International Journal of Computational Intelligence Research.
ISSN 0973-1873 Vol.3, No.4 (2007), pp. 363–370

Modified Great Deluge Algorithm based Auto Associative Neural Network for Bankruptcy Prediction in Banks

C. Pramodh and V. Ravi*

Institute for Development and Research in Banking Technology,
Castle Hills, Road # 1, Masab Tank, Hyderabad - 500 057 (AP) India
E-mail : cpramodh@mtech.idrbt.ac.in; rav_padma@yahoo.com
** Corresponding Author, Phone: +91-40-2353 4981 Ext 2042; FAX: +91-40-2353 5157*

Abstract: **In the bankruptcy prediction literature, researchers proposed various techniques viz., neural networks, regression analysis, etc. But, in practice, the number of bankrupt banks is very low compared to that of nonbankrupt banks, at least in the Indian context. This induces the problem of data imbalance. To overcome this problem Baek and Cho [1] proposed an Auto Associative Neural Network (AANN) trained with one class i.e. solvent banks data. We proposed and implemented a variant for Baek and Cho's neural network and named it Modified Great Deluge Algorithm based Auto Associative Neural Network (MGDAAANN), wherein a meta-heuristic is used to train the auto associative neural network. The efficacy of the MGDAAANN is demonstrated on three bankruptcy data sets taken from the literature. Results indicate that MGDAAANN is a viable single-class classifier to predict bankruptcy in banks with high accuracy. Further, some of the MGDAAANN variants outperformed popular two-class classifiers such as RBF, orthogonal RBF, MLP, SVM and ANFIS.**

Keywords: Auto Associative Neural Networks, Modified Great Deluge Algorithm, Soft Computing and Bankruptcy prediction in banks.

I. Introduction

The prediction of bankruptcy for financial firms especially banks has been the extensively researched area since late 1960s [2]. Creditors, auditors, stockholders and senior management are all interested in bankruptcy prediction because it adversely affects all of them alike [3]. The most precise way of monitoring banks is by conducting on-site examinations. These examinations are conducted on a bank's premises by regulators every 12–18 months, as mandated by the Federal Deposit Insurance Corporation Improvement Act of 1991. Regulators utilize a six part rating system to indicate the safety and soundness of the institution. This rating, referred to as the CAMELS rating, evaluates banks according to their basic functional areas: ***C****apital adequacy*, ***A****sset quality*, ***M****anagement expertise*, ***E****arnings strength*, ***L****iquidity*, and ***S****ensitivity to market risk*. While CAMELS ratings clearly provide regulators with important information, Cole and Gunther reported that these CAMELS ratings decay rapidly [4]. Fraser noted that banks performed better by holding relatively more securities and fewer loans in their portfolios [5].

A variety of statistical techniques such as regression analysis, logistic regression have been used to solve the problem of bankruptcy prediction. These techniques typically make use of the company's financial data to predict the financial state of the company (healthy, distressed, high probability of bankruptcy). Altman pioneered the work of using financial ratios and multiple discriminant analysis (MDA) to predict financially distressed firms. However, the usage of MDA or statistical techniques, in general, relies on the restrictive assumption on *linear separability*, *multivariate normality* and *independence of the predictive variables* [6]-[8]. Unfortunately, many of the common financial ratios violate these assumptions. Bankruptcy prediction problem for financial firms can also be solved using various other types of classifiers. An overview of such works is given as follows. Tam explored a backpropagation trained neural network (BPNN) for this problem and compared its performance with methods such as *MDA*, *logistic regression*, *k-nearest neighbour* (k-NN) method and *ID3* [9]. He concluded that neural network outperformed other prediction techniques. Salchenberger et al. found that the neural network produced fewer or equal number of classification errors for each of the forecast periods compared to the logit model [10]. This conclusion holds for total errors, type I errors, and type II errors.

Ryu and Yue introduced a method called isotonic separation and its efficacy is evaluated in the prediction of firm bankruptcy. They claimed that isotonic separation is a viable technique for short-term bankruptcy prediction [11].

Tam and Kiang found that a neural network performs better than *statistical methods* and *decision trees* [12]. Consequently, many researchers viewed neural network as an attractive alternative to statistical techniques for bankruptcy prediction**.** Wilson and Sharda compared the performance of the neural networks vis-à-vis the MDA proposed in Altman. Atiya provided a survey of all the prediction techniques including neural networks applied to the bankruptcy prediction problem and proposed more financial indicators, in addition to the traditional ones, which he used in the design of a new neural network model [13]. Becerra et al., analyzed the use of linear discriminant models, multi-layer perceptron and wavelet networks for corporate financial distress prediction. They reported that the nonlinear models may be a valid alternative to the linear discriminant models and wavelet networks may have advantages over the multi-layer perceptron [14]. Shin et al. applied SVM to the problem of corporate bankruptcy prediction [15]. They concluded that SVM outperformed the MLFF-BP in terms of accuracy as the training dataset size got smaller. Canbas et al. proposed a methodological framework for constructing the integrated early warning system (IEWS) that can be used as a decision support tool in bank examinations and supervision process for detection of banks, which are experiencing serious problems [16].

Ravi Kumar and Ravi proposed a fuzzy rule based classifier for bankruptcy prediction [17]. They reported that fuzzy rule based classifier outperformed the well-known technique, MLFF-BP in the case of US banks data. Ravi et al. proposed a semi-online training algorithm for the radial basis function neural networks (SORBF) and applied it to bankruptcy prediction in banks [18]. Semi Online RBFN without linear terms performed better than techniques such as ANFIS, SVM, MLFF-BP, RBF and Orthogonal RBF. Cheng et al, combined Radial Basis Function Network with Logit Analysis Learning to predict Financial Distress [19]. They compared the proposed technique with logit analysis and a backporpagation neural network and found that their method is superior to both the techniques. Ravi Kumar and Ravi proposed an ensemble classifier for the bankruptcy prediction problem based on a host of intelligent techniques [20]. The ensemble classifier was developed using simple majority voting scheme and as part of the ensemble they employed seven classifiers such as ANFIS, SVM, RBF, SORBF1, SORBF2, Orthogonal RBF and MLFF-BP. They reported that, models ANFIS, SORBF2, MLP are the most prominent as they appeared in the best ensemble classifier combinations. Further, Ravi et al. developed a novel soft computing system for bank performance prediction based on MLP, RBF, CART, PNN, FRBC and PCA based hybrid techniques [21].

In another work, Ravi Kumar and Ravi conducted a comprehensive review of all the works reported using statistical and intelligent techniques to solve the problem of bankruptcy prediction in banks and firms during 1968-2005 [22]. It compares the techniques in terms of prediction accuracy, data sources, timeline of each study wherever available. Recently, Hu et al., proposed a method for bankruptcy prediction called Functional Link Network with Fuzzy Integral (FIFLN) [23]. Functional Link net can be used to enhance the performance of the traditional single-layer perceptron (SLP). They also designed a genetic algorithm based training method to estimate the weights in their network. They concluded that the FIFLN outperformed the functional-link net and SLP in forecasting bankruptcy. It should be noted that all these techniques are two-class classifiers.

As regards, single-class classifiers, Baek and Cho [1] proposed an auto associative neural network for bankruptcy prediction and concluded that it outperformed the 2-class neural networks. In this present paper, we developed a soft computing framework where a global optimization meta-heuristic viz., Modified Great Deluge algorithm (MGDA) proposed in [24] is employed to train an auto associative neural network.

The rest of the paper is organized as follows. We present the architectural details of the network in section 2. MGDA based training algorithm and flow chart are presented in section 3. The results and discussion and conclusions are presented in sections 4 and 5 respectively.

II. Architecture of the modified great deluge algorithm trained auto associative neural network

The auto associative neural network identically maps the predictor variables into themselves. Therefore, the number of nodes in the input layer and output layer is equal to the number of predictor variables in the data set. Auto associative neural networks essentially have three hidden layers as represented in the Figure-1, viz., compression layer, bottleneck layer and decompression layer. In the first part of the network the input feature space is compressed and mapped onto the subspace and is captured in the weights between the compression layer and bottleneck layer represented by w_{h1h2} as shown in the Figure 1. And in the second part of the network the decompression takes place that the feature subspace is mapped onto the hyper surface and is captured in the weights between bottleneck layer and decompression layer represented by w_{h2h3}.

We proposed four variants in the process viz., 1) MGDAAANN with nonlinear bottleneck layer and with bias nodes, 2) MGDAAANN with nonlinear bottleneck layer and without bias nodes, 3) MGDAAANN without nonlinear bottleneck layer and with bias nodes, 4) MGDAAANN without nonlinear bottleneck layer and without bias nodes. The name of the variant itself explains the nature of the network, i.e. MGDAAANN with nonlinear bottleneck layer, consists of three hidden layers in which, all the layers are nonlinear in nature, and there exists bias nodes in all the layers.

While training the auto associative neural network in the first iteration we initialized the weights between all the layers, which are drawn from a particular range using a uniform random number generator. In the subsequent iterations we employed a neighborhood search for updating the weights using the Modified great deluge algorithm. The inputs are pass through the Compression layer; the outputs of the compression layer are pass through the bottleneck layer, where the dimensionality reduction takes place. The outputs of the bottleneck layer are supplied to the decompression layer where the dimension expansion takes place. These outputs are again passed through a sigmoid transfer function to get the output values. The same will be repeated until we get the least MSE (mean squared error) in the order of two decimal points for the training patterns.

Once the network is trained sufficiently, we used it for testing. While training we supplied only solvent banks data like Baek and Cho [1] and this is where we exploited the advantages of AANN. In testing, we supplied insolvent banks data and observed that the training data set yielded very low MSE value whereas the test data yielded high MSE values for the same weight vectors.

Because we trained the network with only the solvent data, the network will learn the characteristics of the solvent banks only but not the insolvent banks. Thus, high MSE value for test data is obtained. With this scheme, we can classify the test pattern as a solvent or an insolvent bank. We employed a mechanism using a threshold for classifying the pattern. In testing, the pattern should result in a higher MSE value than the lowest MSE value obtained while training in order to be classified as insolvent bank. We calculated the classification rates for the threshold value of 0.05 and presented in Table 2.

III. Training Algorithm

In our study, we employed the global optimization meta heuristic Modified Great Deluge algorithm to train the 5-layered auto associative neural network unlike Baek and Cho [1]. Ravi demonstrated that MGDA is a sound alternative to the meta-heuristics like SA (Simulated Annealing), INESA (Improved Non Equilibrium Simulated Annealing) and ACO (Ant Colony Optimization) in reliability optimization [24]. To overview Great Deluge algorithm briefly, Dueck and Scheuer proposed a faster variant of SA namely the Threshold Accepting (TA) algorithm [25]. Deuck further extended the TA by proposing two new optimization meta-heuristics viz., the great deluge algorithm (GDA) and record-to-record travel (RRT) [26]. He observed that GDA outperformed TA in case of some hard, benchmark instances of the traveling salesman problem. Ravi developed an extended version of the GDA called Modified Great deluge Algorithm (MGDA) and demonstrated its effectiveness in solving reliability optimization problems [24].

The decision variables for MGD training algorithm are all the four sets of weights between all the layers. We employed the mean squared error as an objective function, sigmoid as a transfer function in all the layers, which is given as follows.

$$f(x) = \frac{1}{(1 + e^{-x})}$$

After the network is trained sufficiently the testing phase consisting of only forward pass uses the optimal weights obtained of the training phase. The outputs are calculated and the corresponding patterns are classified using a prespecified threshold value. We present the step-by-step procedure of the MGDA trained AANN as follows.

Step 1

Normalize the input data to make sure that all the attributes are in the same range [0, 1] as follows

$$x_{ij} = \frac{x_{ij} - x\min_j}{x\max_j - x\min_j}$$

$$i = 1, 2, \ldots\ldots\ldots n \;\; and \;\; j = 1, 2, \ldots.m.$$

where x_{minj}, x_{maxj} are the minimum and maximum of the j^{th} input variable and n, m are the number of patterns and the input variables respectively.

Step 2

For the sake of simplicity of computations in the network we break up the network into two parts, first part which maps the input patterns onto the subspace, dimension reduction; second part which maps the reduced dimension onto the hyper surface, dimension expansion.

Computations in the first part of the network are as follows. The net input to the compression layer nodes can be calculated as below,

$$neth1_j = \sum_{i=1}^{nin} wih1_{ij}\, x_i\,,\; j = 1, 2, \ldots, nhn1$$

where , i = 1,2,…nin; nin is the number of nodes in input layer; j = 1,2…nhn1; nhn1 is the number of nodes in compression layer; k = 1,2…n. Then, the output of the j^{th} node of the compression layer is computed as follows:

$$outh1_j = \frac{1}{1 + e^{-neth1_j}}$$

Step 3

Now supply the outputs of compression layer as inputs to the bottleneck layer. Then, net input to the bottleneck layer nodes can be calculated as follows:

$$neth2_j = \sum_{i=1}^{nhn1} wh_1h_{2ij}\, outh1_i\,,\; j = 1, 2, \ldots, nhn2$$

where i = 1,2,…nhn1 number of nodes in compression layer, j = 1,2…nhn2 number of nodes in bottleneck layer. Then the outputs of the bottleneck layer can be calculated as follows:

Output of the j^{th} node of the bottleneck layer depends on the variant type, because we employed two types of bottleneck layer nodes viz., linear and nonlinear. If nonlinear bottleneck layer nodes are used, then output of the j^{th} node of the bottleneck layer is given as follows:

$$outh_{2j} = \frac{1}{1+e^{-neth2_j}}$$

Step 4

This completes the first part. Second part starts by supplying these bottleneck layer outputs to the decompression layer. Then, the net input to the nodes of decompression layer can be calculated as follows:

$$neth3_j = \sum_{i=1}^{nhn2} wh_2h_{3ij}\, outh2_i,\ j = 1,2,\ldots,nhn3$$

where i = 1,2,…nhn2 number of nodes in bottleneck layer, j = 1,2,…nhn3 number of nodes in decompression layer. Then the outputs of the bottleneck layer can be calculated as follows:

$$outh_{3j} = \frac{1}{1+e^{-neth3_j}}$$

Step 5

Now the final computation in the second part comprises the calculation of the final outputs. The net input to the output layer nodes are calculated as below,

$$neto_j = \sum_{i=1}^{nhn3} wh_3o_{ij}\, outh3_i,\ j = 1,2,\ldots,non$$

where, i = 1,2,… nhn3 number of nodes in bottleneck layer, j = 1,2… nin number of nodes in decompression layer.

The final outputs are calculated as below,

$$out_j = \frac{1}{1+e^{-neto_j}}$$

This completes the second part of the network. Now using these predicted output values we should calculate the objective function value.

Step 6

Repeat steps 2-5 for all patterns. Then calculate the objective function value, which is mean squared error (MSE) for the current iteration

$$MSE = \sum_{i=1}^{n}\sum_{j=1}^{m}(x_{ij} - out_{ij})^2$$

If the MSE value is acceptable, stop the algorithm. Otherwise, continue the same with the different set of weights updated by using the neighborhood search scheme of the MGDA.

The flow chart for the MGDA is presented in the Figure 2. The inner iterations perform a neighborhood search. For neighborhood search the formula used is as follows:

$$w_j[i][j]=w_i[i][j]+(2u-1)^p$$

Where, w_j is the new set of weights obtained from the previous set of weights w_i, u is the random number drawn from a uniform distribution in [0,1], p is an odd integer. *UP* is another important parameter, which is used in accepting or rejecting a neighborhood solution. Parameter *limit* suggests the number of neighborhood searches to be made in a given global iteration.

IV. Results and Discussions

We worked on three different banks data sets viz., Turkish banks, Spanish banks and US banks datasets. Turkish banks' data, obtained from Canbas et al. is available at (http://www.tbb.org.tr/english/bulten/yillik/2000/ratios.xls) [16]. Banks association of Turkey published 49 financial ratios. Initially, Canbas applied univariate analysis of variance (ANOVA) test on these 49 ratios of previous year for predicting the health of the bank in present year. However, Canbas et al. chose only 12 ratios as the early warning indicators that have the discriminating ability (i.e. significant level is <5%) for healthy and failed banks one year in advance. Among these variables, 12^{th} variable has some missing values meaning that the data for some of the banks are not given. So, we filled those missing values with the mean value of the variable following the general approach in data mining. The financial ratios, which are considered as predictor variables are presented at the end of the paper in Table 1. This dataset contains 40 banks where 22 banks went bankrupt and 18 banks are healthy. The Spanish banks' data is obtained from Olmeda and Fernandez [27]. Spanish banking industry suffered the worst crisis during 1977-85 resulting in a total cost of 12 billion dollars. The considered financial ratios are presented in the end of the paper in Table 1. The ratios used for the failed banks were taken from the last financial statements before the bankruptcy was declared and the data of non-failed banks was taken from 1982 statements. This dataset contains 66 banks where 37 went bankrupt and 29 healthy banks. The US banks' data is obtained from Rahimian et al. [28]. The financial ratios used by them are presented in Table 1. They obtained the data of 129 banks from the Moody's Industrial Manual, where banks went bankrupt during 1975-1982.

Table 1 : Predictor variableof datasets used in the study.

Turkish banks' data	
S. NO.	**Predictor Variable Name**
1	Interest expenses/Average profitable assets
2	Interest expenses/Average non-profitable assets
3	(Share holders' Equity + Total income)/(Deposits + Non-deposit funds)
4	Interest income/Interest expenses
5	(Share holders' Equity + Total income)/Total assets
6	(Share holders' Equity + Total income)/(Total assets + Contingencies & Commitments)
7	Networking Capital/Total assets
8	(Salary and Employees' benefits + Reserve for retirement)/No.

	of personnel
9	Liquid Assets/(Deposits + non-deposit funds)
10	Interest expenses/Total expenses
11	Liquid assets/total assets
12	Standard Capital ratio
Spanish banks' data	
S. NO.	Predictor Variable Name
1	Current assets/total assets
2	Current assets-cash/total assets
3	Current assets/loans
4	Reserves/loans
5	Net income/total assets
6	Net income/total equity capital
7	Net income/loans
8	Cost of sales/sales
9	Cash flow/loans
US banks' data	
S. NO.	**Predictor Variable Name**
1	Working capital/Total assets
2	Retained earnings/Total Assets
3	Earnings before interest and taxes/Total Assets
4	Market value of equity/Total Debt
5	Sales/Total Assets.

We calculated the classification rate in testing phase as follows. We obtained the relative error between the actual output and predicted output for all the testing patterns for each attribute. If that error is above the threshold for all the attributes of a pattern then the pattern is classified correctly otherwise it is misclassified. We implemented the MGDAAANN in C language in visual studio 6.0 IDE. All the simulations are done on the system with 2.4 GHz Processor, 256 MB RAM.

Till now there are no prescribed rules reported in literature for selecting the number of nodes in the hidden layers of auto associative neural network. Hence, we selected the hidden nodes in a trial and error manner. Typically, the number of nodes in the compression and decompression layers is one and the same and the nodes in the bottleneck layer are less than the input and output layer nodes.

According to the results presented in the Table 2, variants with bottleneck layer outperformed the variants without bottleneck layer. In the variants with bottleneck layer, the one without bias nodes outperformed the one with bias nodes with the classification rates of 72.97%, 70.27% in case of Spanish banks and 87.69%, 84.62% in the case of US banks case respectively. In the case of Turkish banks, both the variants performed equally well with the classification rate of 72.73%. In the variants without bottleneck layer also, the variant without bias nodes outperformed the one with bias nodes with the classification rates of 70.27%, 67.57% in the case of Spanish banks, 68.2%, 63% in the case of US banks and 76.92%, 72.73% in the case of Turkish banks respectively. With the above observations we can infer that having a bottleneck layer is helpful whereas having the bias nodes is not helpful in an auto associative neural network.

The variant with bottleneck layer and without bias nodes outperformed all other variants in case of Spanish banks and US banks, whereas in case of Turkish banks the variant without bottleneck layer and without bias nodes outperformed others.

It should be noted that MGDAAANN is a single-class classifier. However, we wanted to test its efficacy vis-à-vis several two-class classifiers viz., multi layer perceptron (MLP), radial basis function network (RBFN), orthogonal RBFN, Semi-online RBF with and without linear terms, rough set based expert system (RSES), support vector machine (SVM) and adaptive neuro-fuzzy inference system (ANFIS). The results presented in Table 2 for these two-class classifiers are the average classification rates over all the folds in 10-fold cross validation. However, 10-fold cross validation in the case of single-class classifiers does not make sense. Notwithstanding these facts, the comparative results indicate that in the case of Spanish banks, MGDAAANN variants outperformed orthogonal RBFN, ANFIS and Semi-online RBF with linear terms. In the case of US banks, MGDAAANN with bottleneck layer and without bias nodes outscored not only the other MGDAAANN variants but also the orthogonal RBF, RBF, semi-online RBF with linear terms, semi-online RBF without linear terms, SVM and MLP. In the case of Turkish banks, MGDAAANN without bottleneck layer and without bias nodes outperformed not only the other MGDAAANN variants but also the orthogonal RBF, RBF and ANFIS.

In Turkish banks case almost all the variants performed equally well with the classification rate of 72.73% except the variant without bottleneck layer and without bias nodes with the classification rate of 76.92%. Hence we can compare the variants with respect to the Spanish banks and US banks data sets. We can arrange the variants in ascending order of their classification rates as follows, MGDAAANN with bottleneck layer without bias nodes, MGDAAANN with bottleneck layer and with bias nodes, MGDAAANN without bottleneck layer without bias nodes and MGDAAANN without bottleneck layer with bias nodes.

Table 2 : Comparison of MGDAAANN variants with other techniques.

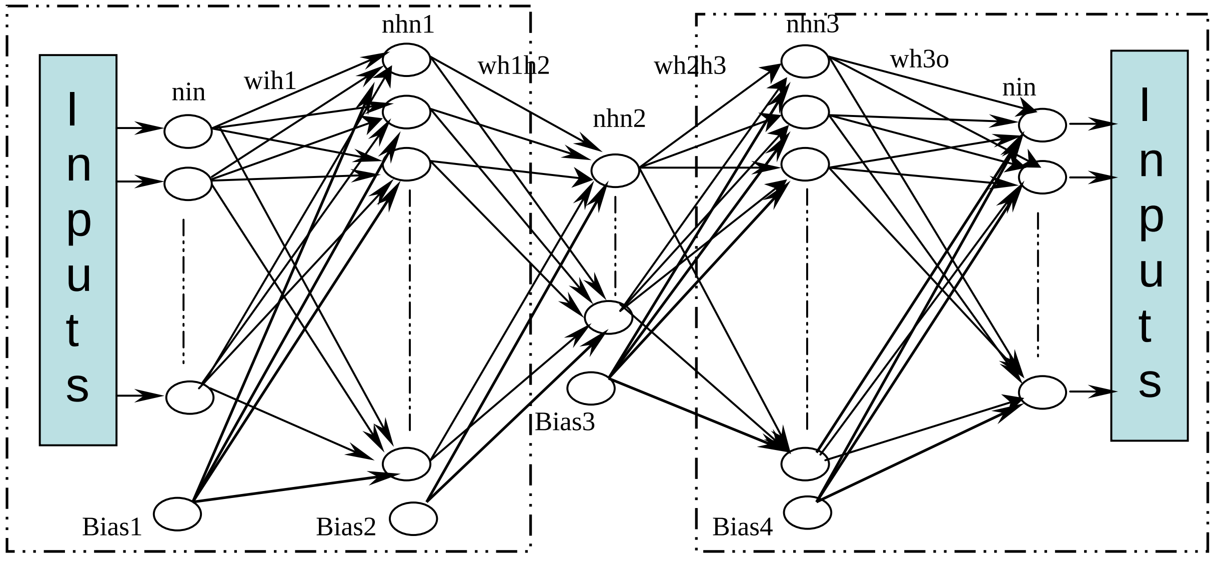
S. No.	**Classification Techniques**	**Classification Rate**		
		Spanish banks	**US banks**	**Turkish banks**
1	MGDAAANN with Bottleneck layer and with bias nodes	70.27	84.62	72.73
2	MGDAAANN with Bottleneck layer and without bias nodes	72.97	87.69	72.73
3	MGDAAANN without Bottleneck layer and with bias nodes	67.57	63.00	72.73
4	MGDAAANN without Bottleneck layer and without bias nodes	70.27	68.20	76.92

5	Orthogonal RBF*	40.83	55.95	50
6	RBF*	75	77.26	65
7	Semi-online RBF with linear terms*	58.31	76.9	80
8	Semi-online RBF without linear terms *	88.32	87.38	87.5
9	RSES*	92.5	95.89	87.5
10	SVM*	82.5	86.42	90
11	MLP*	80.0	87.4	87.5
12	ANFIS*	63.34	92.03	60

* 10-fold cross validation was performed in these cases and average results over 10 folds are presented

V. Conclusions

In this paper we proposed four variants of modified great deluge auto associative neural network viz., 1) MGDAAANN with nonlinear bottleneck layer and with bias nodes, 2) MGDAAANN with nonlinear bottleneck layer and without bias nodes, 3) MGDAAANN without nonlinear bottleneck layer and with bias nodes and 4) MGDAAANN without nonlinear bottleneck layer and without bias nodes. Out of the four variants, the variants without bias nodes outperformed other ones over all data sets. The second variant outperformed other variants in case of Spanish and US banks datasets, whereas the fourth variant outperformed others in case of Turkish banks data set. Out of the four variants, the third variant did not performed well uniformly in all data sets. Some of the MGDAAANN variants outperformed popular two-class classifiers such as RBF, orthogonal RBF, MLP, SVM and ANFIS. Based on the results, it is concluded that MGDAAANN can be employed as the sound alternative to the extant one-class classifiers.



1) *Modified Great Deluge Algorithm Auto Associative Neural Network*

Figure 1 : Modified Great Deluge Algorithm Auto Associative Neural Network

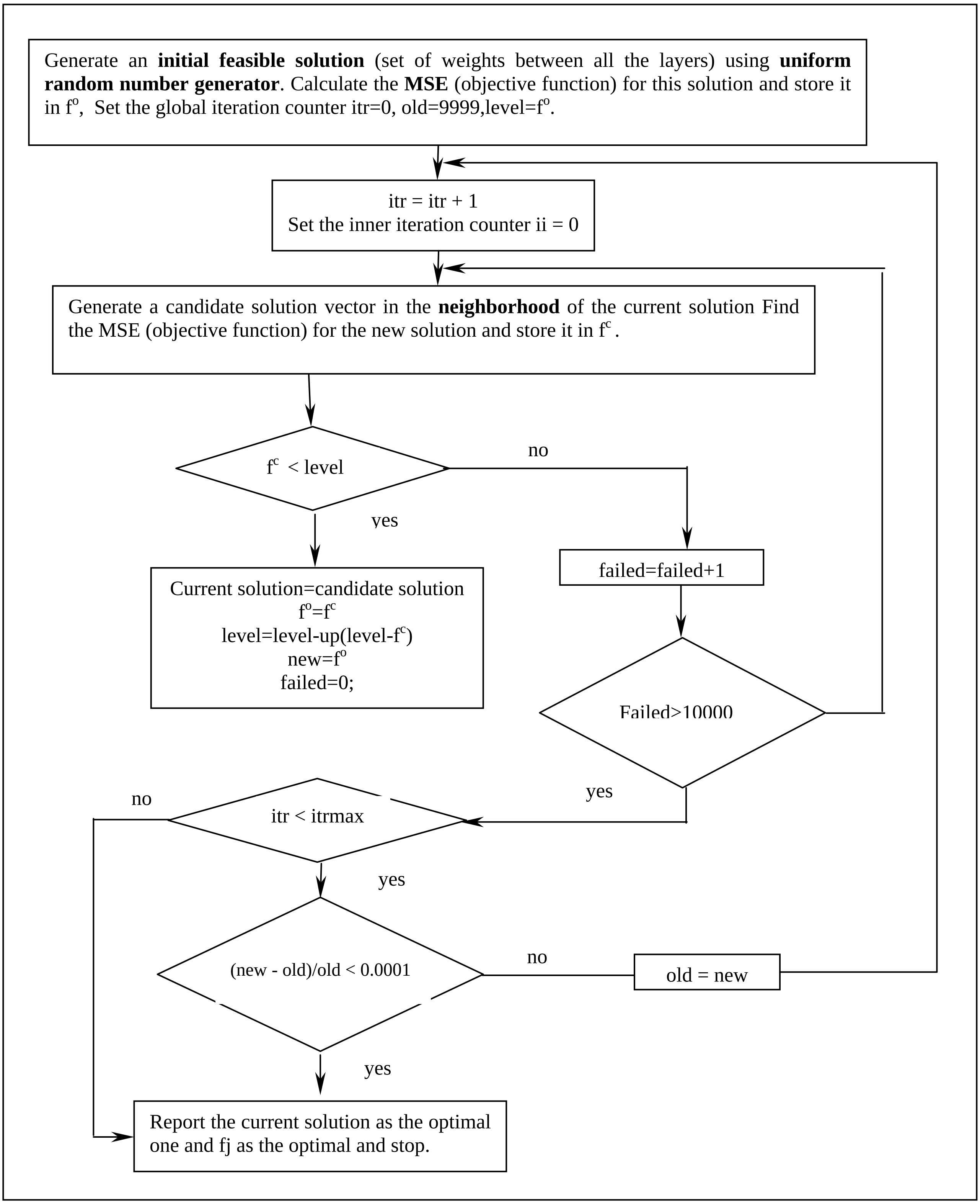


Figure 2. Flow chart for Modified Great Deluge algorithm

References

[1] Jinwoo Baek, Sungzoon cho, "Bankruptcy Prediction for credit Risk Using an Auto-Associative Neural Network in Korean Firms", *In the proceedings of the CIFEr* Hong Kong, 2003.

[2] E. Altman. "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy", Journal of Finance, 23, pp. 589-609, 1968.

[3] R. L. Wilson, & R.Sharda. "Bankruptcy prediction using neural networks", Decision Support Systems, 11 pp. 545-557, 1994.

[4] R. Cole & J. Gunther. "A CAMEL Rating's Shelf Life", Federal Reserve Bank of Dallas Review, pp. 13-20, December, 1995.

[5] D. Fraser. "The Determinants of Bank Profits: An Analysis of Extremes", Financial Review, 11, pp. 69-87, 1976.

[6] G. V. Karels and A. J. Prakash. "Multivariate normality and forecasting for business bankruptcy", Journal of Business Finance & Accounting, 14, pp. 573-593, 1987.

[7] M. Odom, R. Sharda. "A Neural Network for Bankruptcy Prediction", *In proceedings of the IJCNN International Conference on Neural Networks*, San Diego, CA, 1990.

[8] J. A. Ohlson. "Financial Rations and the Probabilistic Prediction of Bankruptcy", Journal of Accounting Research, 18, pp. 109-131, 1980.

[9] K. Y. Tam. "Neural Network Models and the Prediction of Bank Bankruptcy", OMEGA, 19, pp. 429-445, 1991.

[10] L. Salchenberger, C. Mine and N. Lash. "Neural Networks: A Tool for Predicting Thrift Failures", Decision Sciences, 23, pp. 899-916, 1992.

[11] Y. U. Ryu, W. T. Yue, "Firm Bankruptcy Prediction: Experimental Comparison of Isotonic Seperation and other classification Approaches", IEEE transactions on Systems, MAN and Cybernetics, systems and humans, vol 35, No 5, pp.727-737, September, 2005.

[12] K. Y. Tam, M. Kiang. "Predicting Bank Failures: A Neural Network Approach", Decision Sciences, 23, pp. 926-947, 1992.

[13] A.F. Atiya. "Bankruptcy prediction for credit risk using neural networks: A survey and new results", *In the IEEE Transactions on Neural Networks*, 12, pp. 929-935, 2001.

[14] V. M. Becerra, R. K. H. Galvao, M. Abou-Seads, "Neural and Wavelet Network Models for Financial Distress Classification", Data Mining and Knowledge Discovery, 11, pp. 35-55, 2005.

[15] K. S. Shin, T. S. Lee and H. J. Kim. "An application of support vector machines in bankruptcy prediction model", Expert Systems with Applications, 28, pp. 127-135, 2005.

[16] S. Canbas, Caubak and S. B. Kilic. "Prediction of commercial bank failure via multivariate statistical analysis of financial structures: The Turkish case", European Journal of Operational Research, 166, pp. 528-546, 2005.

[17] P. Ravikumar and V. Ravi. "Bankruptcy prediction in banks by Fuzzy Rule based classifier", *In the proceedings of 1st IEEE International Conference on Digital and Information Management*, Bangalore, pp.222-227, 2006.

[18] V. Ravi, P. Ravi Kumar, E. Ravi Srinivas and N. K. Kasabov. "A Semi-Online training algorithm for the Radial Basis Function Neural Networks: Applications to Bankruptcy Prediction in Banks", in *Advances in Banking Technology and Management: Impact of ICT and CRM*", V. Ravi (eds.), IGI Global, USA, 2007.

[19] C. B. Cheng, C. L. Chen, C. J. Fu, "Financial Distress Prediction by a Radial Basis Function Network with Logit Analysis Learning, Computers and Mathematics with Applications, 51, pp. 579-588, 2006.

[20] P. Ravi Kumar and V. Ravi. "Bankruptcy prediction in Banks by an Ensemble classifier", *In the proceedings of IEEE International Conference on Industrial Technology*, Mumbai, pp.2032-2036, 2006.

[21] V. Ravi, H. Kurniawan, Peter NweeKok Thai and P. Ravi Kumar, Soft Computing System for Bank Performance prediction, accepted in Applied Soft Computing Journal, 8, 1, pp.305-315, 2008.

[22] P. Ravi Kumar and V. Ravi. "Bankruptcy prediction in banks and firms via statistical and intelligent techniques - A Review", European Journal of Operational Research, 180, 1, pp.1-28. 2007.

[23] Y. C. Hu, F. M. Tseng, "Functional-link net with fuzzy integral for bankruptcy prediction", Neurocomputing, 70, pp. 2959-2968, 2007.

[24] V. Ravi. "Modified Great deluge Algorithm versus Other Metaheuristics in Reliability Optimization, Asia-Pacific Journal of Operational Research, Vol. 21, No. 4, pp. 487-497, 2004.

[25] G. Dueck and T. Scheuer, "Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing", Journal of Computational Physics, 90, pp. 161-175, 1990.

[26] G. Dueck, New optimization heuristics, Journal of Computational Physics, 104, 1, pp. 86-92, 1993.

[27] I. Olmeda and E. Fernandez. "Hybrid classifiers for financial multicriteria decision making: the case of bankruptcy prediction", Computational Economics, 10, pp. 317-335, 1997.

[28] E. Rahimian, S. Singh, T. Thammachote and R. Virmani, "Bankruptcy prediction by Neural network" in R. R. Trippi and E. Turban (Eds.) Neural Networks in Finance and Investing, Irwin Professional Publishing, Burr Ridge, USA, 1996.