Modeling Consumer Loan Default Prediction Using Ensemble Neural Networks

Amira Kamil Ibrahim Hassan
Department of computer science
Sudan University of Science and Technology, Sudan
Khartoum, Sudan
amirakamil2@yahoo.com

Ajith Abraham[1, 2]
[1]Machine Intelligence Research Labs (MIR Labs), WA, USA
[2]IT4Innovations, VSB - Technical University of Ostrava,
Czech Republic
ajith.abraham@ieee.org

Abstract—In this paper, a loan default prediction model is constricted using three different training algorithms, to train a supervised two-layer feed-forward network to produce the prediction model. But first, two attribute filtering functions were used, resulting in two data sets with reduced attributes and the original data-set. Back propagation based learning algorithms was used for training the network. The neural networks are trained using real world credit application cases from a German bank datasets which has 1000 cases; each case with 24 numerical attributes; upon, which the decision is based. The aim of this paper was to compare between the resulting models produced from using different training algorithms, scaled conjugate gradient backpropagation, Levenberg-Marquardt algorithm, One-step secant backpropagation (SCG, LM and OSS) and an ensemble of SCG, LM and OSS. Empirical results indicate that training algorithms improve the design of a loan default prediction model and ensemble model works better than the individual models.

Index Terms - credit risk, loan default, neural network, scaled conjugate gradient backpropagation, Levenberg-Marquardt algorithm and One-step secant backpropagation.

I. Introduction

Credit risk is one of the most studied and researched areas in banking. The loan default predicting model makes use of analysis techniques that use the current and historic information of the credit customer to make prediction about the credit customer ability to pay back on time [2]. An accurate consumer loan default detection system is an important reason for the bank profitability. Although techniques of credit measurement had advanced still it is a large risk [1]. The main aim of this paper is applying three different neural network-training algorithms; on a German bank real world credit application cases datasets, to produce a loan default prediction model. The original dataset has 1000 cases; each case with 24 numerical attributes. The creditability of a customer for loan giving depend on several parameters, such parameters include credit history, Installment rate, employment …etc. Another aim of this paper is to test benefit of using attribute filter on the model accuracy and develop an ensemble model by combining the outputs of the three different learning methods. The remaining part of this paper is organized as follows. Section 2 provides a summary of related work and short definitions of data mining techniques used. Section 3 describes the methods and ways of applying the used technique. Section 4 describes the data used in the study and introduces the results of the experiment. Section 5 discusses the results of the experiment and Section 6 concludes this paper.

II. Related Works

Angelini et al. [1] used a feed-forward neural network with classical topology and a feed-forward neural network with ad hoc connections, justifying their use of neural network that it is one of the best methods to design a prediction model. In their experiments, data of 76 small businesses from a bank in Italy were used. The conclusions reached that both methods produced efficient models that can correctly predict default with low error.

Tsai et al. [2] produced loan default prediction model using advanced Data Envelopment Analysis Discriminant Analysis (DEA–DA), the statistics-oriented discriminant analysis (DA), logistic regression (LR), and the neural networks (NN). A comparison was done between all these methods, using the accuracy percentage and found that DEA–DA and NN produced the best prediction models.

Akkoç [3], used a three stage hybrid Adaptive Neuro-Fuzzy Inference model, which is combination of statistics and Neuro-Fuzzy. A 10-fold cross was used for validation and a comparison with traditional models show that the produced model is much better.

Credit risk or loan default is considered part of CRM (customer relationship management). Jafarpour and Garvandani [4] showed the importance of the use of CRM system in banks, taking Iranian banks as an example. The suggested banking CRM model is based on relation between banks and customers dimensions through different relationship channels, which cause improvement in loyalty, life cycle and lifetime value of a customer. A Table and formula is designed that banks can use them to find their customers who can change to higher-level customers and can invest on them to change such customers to more loyal and profiTable customers. However it was not explained well how this Table and formula were designed and which technique was used.

Rani and Loshma [5] presented a framework of an evolving information system based on knowledge from data mining, and

978-1-4673-6232-0/13/$31.00 ©2013 IEEE

has discussed the framework by focusing on knowledge of classification. Their main focus was to research customer classification and prediction in Customer Relation Management concerned with data mining based on Back propagation technique. However back propagation can be time demanding but the use of multicore computers can solve the problem.

Ngai et al. [6] identified eighty seven articles related to application of data mining techniques in CRM, and published between 2000 and 2006. The majority of the reviewed articles relate to customer retention. The classification model is the most commonly applied model in CRM for predicting future customer behaviors. They also stated that neural networks were used in a wide range of CRM domains. However this study has some limitations, it surveyed articles published between 2000 and 2006.

Hsu and Hung [7] illustrated that support vector machine (SVM) is suiTable for the bank credit rating classifications. Furthermore, if the data samples increases and applied normal correlation significant test, or adopt other feature selection approach, the SVM predicting accuracy may increase, to make it more effective in rating issues such as the bank credit rating. As for multiple discriminate analysis (MDA), although it has the lowest training errors, it is likely to result in over-fitting, which caused the testing accuracy not accepTable. General if well chosen, feature selection approach improve the model accuracy.

This real world dataset, which classifies credit applicants described by a set of attributes as good or bad credit risks, has been successfully used for credit scoring and evaluation systems in many previous works [11-21].

ConsistencySubsetEval (CFs) "assesses each attribute predictive ability individually and degree of redundancy among the attributes, preferring sets of attributes that are highly correlated with the class but have low inter-correlation. An option iteratively adds attributes that have the highest correlation with the class, provided that the set does not already contain an attribute whose correlation with the attribute in question is even higher. ConsistencySubsetEval evaluates attribute sets by the degree of consistency in class values when the training instances are projected onto the set. The consistency of any subset of attributes can never improve on that of the full set, so this evaluator is usually used in conjunction with a random or exhaustive search that seeks the smallest subset whose consistency is the same as that of the full attribute set" [24].

PLsFilter "performs partial least square regression over the given instances and computes the resulting beta matrix for prediction" [24].

III. Methodology

An Artificial Neural Network (ANN) is characterized by the network architecture, the connection strength between pairs of neurons (weights), node properties, and updating rules. The updating or learning rules control weights and/or states of the processing elements. The network is initially randomized to avoid imposing any of our own prejudices about an application on the network. The training patterns can be thought of as a set of ordered pairs $\{(x_1, y_1), (x_2, y_2), \ldots, (x_P, y_P)\}$ where x_i represents an input pattern and y_i represents the output pattern vector associated with the input vector x_i .

Most of the ANN training algorithms use the gradient of the function to determine how to adjust the weights to minimize performance. The gradient is determined using a technique called backpropagation, which involves performing computations backwards through the network. One iteration of this algorithm can be written as:

$$x_{k+1} = x_k - \alpha_k g_k \qquad (1)$$

where x_k is a vector of current weights and biases, g_k is the current gradient, and α_k is the learning rate.

In the Conjugate Gradient Algorithm a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. A search is made along the conjugate gradient direction to determine the step size, which will minimize the performance function along that line. The Scaled Conjugate Algorithm (SCG) was designed to avoid the time consuming the line search. The key principle is to combine the model trust region approach with the conjugate gradient approach [22].

In a Quasi - Newton method (or secant), an approximate Hessian matrix is updated at each iteration of the algorithm. The update is computed as a function of the gradient. The One Step Secant (OSS) method is an attempt to bridge the gap between the computational complexity of conjugate gradient algorithms and the storage and computation in each iteration requirement in the Quasi-Newton algorithm. This algorithm does not store the complete Hessian matrix, it assumes that at each iteration the previous Hessian was the identity matrix [23]. Levenberg-Marquardt (LM) algorithm was designed to approach second order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares, then the Hessian matrix can be approximated to

$$H = J^T J; \qquad (2)$$

and the gradient can be computed as $g = J^T e$,

where J is the Jacobian matrix, which contains first derivatives of the network errors with respect to the weights, and e is a vector of network errors. The LM algorithm uses this approximation to the Hessian matrix in the following Newton-like update [23]:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e; \qquad (3)$$

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size.

IV. Data Pre-processing and Experimental Setep

A. Data description

A German bank real world credit application cases datasets consists of 20 attributes (7 numerical, 13 categorical). The categorical attributes were coded to form 24 attributes. The number of instances is1000. The last attribute (21st in the original dataset and 25th in the coded data set) is the output "should the customer be granted the loan, yes/no". Table (I)

shows the attributes in the original dataset they did not change much in the coded dataset only few attributes were broken into two or more attributes so they could be represented numerically.

TABLE 1. LIST OF ATTRIBUTES

Status of existing checking account	qualitative
Duration in month	numerical
Credit history	qualitative
Purpose	qualitative
Credit amount	numerical
Savings account/bonds	qualitative
Present employment since	qualitative
Installment rate in percentage of income	numerical
Personal status and sex	qualitative
Other debtors / guarantors	qualitative
Present residence since	numerical
Property	qualitative
Age in years	numerical
Other installment plans	qualitative
Housing	qualitative
Number of existing credits at this bank	numerical
Job	qualitative
Number of people liable to provide maintenance	numerical
Telephone	qualitative
foreign worker	qualitative

B. Data Cleaning and Preparation

The first step in data preparation was a descriptive statistics of the data shown in Table (II). The second step is to check for missing values or extreme values because if found they could affect the results of the experiment. As seen in the above Table there is no extreme values were found, since all the StdDev valves are relatively small. Also no missing valves were found in the dataset. The third step is to normalize the data; all the attributes were scaled to real numbers in the interval (0, 1). The normalization is important step before the use of neural network.

C. Experimental setup

A two-stage experiment was designed. In the first stage, two-attribute filtering functions (PLsFilter) and (ConsistencySubsetEval) were implied on the dataset, resulting in three different datasets. The original dataset with 24 attributes the second with 20 attributes and the third with 9 attributes. The role of attribute selection is to reduce the amount of data processing. Some data may not be useful, thus can be eliminated. This has the advantage of memory needs reduction, processing time reduction and improving the model [8].

In the second stage of the experiment a supervised two layer feed forward network, with sigmoid hidden neurons and output neurons was used. Back propagation learning algorithm was used for the network. After a trial and error approach by varying the number of neurons from 10 (default) to fifty, we finalized the architecture with 25 neurons. The input layer has 24 neurons, 20 neurons and 9 neurons. The output layer has 1 neuron. The network will be trained using SCG, OSS and LM algorithms. We use a default split of 60% data for training, 20% for testing, and the remaining 20% for validation. We used 10000 epochs. Join the output of the three models of each dataset to produce another three ensemble models.

TABLE 2. DESCRIPTIVE STATISTICS

	Max	Min	Average	StdDev
1	4	1	2.6	1.26
2	72	4	20.9	12.06
3	4	0	2.5	1.08
4	184	2	32.7	28.25
5	5	1	2.1	1.58
6	5	1	3.4	1.21
7	4	1	2.7	0.71
8	4	1	2.8	1.1
9	4	1	2.4	1.05
10	75	19	35.5	11.38
11	3	1	2.7	0.71
12	4	1	1.4	0.58
13	2	1	1.2	0.36
14	2	1	1.4	0.49
15	2	1	1.0	0.19
16	1	0	0.2	0.42
17	1	0	0.1	0.3
18	1	0	0.9	0.29
19	1	0	0.0	0.2
20	1	0	0.2	0.38
21	1	0	0.7	0.45
22	1	0	0.0	0.15
23	1	0	0.2	0.4
24	1	0	0.6	0.48
Output	2	1	1.3	0.46

V. Results and Analysis

Two attribute filtering functions were applied on the original 24 attribute dataset (DS1). PLsFilter was used on the original dataset and the result was a 20-attribute dataset (DS2). Then another attribute selection function was applied CfsSubsetEval, and the result was 9 attribute dataset (DS3).

The number of hidden neurons was adjusted starting from 10 up to 50, and the resulting networks were compared. Then the percentage of data allocated for testing and validation was changed from 15% up to 35%, observing the improvement in the accuracy percentage. A neural network with 25 hidden neurons was chosen, since it gives meaningful result with reasonable computational cost. The datasets were split 60% for training and 40% for testing and validation.

The neural network models final parameters are as follows; input layer neurons are 24, 20 and 9 respectively, hidden layer neurons 25, output neuron 1, maximum allowed iteration 10000 which was seen as suiTable size, training to validation and testing ratio 60% to 40% and training algorithms (LM, SCG and OSS). These models were run on three datasets (DS1, DS2 and DS3). For each one of the datasets three neural networks are modeled each one using one of the three training

algorithms which are studied in this paper. The experiment layout is shown in Figure 1.

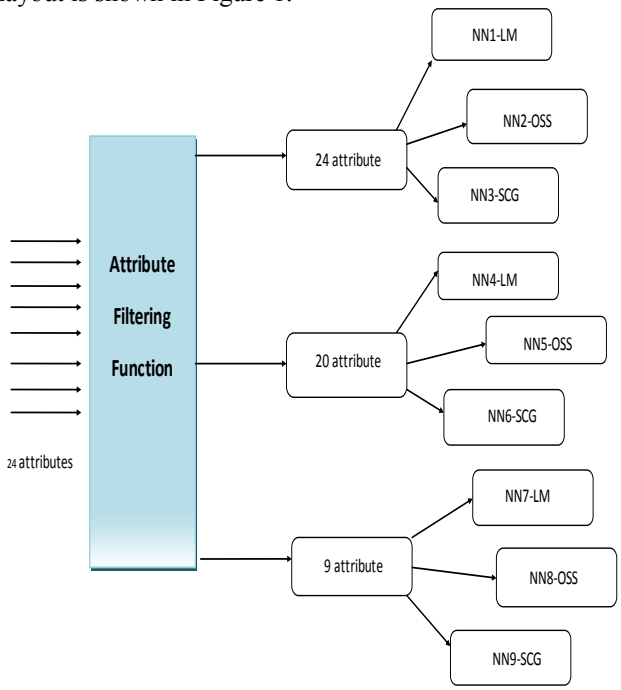

Fig.1. The experimental design

Table 3. Experimental parameters for the 9 models

Attribute selection	Algorithm	Itn	Trng time (min)	MSE	R
DS1 (24 attribute)	LM	5116	06:22	0.14	0.77
	SCG	9947	02:11	0.17	0.80
	OSS	10000	13:35	0.15	0.52
DS2 (20 attributes)	LM	6309	06:27	0.08	0.84
	SCG	9957	08:49	0.11	0.76
	OSS	10000	13:13	0.08	0.55
DS3 (9 attributes)	LM	5745	03:04	0.15	0.83
	SCG	9971	02:56	0.17	0.78
	OSS	10000	05:55	0.16	0.35

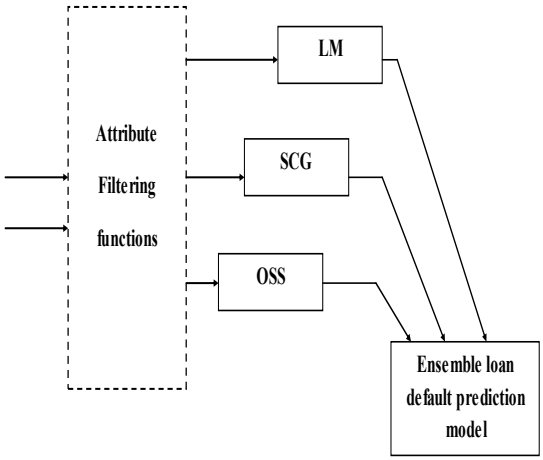

Fig.2. The ensemble model

For each network; iteration, training time, MSE and R, are recorded and shown in Table 3. In this experiment we have 9 models for each one of this model the accuracy percentages are calculated. These percentages are recorded in Table 4. After that three models were produced the output of the three models are combined as illustrated in Figure 2. If all three output agree then it is taken as the output of the ensemble model, if not then the output of the model with the higher weight is taken as the output of the ensemble model. The accuracy percentages of these three models were again calculated. The results are shown in Table 5. In Tables 4 and 5, non-default % is the percentage of non-default consumers that were correctly predicted, default% is the percentage of default consumers that were predicted correctly and the accuracy% is the total percentage.

TABLE 4. ACCURACY PERCENTAGE FOR THE 9 MODELS

Dataset	Algorithm	Non - default %	Default %	Accuracy%
DS1(24 attribute)	LM	89%	68%	83%
	SCG	88%	66%	81%
	OSS	92%	60%	82%
DS2(20 attributes)	LM	94%	89%	92%
	SCG	95%	76%	89%
	OSS	94%	62%	84%
DS3(9 attributes)	LM	88%	55%	78%
	SCG	90%	49%	76%
	OSS	90%	59%	81%

TABLE 5. ACCURACY PERCENTAGE FOR THE ENSEMBLE MODELS

Dataset	Non-default %	Default %	Accuracy%
DS1	94%	79%	89%
DS2	96%	98%	97%
DS3	91%	73%	86%

Comparing the Figures shown in Table 3 it is clear that OSS is the slowest of the three algorithms (13:35, 13:13 and 5:55), which mean that it is the most computationally expensive. The LM algorithm has the higher R (0.77, 0.84, and 0.83), making it very suiTable for models that use regression. SCG algorithm is had the highest R for DS1 making it more suiTable for larger datasets.

The accuracy percentage is the best parameter for comparison between the nine models. The best of the datasets is DS2 (92%, 89%, 84%) showing that the (PLsFilter) filtering function is the better filtering function which is logical since the other filtering function (ConsistencySubsetEval) produced a very small number of attributes. LM models are the best (83%, 92%) except in DS3 (78%). The results up to this stage show that the best model is that using (PLsFilter) filtering function and LM algorithm (92%). All three ensemble models gave much better results than all previous models (89%, 97%, 86%). The best of all these three models is the ensemble model of DS2 (97%).

VI. Conclusions

This paper presented an investigation of the use of supervised neural network models for customer loan default prediction under different training algorithms scaled conjugate gradient backpropagation, Levenberg-Marquardt algorithm and One-step secant backpropagation (SCG, LM and OSS). This paper also compared between two filtering functions and evaluation of the ensemble models. Several parameters were used in the experiment to do this comparison; training time , iteration, MSE and R. The slowest algorithm was OSS. The best algorithm was LM because it had the largest R.

The accuracy percentages of all models were calculated. First the filtering function was applied on the original dataset producing another two datasets. Then for each dataset three supervised neural network models, each one using different training algorithms. The results in Table 4 shows that LM algorithm and (PLsFilter) filtering function gave the best model. The ensemble models accuracy percentage were calculated and recorded in Table 5, showing that the ensemble model of the three algorithms (LM, OSS and SCG) of dataset (DS2) was the best model.

References

[1] E. Angelini, A. Roli, and G. di Tollo, "A neural network approach for credit risk evaluation," *elsevier, The Quarterly Review of Economics and Finance*, vol. 48, pp. 733–755, 2008.

[2] M. Tsai, S. Lin, C. Cheng, and Y. Lin, "The consumer loan default predicting model – An application of DEA – DA and neural network," *elsevier, Expert Systems With Applications*, vol. 36, no. 9, pp. 11682–11690, 2009.

[3] S. Akkoç, "An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis : The case of Turkish credit card data," *elsevier, European Journal of Operational Research*, vol. 222, pp. 168–178, 2012.

[4] H. Jafarpour and H. SheikholeslamiGarvandani, "New model of Customer Relationship Management in Iranian Banks," *icbme.yasar.edu.tr*, pp. 1–12, 2012.

[5] R. K Leela and G. Loshma, "Classification and Prediction in CRM Using Back Propagation Multilayer Feedforward Neural Network Approach," *IRACST – International Journal of Commerce, Business and Management (IJCBM)*, vol. 1, no. 1, pp. 1–7, 2012.

[6] E. W. T. Ngai, L. Xiu, and D. C. K. Chau, "Application of data mining techniques in customer relationship management: A literature review and classification," *elsevier, Expert Systems with Applications*, vol. 36, no. 2, pp. 2592–2602, Mar. 2009.

[7] C. F. Hsu and H. F. Hung, "Classification Methods of Credit Rating - A Comparative Analysis on SVM, MDA and RST," *2009 International Conference on Computational Intelligence and Software Engineering*, pp. 1–4, Dec. 2009.

[8] S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," *elsevier, Computer and Security*, vol. 24, pp. 295 – 307, 2005.

[9] S. Mukkamala, A. H. Sung, and A. Abraham, "Intrusion detection using an ensemble of intelligent paradigms," *elsevier, Journal of Network and Computer Applications*, vol. 28, pp. 167–182, 2005.

[10] I. Maqsood, M. R. Khan, and A. Abraham, "An ensemble of neural networks for weather forecasting," *Neural Comput & Applic*, vol. 13, pp. 112–122, 2004.

[11] P. O'Dea, J. Griffith, and C. O'Riordan, "Combining feature selection and neural networks for solving classification problems.," *Technical Report of the Dept of IT, NUI Galway*, vol. Number NUI, no. -IT-130601, 2001.

[12] J. Eggermont, J. N. Kok, and W. A. Kosters, "Genetic programming for data classification: Partitioning the search space.," *In Proceedings of the 2004 symposium on applied computing Cyprus.*, pp. pp. 1001–1005, 2004..

[13] J. J. Huang, G. H. Tzeng, and C. S. Ong, "Two-stage genetic programming (2SGP) for the credit scoring model," *Applied Mathematics and Computation*, vol. 174, pp. 1039–1053, 2006.

[14] Li, S. T., Shiue, W., and M. H. Huang, "The evaluation of consumer loans using support vector machines.," *Expert Systems with Applications*, vol. 30, no. 4, pp. 772–782, 2006.

[15] S. Piramuthu, "On preprocessing data for financial credit risk evaluation," *Expert Systems with Applications*, vol. 30, no. 3, pp. 489–497, 2006.

[16] C. L. Huang, M. C. Chen, and C. J. Wang, "Credit scoring with a data mining approach based on support vector machines.," *Expert Systems with Applications*, vol. 33, no. 4, pp. 847–856, 2007.

[17] A. Laha, "Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring.," *Advanced Engineering Informatics*, vol. 21, pp. 281–291, 2007.

[18] R. Setiono, B. Baesens, and C. Mues, "Recursive neural network rule extraction for data with mixed attributes," *IEEE Transactions on Neural Networks*, vol. 19, no. 2, pp. 299–307, 2008.

[19] C. F. Tsai and J. W. Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring .," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2639–2649, 2008.

[20] W. Yu, L., S. Y., and K. K. Lai, "An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring," *European Journal of Operational Research*, vol. 195, pp. 942–959, 2009.

[21] M. Šušteršic, D. Mramor, and J. Zupan, "Consumer credit scoring models with limited data," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4736–4744, 2009.

[22] A. F. Moller, "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning," *Neural Networks*, vol. 6, pp. 525–533, 1993.

[23] T. M. Hagan, H. B. Demuth, and M. H. Beale, "Neural Network Design," *Boston, MA: PWS Publishing*, 1996.

[24] I. H. Witten and E. Frank, *Data Mining Practical Machine Learning Tools and Techniques*, 2nd ed. Elsevier. 500 Sansome Street, Suite 400, San Francisco, CA 94111, 2005, pp. 393 – 402.