Hybrid Intelligent Classifier for Business Insolvency Classification System.

Conference Paper · January 2003

Source: DBLP

3 authors, including:

Azizi Ab Aziz
Universiti Utara Malaysia

Fadzilah Siraj
Universiti Utara Malaysia

Fadzilah Siraj, Azizi Zakaria, Azizi Ab. Azizi & Zulhazlin Abas. (2003). Paper 53. *AIAI 2003. Malaysian-Japan Seminar on Artificial Intelligence Applications in Industry*. 24-25 June 2003. Park Plaza Hotel, Kuala Lumpur.

WEBIC: A Web Based Business Insolvency Classifier using Neural Networks

Fadzilah Siraj, Azizi Zakaria, Azizi Ab Aziz and Zulhazlin Abas

Artificial Intelligence Special Interest Group (AISIG)
School of Information Technology
Universiti Utara Malaysia, 06010 Sintok, Kedah
E-mail:fad173@uum.edu.my,zizi1199@uum.edu.my,
azizi_ab_aziz@hotmail.com,zulhazlin@hotmail.com

Abstract— **Business insolvency is one of the major problems faced by decision makers, especially to detect the early symptom that may contribute to critical business condition. This paper discusses the implementation of neural networks in classifying business insolvency cases in Malaysia. The developed prototype can be accessed remotely via World Wide Web (WWW). For the development purposes, the data was obtained from the Registrar of Business / Companies (ROB/ROC), Kuala Lumpur Stock Exchange and Bank Negara Malaysia (Central Bank of Malaysia). Several experiments were conducted to determine the most suitable parameters for the neural network model. Based on the experimental results, a network with an architecture of 11-6-1 with learning rate 0.1 and momentum term of 0.5. The prototype obtained 90.25% generalization and therefore indicates that the prototype has the potential to be used as a tool for classifying business insolvency. Hence, the prototype provides a basic framework for developing such a classifier.**

Index Terms— **Business Insolvency, Classification, Decision Support, Neural Networks.**

I. Introduction

Business insolvency classification is a subject of great interest to practitioners and researchers alike. Started in early 1960's, business insolvency studies attract other researchers from numerous fields such as economic, statistics and also artificial intelligence. The study of this field is very important since firm contributes a major asset to sustain the economic growth. It allows for timely decision to be made relative to the reallocation of resources to more efficient uses. Technically, business insolvency whether refers the cessation of operations and liquidation of assets, or insolvency in either the equitable or legal sense, is of critical importance to both business community in general and society as a whole [1][2].

The ability to classify potential insolvency provides an early warning mechanism so that an appropriate adjustment in resource allocation can take place [3]. The rapid growth in computing technologies and the advancement in classification techniques give a new insight in business insolvency classification research works. This project concentrates on business insolvency classification using intelligent classifiers via World Wide Web.

WEBIC is a prototype of intelligent web based business insolvency classification system using Neural Networks. At present, the system employs a standard backpropagation learning algorithm and focuses on a few important concepts of business insolvency classification task such as variables selection, analysis and classifying the risk. The system can adapt to the changing business environment as a human expert, through artificial learning process [4]. Through appropriate learning model, the system can assist decision maker especially to foresee possible business risk and allowing an ample time for business assessment [5].

II. Artificial Neural Networks (ANN)

An ANN can be defined as "information processing system consisting of a large number of simple, highly interconnected processing elements in an architecture inspired by the structure of cerebral cortex of the brain".

Fundamentally, the ANN tries to mimic the functioning of human brain, which contains billions of neurons and interconnections [6].

Neural networks can learn from experience, generalize from previous example to new ones and abstract essential characteristics from inputs containing irrelevant data [7]. The neural network used in this prototype is a Multilayer Perceptron (MLP) model (refer Figure 1.0). The MLP is a three layer network with one hidden or more hidden layer and trained with back propagation algorithm.

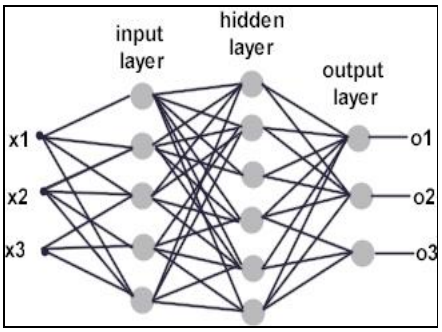

Figure 1.0: Network Architecture for MLP

The weights (or explicit knowledge) are updated after each training set is presented to the networks. The error for a single pattern is calculated as:

$$E_P = \frac{1}{2}\sum_{K=1}^{K}(T_K - O_K)^2 \qquad (1)$$

where K = output layer size

T_K = target value (desired)

O_K = actual output

The weights update are written as in equation (2) and (3):

$$w(t+1) = w(t) + \Delta w(t+1) \qquad (2)$$

$$\Delta w(t+1) = \eta \delta y(t) + \alpha \Delta w(t) \qquad (3)$$

where $w(t+1)$ = new weight

$w(t)$ = old weight

$\Delta w(t+1)$ = weight correction term

η = learning rate

α = momentum term

δ = local gradient (error signal)

Momentum term is added to accelerate the speed of convergence. The $y(t)$ is an activation function (squashing function) that computes the internal activity (signal) for neuron in the respective network layers. Equation (4) shows the computation for $y(t)$.

$$y(t) = \frac{1}{1+\exp^{-v(t)}} \qquad (4)$$

where *v(t)* is an incoming signal from the network.

III. Prototype Design and Implementation

The prototype was designed using Rational Rose 2000, and developed using Visual Basic 6.0 (classifier module), Microsoft FrontPage XP (web based module) and MS Access 97(database). The Active Server Page (ASP) technology was used for database interaction. There are four main modules involved, namely as the Database Entry module, Neural Networks module, Classification module and Reporting module.

A. DATABASE ENTRY MODULE

The financial ratios and macro economics parameter data were used as input data. The input parameters were chosen based on several references and expert advices [1][2][8][11]. These variables include the following: (see Figure 2.0):

- Working Capital/Total Assets
- Retained Earning/Total Assets
- Earning before Income Tax/Total Assets
- Total Sales /Total Assets
- Total Debt / Total Assets
- Inflation Rate
- Inter-Bank Rate
- Operational Size
- National Gross Domestic Product
- Firm's Age
- Types of Industry

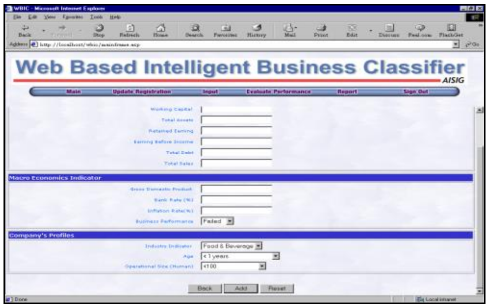

Figure 2.0: The Database Entry Module

The data is then converted to other pre-processing values through symbolization and binarization processes. These values are normalized using *Linear Scaling* method that can be written as ([10]):.

$$\bar{x} = \left(X_{\min} + (X_{\max} - X_{\min}) * \frac{D - D_{\min}}{D_{\max} - D_{\min}} \right) \quad (5)$$

where *D* is the original dataset and *X* is a pre-processed value. The data comprises of 350 datasets, which covers 150 insolvency cases and 200 healthy firms.

B. NEURAL NETWORKS MODULE

The neural networks module is the core module in the prototype. It utilizes the state of the art technology of the AI to mimic human expert for classifying business insolvency cases. This module implements MLP Backpropagation architecture with eleven input nodes, six hidden nodes and one output node. The initial weights are seeded either using *Nguyen-Widrow* normalization. The *Nguyen-Widrow* normalization algorithm is summarized as:

Start:

1.Calculate β, a scale factor for *n* input nodes and p hidden nodes $\beta = 0.7^n \sqrt{p}$

2. Set random initialization weights value to v_{ij}

3. Compute Euclidean length of j^{th} column of v, $|v_j|$

4. Update all the weights, $v_{ij} = \dfrac{\beta v_{ij}(old)}{|v_j|}$

5. Use a bias input, $v_{oj} = random(-\beta, \beta)$

End:

Figure 3.0 shows the GUI of neural networks module.

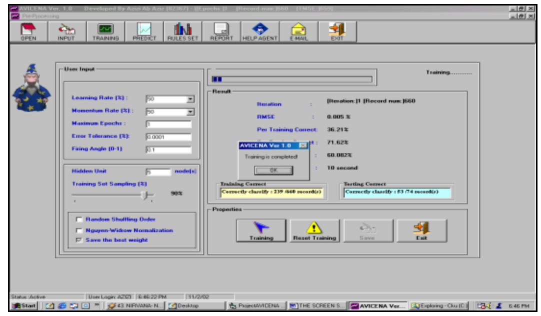

Figure 3.0: Neural Networks Module

The data is separated using ten-folds cross validation techniques. This method was carried out to dampen fluctuation due to random partitioning of the data [11]. The patterns used for training were shuffled randomly to avoid memorization and saturation in the learning process [12].

C. CLASSIFICATION MODULE

The Classification module allows the users to classify the new cases based on the model generated in the neural networks module. Figure 4.0 shows the screen snapshot of the module.

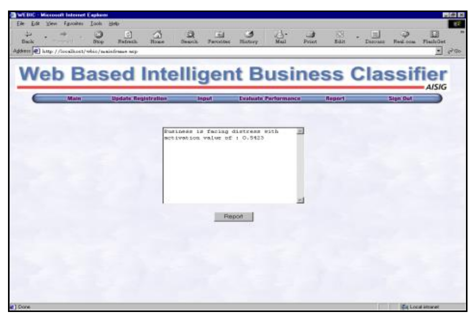

Figure 4.0: Screen Snapshot for Classification Module

The Classification module consists of three main components, namely the pre-processing and transformation; business classifier model and post processing (refer to Figure 5.0).

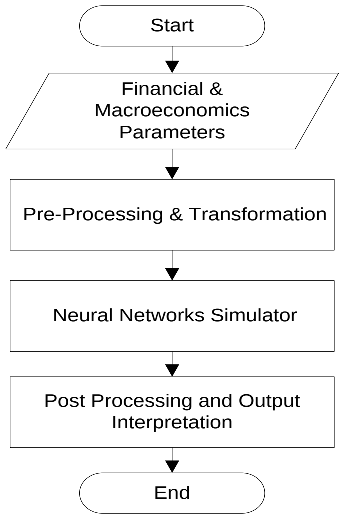

Figure 5.0: Processing Steps in Classification Module

D. REPORTING MODULE

The reporting module provides report to the user. In this module, the reports can be classified into two groups, namely as corporate profiles and suggested classification result. Corporate profiles are concerned with historical data that has been entered into the prototype. It consists of information about corporate prospects and past performances. The suggested classification result on the other hand provides the results based on the corporate performance suggested by the classification module.

IV. Results

Several experiments have been conducted to determine the parameters that influence the performance of MLP. The experiments were conducted to find suitable values for momentum terms, learning rate and number hidden units. In MLP, the experiment typically starts with training set and use back propagation algorithm to compute synaptic weights by loading as many training examples into the network for generalization purpose.

Generalization is acquired when an input output relationship computed during training process is correct or nearly so) for unknown test data [13]. Training parameters are dependent on the problem and MLP configuration. For example, learning rate and momentum term that are suitable for a network with one hidden layer may not be suitable for network with two hidden layers.

In this study, the initial values for hidden unit started from three hidden units as suggested by Bigus (see [6]), where

$$\textit{hidden unit} = 2n + 1, \text{ n is output node} \qquad (6)$$

From Table 1.0, a network with 6 hidden units produces the highest accuracy. For this reason, the learning rate and momentum term were fixed to 0.1 and 0.5 respectively.

Table 1.0: Results Using Various Numbers of Hidden Units

Hidden Unit	Training Set	Testing Set
3	76.54 %	77.50%
4	79.67%	79.13%
6	**85.57%**	**88.77%**
8	82.55%	82.53%
10	81.66%	80.44%
12	83.50%	79.18%
14	82.15%	79.15%
16	85.12%	72.23%
18	84.17%	72.12%
20	83.52%	71.19%

The results exhibited in Table 2.0 indicate that the highest test result was achieved by a network structure that has a learning rate of 0.1 and momentum term of 0.5 (see Table 2.0). Another evaluation factor is the influence of epochs compared to the mean square error (MSE).

Table 2.0: Experiments Results from Various Learning Rate (η) and Momentum Term (α)

η	α	Epochs	Training	Testing
0.025	0.5	1000	83.62%	85.45%
0.05	0.9	800	82.55%	79.50%
0.1	0.9	750	85.00%	87.33%
0.25	0.5	740	78.56%	81.45%
0.5	0.75	710	72.25%	76.00%
0.25	0.9	687	76.34%	76.24%
0.1	**0.5**	**693**	**87.43%**	**90.45%**
0.05	0.5	920	85.45%	88.33%
0.025	0.9	984	84.55%	81.33%
0.1	0.75	840	86.25%	87.00%

The relationship between the number of epochs and the mean square error is shown in Figure 6.0. It shows that after 900 epochs, the MSE rate remain constant (0.014 percent). The learning ability became saturated when the epoch reached more than 900 learning cycles.

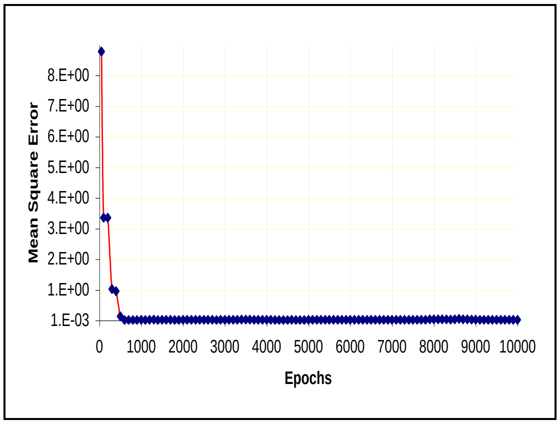

Figure 6.0: Relationship between Epoch and Mean Square Error

Once the learning parameters were obtained and fixed, ten folds cross validation took place. This method was implemented in order to avoid overlapping in training and testing sets that might occur during *holdout* splitting procedure. There are two main phases in this process. First, the data is split into *k* subsets (k=10) of equal sizes. In the second phase, each subset is used for training and the rest for testing. The validation results are shown in Table 3.0.

Table 3.0: Ten Folds Validation Results

Folds	**Training Accuracy**	**Testing Accuracy**
Fold 1	79.55%	83.33%
Fold 2	83.45%	85.55%
Fold 3	76.00%	78.55%
Fold 4	84.55%	79.33%
Fold 5	86.82%	88.55%
Fold 6	78.86%	79.65%
Fold 7	**85.55%**	**90.25%**
Fold 8	80.25%	83.33%
Fold 9	87.35%	85.33%
Fold 10	79.85%	83.55%
Average	**82.23%**	**83.74%**

The data was tested using 11-6-1 neural network architecture with learning rate 0.1, momentum term 0.5, maximum epoch 1500 and error tolerance 0.001. Table 4.0 summarizes the final neural networks architecture for business insolvency model. These configurations were implemented in web enabled format.

Table 4.0: The Selected Neural Networks Configuration

Properties	**Results**
Dataset	350
Maximum Epochs	1000 iterations
Network Topology	11-6-1
Learning Rate	0.1
Momentum Term	0.5
Error Rate	0.00126
Testing Accuracy	90.25%
Weight Initialization	Nguyen-Widrow

V. Conclusion and Recommendations

Currently WEBIC has been designed and tested on real data from the Register of Business, Register of Companies, Kuala Lumpur Stock Exchange and Bank Negara Malaysia (Central Bank of Malaysia). Further improvement can be made to enhance the user interface design, particularly in explaining the result produced by the prototype. The practical improvement for the system are listed below:

- Implementation of an automated adaptive tuning for learning rate and momentum rate in order to increase processing speed.
- Further study on the classification engine by optimizing the neural networks architecture with another techniques (such as Neuro-Fuzzy, Fuzzy ARTMAP and Kohonen Self Organizing Map).
- Further study on financial and economics data parameters in term of global features that brought into business insolvency point of view (Chaos Theory).

Clearly, the prototype provides basic framework for developing web based intelligent business insolvency classifier. Hopefully, documentations of this research would serve as a practical guide for academicians to continue based on the findings of this research or for practitioners to dynamically employ and empower this idea at their respective workstations.

ACKNOWLEDGEMENT

The authors would like to thank Universiti Utara Malayisa for providing the research grant (S.O: 10891) to carry out this research.

REFERENCES

[1] Aziz, A., and Lawson. (1999). *Bankruptcy Prediction: An Investigation of Cash Flow Based Models*, Journal of Management Studies 25(5), pp. 19-37.

[2] Altman E. (1993). *Corporate Financial Distress and Bankruptcy 2nd* , John Wiley, New York.

[3] Cybinski, P. (2000). *The Evolution of Bankruptcy Studies: From Prediction to Explanation*, Journal of Applied Business Research Vol. 33 (Summer), pp. 60-71.

[4] Luger, G.F. (2002). Artificial Intelligence: Structures and Strategies for Complex Problem Solving (IV Edition), Addison-Willey, pp. 24.

[5] Goonatilake, S. (1995). *Intelligent Hybrid Systems for Financial Decision Making*, ACM Journal of Applied Computing, United State, pp. 55-57.

[6] Haykin, S. (1994). *Neural Networks: A Comprehensive Foundation*, Springer, New York.

[7] Agrawal, R. and Srikant, R. (1994). *Fast Algorithms for Mining Association*,Proceedings of the 20th International Conference on Very Large Database, Santiago, Chile.

[8] Caillie, D. (1999). *Business Failure Prediction Models: What Is The theory Looking For?*, 2th Conference of Risk Management, pp. 22-31.

[9] Dizdarevic, S., Larrañaga, P., Sierra, B., and Peña, J.M. (1999). *Combining Statistical and Machine Learning Based Classifiers In The Prediction of Corporate Failure*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24. 1. January, pp. 65-75.

[10] Bigus, J. B. (1996). *Data Mining with Neural Networks.* Mac Grawhil, New York.

[11] Abu Bakar, A., Sulaiman, M.N and Selamat, M.H. (2001). An Improved Rough Classification Model: A Comparison with Neural Classifier. Journal of Institute of Mathematics and Computer Sciences (Vol. 12, Num.1), pp.43.

[12] Chua, B,L., Khalid, M. and Yusof, R. (2000). *An Enhanced Intelligent Database Engine by Neural Networks and Data Mining*, TENCON 2000 Proceedings Vol. II, pp. 518-522.

[13] Gonzalez, R., and Dankel, J., (1993). F*undamental Knowledge Based System Architecture*, Prentice-Hall International Edition, pp. 45-55.