasting stock market indexes using principal component analysis and stochastic time effective neural networks*, *Neurocomputing, Volume 156, 25 May 2015, Pages 68-78*
 38. Abdi. H., & Williams, L.J. (2010). "Principal component analysis." *Wiley Interdisciplinary Reviews: Computational Statistics*, 2: 433–459.
 39. T. Jolliffe, *Principal Component Analysis* , Springer Secaucus, NJ,USA,2002.
 40. Hussein A. Abdou1,* and John Pointon , *credit scoring, statistical techniques and evaluation criteria: a review of the literature*
 41. Rumelhart,D.E. G.E. Hinton, R.J. Williams *Learning internal representation by error propagation Parallel Distrib. Process.*, 1 (1986), pp. 318–362