# A Neural Networks System for Predicting Islamic Banks Performance

MAHMOUD H. AL-OSAIMY
*Assistant Professor,*
*Department of Economics,*
*Faculty of Economics and Administration*
*King Abdul Aziz University, Jeddah, Saudi Arabia*

( Received 27-1-1418H and Accepted 16-5-1418H )

**ABSTRACT.** The objective of this research is to use neural networks for predicting Islamic banks performance. A data sample of twenty six Islamic banks has been collected for the period 1991 - 1993. Seven financial ratios were constructed from the data sample. Kohonen neural network was used first to group the Islamic banks into high and low performance groups using the seven financial ratios for the performance year (1993). The results of this network have assigned twelve banks to the high performance group and fourteen banks to the low performance group. Using these results, a three layers Backpropagation neural network was built to predict the Islamic banks performance. The network was trained on the one year prior to the performance year data set while using the performance year data as a test set. The two year prior to the performance year data set was used as a holdout sample to check the network prediction abilities. When the network was applied, the results showed high prediction abilities and generalization. It correctly classified all the Islamic banks on the performance year and on the one year prior data sets. It also correctly classified 89% of the banks on the two years prior data set which was unseen by the network.

## I. Introduction

In the last few years, neural networks have received a great deal of attention and were subject of research activity and practical applications across a range of fields. One area of application is the business and financial management which includes, and not limited to, company and bank failures, stock price predictions, and bonds rating. In the area of predicting commercial bank failures, Bell et al. (1990), Tam (1991), and Tam and Kiang (1992) have used neural networks for prediction of bankruptcy in the banking industry. For predicting company bankruptcy, the neural networks have also been used by Odom and Sharda (1990), Raghupathi et al. (1991), Koster et al. (1990), Coats and Fant (1993), Tsukuda and Baba (1994), and Altman et al. (1994). In the assignment of ratings to bonds, Dutta and Shekhar (1988) and Surkan and Singleton (1990) applied neural networks to predicting ratings of corporate bonds and their expected profitability. The neural networks also were applied by Yoon and Swales (1990) to predict stock price performance, and by White (1988), Kamijo and Tanigawa (1990), and Kryzanwski et al. (1993) for the prediction of common stock returns. Other applications of neural networks in finance were provided in Trippi and Turban (1992).

In all of these studies, the neural networks model outperformed the traditional statistical methods such as Discriminant analysis and Logistic regression. In addition, the most commonly approach used in the above and other studies was the backpropagation neural networks models, with the exception of Coats and Fant (1993), and Altman et al. (1994) which used cascade correlation neural network.

The present study aims to construct a neural networks model to predict the performance of the Islamic banks based on seven financial ratios. Kohonen neural network will be used first to group the Islamic banks into high and low performance groups using the performance year data set. The Backpropagation neural network will be used then to predict the Islamic banks performance using the one year prior data set. The predictive ability of our model will be checked by using the two years prior data set, which is an unseen data by the network. The rest of the paper is organized as follows. Section 2 provides an overview of neural networks. The data sample and variables constructed are described in section 3. The networks design and results are presented in section 4. The last section will be the conclusion.

## II. Overview of Neural Networks

Neural networks technology mimics the brain’s own problem solving process. Just as humans apply knowledge gained from past experience to new problems or situations, a neural network takes previously solved examples to build a system of “neurons” that makes new decisions, classifications, and forecasts.

Neural networks look for patterns in training sets of data, learn these patterns, and develop the ability to correctly classify new patterns or to make forecasts and predictions. Neural networks excel at problem diagnosis, decision making, prediction, and other classifying problems where pattern recognition is important and precise computational answers are not required.

### *2.1. The General Neural Networks Structure*

Although individual models differ, all neural networks possess similar fundamental features. The basic building block of neural networks technology is the single neural processing unit called the neuron (depicted in Figure 1 as a circle). Independent neurons are of little use, however, unless they are interconnected in a network of neurons. The network processes a number of inputs from the outside world to produce an output, the network’s classifications or predictions. The neurons are connected by weights, (depicted as lines) which are applied to values passed from one neuron to the next.

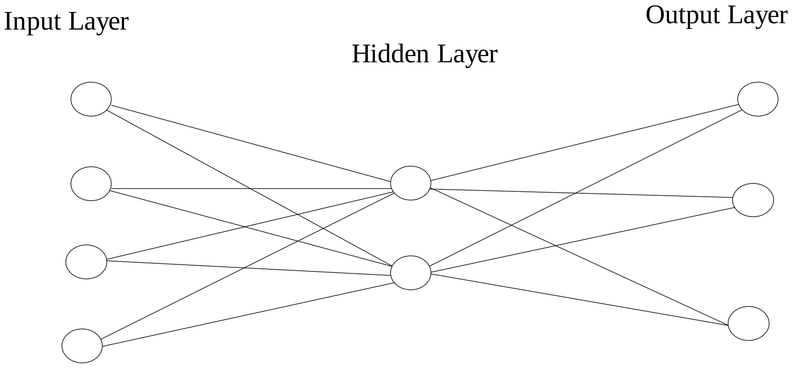


Figure (1) General Neural Network Structure

A group of neurons is called a slab. Neurons are also grouped into layers by their connection to the outside world. For example, if a neuron receives data from outside of the network, it is considered to be in the input layer. If a neuron contains the network's predictions or classifications, it is in the output layer. Neurons in between the input and output layers are in the hidden layer(s). A layer may contain one or more slabs of neurons.

The mathematical representation of the neural network models can be described as follows: a single processing unit (neuron) can be regarded as non-linear function which transforms a set of input variables $x_i$ (i = 1, ....,n) into an output variable z. The signal $x_i$ at input i is first multiplied by a parameter $w_i$ known as weight (coefficient) and is then added to all other weighted input signals to give a total input to the processing unit in the form :

$$a = \sum_{i=1}^{n} w_i x_i + w_0$$

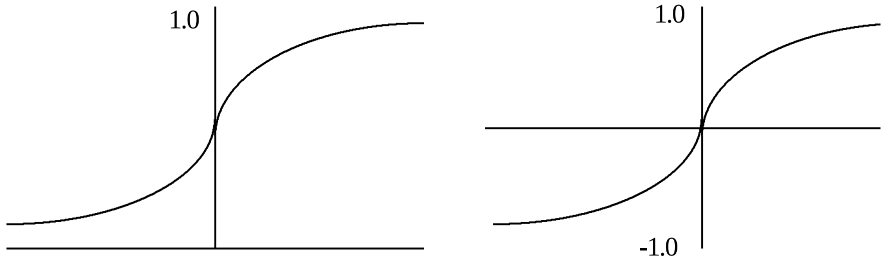
where the parameter $w_0$ is called a bias (corresponds to the intercept coefficient in the regression). Formally, the bias can be regarded as a special case of weight from an extra input whose value $x_0$ is permanently set to +1 . Thus we can write the aggregate input to the processing unit in the form :

$$a = \sum_{i=0}^{n} w_i x_i$$

where $x_0 = 1$. The output z of the processing unit is then given by transforming the total input with a non-linear activation function g (also called squashing function) so that :

$$z = g(a) = g\left( \sum_{i=0}^{n} w_i x_i \right)$$

where g is the activation function which is often used as a sigmoidal function (S-shaped) such as the logistic function and the tangent hyperbolic function. The shape and formula of these functions are shown in Figure (2).



Logistic function $g = \frac{1}{1+e^{-a}}$ Tangent hyperbolic function $g = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$

Figure (2) Sigmoidal Activation Functions

The single processing unit described above, and illustrated in Figure (3), forms the basic element in many artificial neural networks models. By linking together many such processing units it is possible to construct a very general class of multilayer neural network model. A three layers network is commonly used in most practical applications. This network, as shown in Figure (4), consists of input layer, hidden layer, and output layer. The input layer has a set of n input variables, $x_i$ (i = 1,2,...,n) and a bias $x_0 = 1$. The hidden layer has a set of m processing units, $z_j$ (j = 1,2,...,m) and a bias $z_0 = 1$. Each hidden processing unit output $z_j$ is given by :

$$z_j = g\left(\sum_{i=0}^{n} w_{ji}\, x_i\right) \qquad j = 1,2,.......,m \qquad (1)$$

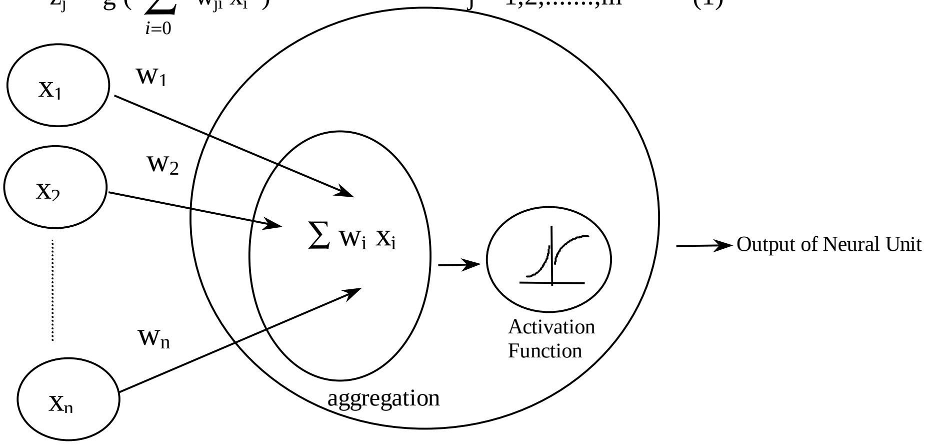


Figure (3) Single Neural Processing Unit

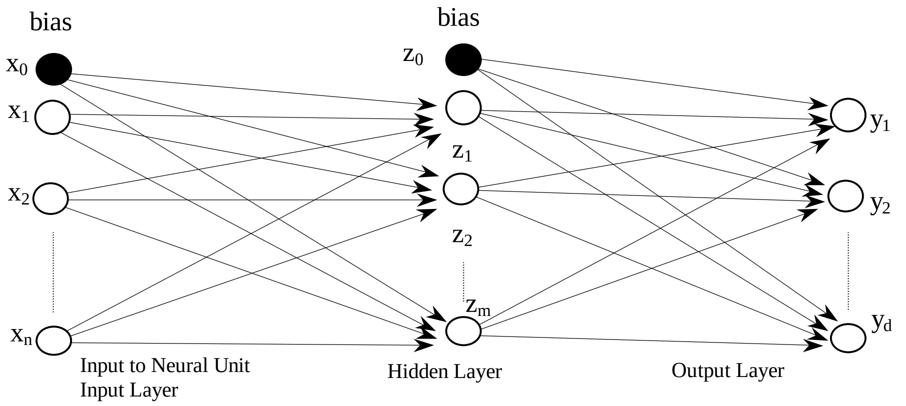


Figure (4) Three Layers Neural Network

where $w_{ji}$ is the weight from input i to the hidden processing unit j, and g is the activation function for the hidden layer. The output layer of the network receives signals from the hidden layer and acting a second transformation to produce a set of d output variables, $y_k$ (k = 1,2,...d). Each output processing unit variable $y_k$ is given by :

$$y_k = \bar{g}\ (\ \sum_{j=0}^{m} \bar{W}_{kj}\ z_j\ ) \qquad k = 1,2,.......,d \qquad (2)$$

where $\bar{W}_{kj}$ is the weight from the hidden processing unit j to the output processing unit k, and $\bar{g}$ is the activation function (Sigmoidal function) applied to the output processing units and it dose not have to be the same as the activation function g used for the hidden units. Equations (1) and (2) can be combined to give the complete expression for the transformation represented by the network in the form :

$$y_k = \bar{g}\ [\ \sum_{j=0}^{m} \bar{W}_{kj}\ g\ (\ \sum_{i=0}^{n} w_{ji}\ x_i\ )\ ]\ .$$

### *2.2. Neural Network Training*

The problem of determining the values for the weights in a neural network is called ‘training’. The network training is performed by seeking a set of values for the weights which minimize some error function, often chosen to be the sum of square errors. There are various algorithms that are used for training which depend on the type of the network and its architecture. The most popular training algorithm which has been used extensively in practical applications is the backpropagation algorithm. This algorithm can be viewed as a gradient search technique where the objective function is to minimize the mean square error between the computed outputs of the network, $y_k$ , the corresponding to the given set of inputs in a multilayer feed forward network and the actual outputs observed in the data set, $y_k$ , for the same given input. The error function takes the form :

$$E = \frac{1}{2}\sum_{t=1}^{T}\sum_{k=1}^{d}(y_k^t - \hat{y}_k^t)^2 \qquad t = 1,2,...,T \quad , \ k = 1,2,...,d$$

where E is the average of the squares of the individual errors summed over all outputs, d, and overall patterns (data set), T. The network is trained by presenting an input pattern vector X to the network, performing the calculations sequentially through the network until an output vector Y is obtained. The output error is computed by comparing the computed output Y with the actual output Y . The network learns by adjusting the weight at each individual neural processing unit to minimize the error function. That is, after each training pattern, the predicted errors are swept backward (backpropagated) through the network, layer by layer, updating (adjusting) the weight of each processing unit. This process is repeated for each pattern until the training set is exhausted. The algorithm starts over again on the training set and readjusts the weights throughout the entire network structure until either the objective function (the sum of squared prediction errors on the training sample) is sufficiently close to zero or the default number of iterations is reached.

### *2.3. Types of Neural Networks*

There are two basic types of neural networks: supervised and unsupervised**:**

1. Supervised networks build models which classify patterns, make predictions or make decisions according to other patterns of inputs and outputs they have “learned”, They give the most reasonable answer based upon the variety of learned patterns. In a supervised network, you show the network how to make predictions, classifications, or decisions by giving it a large number of correct classifications or predictions from which it can learn. Backpropagation networks are supervised network types.

2. Unsupervised networks can classify a set of training patterns into a specified number of categories without being shown in advance how to categorize. The network does this by clustering patterns. It clusters them by their proximity in an N dimensional space where N is the number of inputs. The user tells the network the maximum number of categories and it usually clusters the data into that number of categories. However, occasionally the network may not be able to separate the patterns into that many distinct categories. Kohonen networks are unsupervised.

## III. Data Sample and Variables Constructed

Data have been collected for twenty six Islamic banks from the annual reports during the period 1991-1993. These data will be used to formulate performance variables which are used to predict the performance of the sample banks in the year 1993 using data (financial ratios) for the previous year 1992. The data for the year 1991 will be used as a holdout sample to check the prediction abilities of the model.

Twenty six banks with enough operation experience were selected in the sample, namely those which operated before 1989. Such a relatively small number restricted the size of observations (number of banks) in each group of performance. This had obliged us to compress the number of variables and aggregate them into block variables which in our belief would at most represent the operational factors affecting the performance levels of the studied banks. Another reason which necessiated aggregating variables is the unstandardized items in the balance sheets and income (profit and loss) statements of Islamic banks. Only some banks reported fully detailed assets, liabilities and income entries. Worse than that, some banks treated investments belonging to customers (depositors) as off-balance sheet items. Different reporting methods make disaggregate and standardized classification of data difficult

Seven financial ratios were used in this research. These ratios are shown in Table (1), along with their definitions.

Table (1)
Variables Definitions

$X_1$ = Total Income / Total Assets, representing productivity of bank resources (resource utilization or asset turnover).
$X_2$ = Investment Income / Total Income, representing the level of contribution of income coming from investments to total income.
$X_3$ = Total Income / General and Administrative Expenses, representing the operational efficiency of the bank.
$X_4$ = Provisions for Bad Debts and Investments / Total Assets, representing financing and/or investment risk.
$X_5$ = Cash / Total Deposits, representing the liquidity position in the bank.
$X_6$ = Customers Investment Deposits / Shareholders Equity, representing the leverage level (debt/equity ratio) in the bank's investment.
$X_7$ = Net Profit Before Zakat and/or Taxes / Total Assets, representing the profit rate of the bank.

## IV. Design of The Neural Networks Models

This research will use two neural network models, namely, Kohonen and Backpropagation neural networks. The analysis well be conducted in the following manner. First we will design the Kohonen neural network which will be used to group the Islamic banks into high performance and low performance groups. Second the Backpropagation neural network will be designed to predict the Islamic banks performance. These steps are as follows:

### *4.1. Kohonen Neural Network*

The Kohonen network, also known as Self -Organization network, is a type of unsupervised network, which has the ability to learn without being shown correct outputs in sample patterns. These networks are able to separate data into a specified number of categories.

This research will be using the Kohonen neural network to group the twenty six Islamic banks into high and low performance levels according to the performance financial ratios.

The Kohonen network consist of only two layers: an input layer and an output layer. Each neuron in the input layer is connected to each neuron in the output layer by a connection with associated weight. The input layer’s neurons consist of the seven financial ratios being used to separate Islamic banks into high and low performance groups. The output layer has two neurons, high performance level and low performance level. The architecture of the Kohonen network being used in this study is depicted in Figure (5), where the arrows represent the connection weight associated with each input.

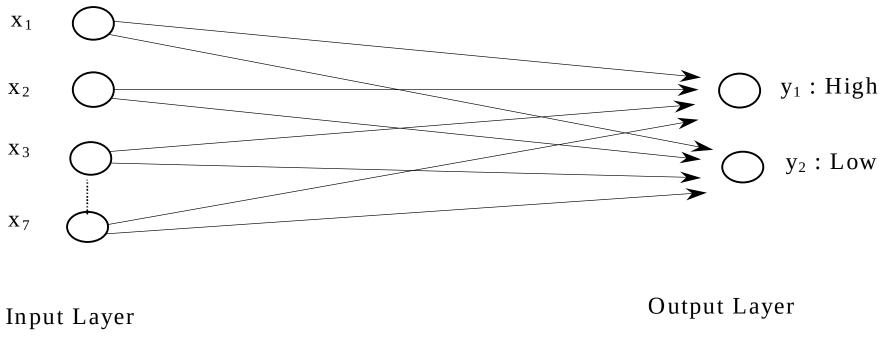


Figure (5) Kohonen Neural Network

### *4.1.1. Training*

The training process consists of the following: The training patterns are presented to the input layer, then propagated to the output layer and evaluated. One output neuron is the "winner". The network weights are adjusted during training. This process is repeated for all patterns for a number of epochs chosen in advance. The network adjusts the weights for the neurons in a neighborhood around the winning neuron. This means that the training for unsupervised Kohonen networks requires that the weights leading to the winning neuron are adjusted after each pattern passes through the network. However, the winning neuron needs to have a positive influence on the neurons which physically surround it, i.e., its neuron neighbors. Therefore, the weights for the neurons in the neighborhood around the winning neuron are changed during training, too. The neighborhood size is variable, starting off fairly large, sometimes even close to the number of categories, and decreasing with learning until during the last training event the neighborhood is zero, meaning, by then, that only the winning neuron's weights are changed. By that time the learning rate is very small, and the clusters have been defined.

The common approach to initialize the network weights is to assign random weight values between 0.4 and 0.6. The average of this range was taken and assigned the initial value of the weight to 0.5. The learning rate was initialized to 0.5, which was considered to be a high number. The training was started with relatively high neighborhood size such as 90% of the number of neurons in the output layer. Finally, training the network for 200 epochs (iterations) was chosen.

### 4.1.2 The Results

The Kohonen network was trained for 200 epochs using the seven financial ratios to group the Islamic banks into high and low performance levels. At the end of the training, the learning rate decreased to 0.001 and the neighborhood size reached zero. Setting the winning neuron to 1, and all the others to 0, the network assigned 12 Islamic banks to the high performance group and 14 Islamic banks to the low performance group.

### *4.2. Backpropagation Network*

A typical three-layers Backpropagation neural network was used. It consists of an input layer which has seven input variables (the seven financial ratios; $X_1$ , $X_2$ , ... , $X_7$), a hidden layer which has nine neurons, and an output which has one neuron. Figure (6) depicts the network architecture adopted.

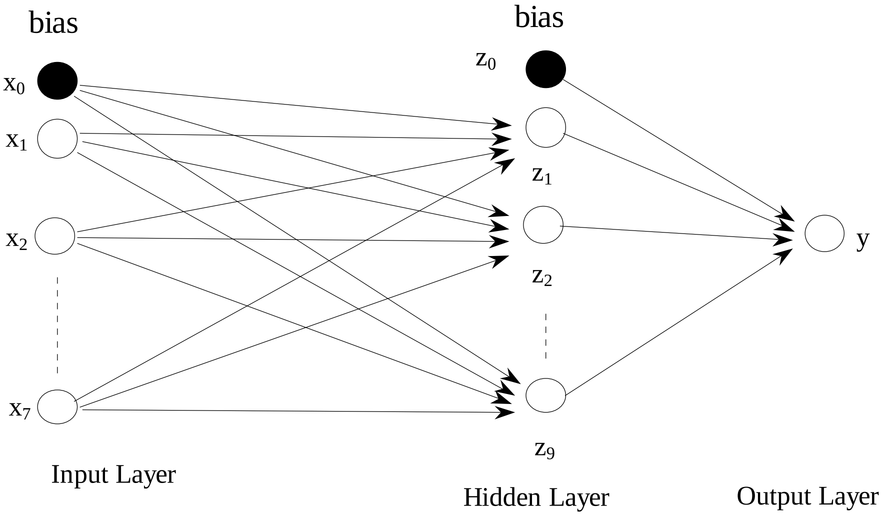


Figure (6) Backpropagation Network Architecture Used for Modeling Islamic Banks performances

The number of neurons in the input layer is determined by the number of variables being used in this study. Since the study uses seven financial ratios, determining the performance of Islamic banks, the input layer contains seven neurons. The output layer has one neuron which is the performance level (high/low) of Islamic banks. Deciding the number of neurons in the hidden layer is difficult. Too few neurons in the hidden layer prevent the network from generalizing well in the training session, while too many neurons complicate training and make the errors more difficult to trace. Following the software manual (Neuro Shell 2) recommendations, the number of neurons in the hidden layer is calculated according to the formula:

Number of hidden neurons = 1/2 (Inputs + Outputs) + Sqrt (Number of Banks)
= 1/2 (7 + 1) + Sqrt(26) = 9 ,

where Sqrt means square root. Thus, we used nine neurons in the hidden layer since we have seven financial ratios, one output, and 26 Islamic banks.

A linear activation function was used in the input layer to normalize the data into (0 to 1) range. The logistic activation functions are used in the hidden layer and the output layer to squash the results into (0 to 1) range.

#### *4.2.1. Training*

The training of the network was carried out using the Backpropagation algorithm described above. In the training session, the biggest secret of neural network models is when to stop training. If there are too few training iterations, the net will not learn the patterns. If there are too many training iterations, the net will learn the noise or memorize the training patterns and not generalize well with the new patterns. Fortunately, the software used here (NeuroShell 2) contains a Calibration technique which prevents overtraining the network.

Calibration trains and computes an average error factor during the training process. For every specific interval, it reads the test set and computes an average error. We use the default setting for the test set error computation interval of 200 training patterns. What usually happens is that the error for the training set continues to get smaller forever, or at least gets to the point where it is fairly flat. The error for the test set continues to get smaller to a point, the optimal point, and then slowly begins to get larger. Calibration saves the network at this optimal point.

In order to use the Calibration and due to limitation of our data sample, we use the data for the one year prior to the performance year as a training set and the data for the performance year as a test set. We used Calibration to save the network at the optimal lowest error occurred in the test set. The training stops when the number of events reaches 20,000 events after the minimum error has been reached for the test set. The minimum average errors in the training process was 0.005 on the test set data and 0.002 on the training set data.

### *4.2.2. The Results*

The results of applying the neural network models to predict Islamic banks performance based on the seven financial ratios show in general very good abilities to classify high and low performance groups. The output of the network has real values from 0 to 1, which were classified as high performance when ≥ 0.5 and as low performance when < 0.5 .

The trained network was applied to the training data set and showed that it correctly classified all Islamic banks with mean square error (MSE) of prediction 0.003. The network was also applied to the performance year data set and to the two years prior data set for checking the network prediction abilities. The network also showed high predictions abilities in both data sets. It correctly classified 100% of Islamic banks on the performance year data set and 89% on the two years prior data set with MSE 0.006 and 0.072 respectively. The results are shown in Table 2.

The interconnection weights between neurons resulting from the training session are presented in Table (3). The link 1 represents the weight values from the input layer processing units (displayed across the top of the table) to the hidden layer processing units (displayed down the side of the table). The link 2 represents the weight values from the hidden layer processing units to the output processing unit.

The Backpropagation network produces a number for each input variable called a contribution factor that is a rough measure of the importance of that variable in predicting the network's output, relative to the other input variables in the same network. The higher the number, the more the variable is contributing to the classification. The contribution factor for each variable, resulting from the training session are presented in Table (4) along with their ranks.

Table (2)
Classification Results Based on One Year Prior Trained Backpropagation Neural Network for the Performance Year, One Year Prior and Two Years Prior.

| 1. The Performance Year (1993) : | | | | | |
|---|---|---|---|---|---|
| Performance Group | No of cases. | Correct Classifications | % | Misclass-ifications | % |
| High | 12 | 12 | 100 | 0 | 0 |
| Low | 14 | 14 | 100 | 0 | 0 |
| Overall | 26 | 26 | 100 | 0 | 0 |
| 2. One year prior (1992) : | | | | | |
| High | 12 | 12 | 100 | 0 | 0 |
| Low | 14 | 14 | 100 | 0 | 0 |
| Overall | 26 | 26 | 100 | 0 | 0 |
| 3. Two years prior (1991) : | | | | | |
| High | 12 | 12 | 100 | 0 | 0 |
| Low | 14 | 11 | 78.6 | 3 | 21.4 |
| Overall | 26 | 23 | 88.5 | 3 | 11.5 |

Table (3)
The Weight Values Between Processing Units in Backpropagation Network Trained on the Data Set for One Year Prior to the Performance Year.

| Link 1 : From input layer to hidden layer : | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| From To | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X7 | | |
| Z1 | -0.22838 | 0.01877 | -0.33975 | -0.20232 | 0.15474 | -0.05680 | 0.05758 | -0.28808 | | |
| Z2 | 0.12952 | -0.54704 | -0.41781 | -1.07021 | 0.54514 | 1.42531 | -0.05475 | -0.26120 | | |
| Z3 | 0.43388 | 0.32345 | 0.59333 | 0.80709 | 0.77662 | -1.32789 | 1.39505 | 0.48293 | | |
| Z4 | -0.72498 | -0.08510 | -0.50955 | -0.94042 | -1.04072 | 0.34142 | -1.79776 | -1.01619 | | |
| Z5 | -1.42301 | 1.48715 | 3.41505 | 6.87847 | 3.21213 | -11.7464 | 7.10676 | 1.01022 | | |
| Z6 | 0.19431 | 0.15797 | -0.11141 | 1.25951 | -0.08972 | -1.59702 | 0.15551 | 0.00492 | | |
| Z7 | 0.95342 | -1.50849 | -2.54763 | -3.49719 | -1.91986 | 4.95898 | -3.38102 | -1.52879 | | |
| Z8 | -0.77592 | -0.89925 | -0.99877 | -1.53555 | -0.95059 | 0.56466 | -2.09916 | -0.81536 | | |
| Z9 | 0.26548 | 0.12188 | 0.13217 | 1.09474 | -0.21054 | -1.12290 | 0.37947 | -1.32200 | | |
| Link 2 : From hidden layer to output layer : | | | | | | | | | | |
| From To | Z0 | Z1 | Z2 | Z3 | Z4 | Z5 | Z6 | Z7 | Z8 | Z9 |
| Y | -0.65468 | -0.07116 | 1.80301 | -1.28254 | -0.24166 | 6.88638 | -2.30613 | -0.65649 | -0.26357 | -1.96106 |

Table (4)
Relative Contribution Factors and Ranks of the Individual Variables Resulting from Backpropagatoin Neural Network Trained on One Year Prior Data Set.

| Variable | Contribution Factor | Rank |
|---|---|---|
| $X_1$ | 0.056004 | 6 |
| $X_2$ | 0.084398 | 5 |
| $X_3$ | 0.243542 | 2 |
| $X_4$ | 0.089381 | 4 |
| $X_5$ | 0.337778 | 1 |
| $X_6$ | 0.152098 | 3 |
| X7 | 0.036795 | 7 |

The contribution factor resulting from the network shows that the liquidity variable ($X_5$), and the efficiency variable ($X_3$) ranked first and second respectively in importance as to the classification of the Islamic banks performance. Leverage and risk levels ($X_6$ and $X_4$) ranked third and forth respectively. The profitability variable is the least important.

## V. Conclusions

The use of Neural Networks in predicting Islamic banks performance was explored by this research. This step is needed to establish an early warning system for the Islamic banks performance, especially those banks which are newly established. The research used the data for twenty six Islamic banks taken from the annual reports. Seven financial ratios were constructed and used to discriminate between high and low performance groups. The Backpropagation neural network model based on the Kohonen neural network grouping was used for predicting Islamic banks performance. The results show that the network provides high prediction abilities on the training data set (1992 data set), and on the holdout sample (1991 data set).

Even with the limited data sample used in this research, neural networks methods show significant promise for providing useful early warning signals for various types of performance including bankruptcy, insolvency and failure.

In spite of the positive results obtained, further investigation is needed. More data samples should be collected and disaggregate financial ratios should be included.

## References

**Altman, E., G. Marco** and **F. Varetto,** (1994), Corporate Distress Diagnosis: Comparisons Using Linear Discriminant Analysis and Neural Networks (the Italian Experience), *Journal of Banking and Finance*, Vol. **18**, pp. 505-529.

**Altman, E., R. Avery, R. Eisenbeis** and **J. Sinkey** (1981), *Application of Classification Techniques in Business, Banking and Finance*, Greenwich, Connecticut, Jai Press Inc.

**Bell, T., G. Ribar** and **J. Verchio,** (1990), Neural Nets vs. Logistic Regression: A Comparison of Each Model's Ability to Predict Commercial Bank Failures, in: **P. Srivastava**, (ed)., Auditing Symposium X, University of Kansas.

**Bishop, C.,** (1994), Neural Networks and Their Applications, *Review of Scientific Instruments*, Vol. **65**, No. **6**, pp. 1803-1832.

**Coats, P.** and **L. Fant,** (1993), Recognizing Financial Distress Patterns Using a Neural Network Tool, *Financial Management*, Vol. **22**, No. **3**, pp. 142-155.

**Dutta, S.** and **S. Shekhar,** (1988), Bond Rating: A Non-Conservative Application of Neural Networks, *Proceedings of IEEE International Conference on Neural Networks*, IEEE Press, Alamitos, CA, Vol. **2**, pp. 443-450.

**Eberhart, R.** and **R. Dobbins,** (1990), Neural Network PC Tools: A Practical Guide, Academic Press, New York.

**Hertz, J., A. Krogh** and **R. Palmer,** (1991), Introduction to Theory of Neural Computation, Addison Wesley, Redwood City, CA.

**Johnsen, T.** and **R. Melicher,** (1994), Predicting Corporate Bankruptcy and Financial Distress: Information Value Added by Multinomial Logit Models, *Journal of Economics and Business*, Vol. **46**, No. **4**, pp. 269-286.

**Kamijo, K.** and **T. Tanigawa,** (1990), Stock Price Pattern Recognition: A Recurrent Neural Network Approach, *Proceedings of IEEE International Conference on Neural Networks*, IEEE Press, Alamitos, CA, Vol. **1**, pp. 215-221.

**Koster, A., N. Sondak** and **W. Bourbia,** (1990), A Business Application of Artificial Neural Network System, *The Journal of Computer Information Systems*, Vol. **XI**, pp. 3-10.

**Kryzanwski, L., M. Galler** and **D. Wright,** (1993), Using Artificial Neural Networks to Pick Stock, *Financial Analysis Journal*, Vol. **12**, No. **1**, pp. 21-27.

**NeuroShell 2,** (1993), *User's Manual*, Ward Systems Group Inc., MD.

**Odom, M.** and **R. Sharda,** (1990), A Neural Network Model for Bankruptcy Prediction, *Proceedings of IEEE International Conference on Neural Networks*, IEEE Press, Alamitos, CA, Vol. **2**, pp. 163-168.

**Raghupathi, W., L. Schkade** and **R. Bapi,** (1991), A Neural Network Application for Bankruptcy Prediction, *Proceedings of the 24th Hawaii International Conference on System Sciences*, IEEE Press, Alamitos, CA, Vol. **4**, pp. 147-155.

**Rumelhart, D.** and **J. McClelland,** (1986), Parallel Distributed Processing, MIT Press, Cambridge.

**Scott, J.,** (1981), The Probability of Bankruptcy: A Comparison of Empirical Predictions and Theoretical Models, *Journal of Banking and Finance*, Vol. **5**, No. **2**, pp. 317 - 344.

**Surkan, A.** and **J. Singleton,** (1990), Neural Networks for Bond Rating Improved by Multiple Hidden Layers, *Proceedings of IEEE International Conference on Neural Networks*, IEEE Press, Alamitos, CA, Vol. **2**, pp. 157-162.

**Tam, K. Y.** and **M.Y. Kiang,** (1992), Managerial Applications of Neural Networks: The Case of Bank Failure Predictions, *Management Science*, Vol. **38**, pp. 926-947.

**Tam, K. Y.,** (1991), Neural Network Models and The Prediction of Bank Bankruptcy, Omega, *The International Journal of Management Science*, Vol. **19**, pp. 429-445.

**Trippi, R.,** and **E. Turban,** (eds)., (1992), *Neural Network Applications in Finance and Investing*, Probes Publishing, Chicago.

**Tsukuda, J.** and **S. Baba,** (1994), Predicting Japanese Corporate Bankruptcy In Term of Financial Data Using Neural Network, *Selected papers from the 16th Annual Conference on Computers and Industrial Engineering*, Elsevier Science Ltd., Vol. **27**, Nos **1-4**, pp. 445-448.

**Wasserman, P. D.,** (1989), Neural Computing: Theory and Practice, Van Nostrand Reinhold, New York.

**White, H.,** (1988), Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns, *Proceedings of IEEE International Conference on Neural Networks*, IEEE Press, San Diego, CA, pp. 451-459.

**Yoon, Y.** and **G. Swales,** (1990), Predicting Stock Price Performance*, Proceedings of the 24th Hawaii International Conference on System Sciences*, IEEE Press, Alamitos, CA, Vol. **4**, pp. 156-162.

# استخدام نظام الشبكات العصبونية
# للتنبؤ بأداء البنوك الإسلامية

**محمود حمدان العصيمي**

*أستاذ مساعد*

*قسم الاقتصاد - كلية الاقتصاد والإدارة*

*جامعة الملك عبدالعزيز- جدة - المملكة العربية السعودية*

**المستخلص:** يهدف هذا البحث إلى استخدام الشبكات العصبونية للتنبؤ بأداء البنوك الإسلامية. لتحقيق هذا الهدف تم الحصول على عينة مكونة من ستة و عشرون بنكاً إسلامياً خلال الأعوام من ١٩٩١م إلى ١٩٩٣م .استخدمت هذه البيانات للحصول على سبع نسب مالية مرتبطة بمستوى الأداء . في البداية تم تصميم شبكة كونن العصبونية المعروفة بقدرتها العالية على التمييز بين مختلف الفئات بدون معرفة مسبقة من قبل الشبكة. استخدمت هذه الشبكة لتقسيم البنوك الإسلامية إلي مجموعتين ، مجموعة البنوك عالية الأداء و مجموعة البنوك منخفضة الأداء ، وذلك باستخدام النسب المالية لسنة الأداء (١٩٩٣م). وقد صنفت هذه الشبكة اثني عشر بنكاً عالي الأداء و أربع عشر بنكاً منخفض الأداء . وباستخدام تقسيم هذه الشبكة تم تدريب شبكة باك بروبقيشن Backpropagation العصبونية وذلك باستخدام بيانات السنة السابقة لسنة الأداء. و عند عرض البيانات على هذه الشبكة المدربة أظهرت قدرة عالية على التنبؤ. فقد استطاعت هذه الشبكة على تصنيف جميع البنوك تصنيفاً صحيحاً عندما عرضت عليها بيانات سنة الأداء والسنة السابقة لها. ولاختبار قدرة هذه الشبكة المدربة على التنبؤ بأداء البنوك فقد عرضنا على الشبكة بيانات لم ترها من قبل وهي بيانات سنة ١٩٩١م فصنفت ثلاثة وعشرين بنكاً تصنيفاً صحيحاً أي بنسبة ٨٩٪ . هذه النسبة المرتفعة تعني إمكانية استخدام الشبكة كنظام إنذار مبكر لأداء البنوك الإسلامية عندما تعرض عليها بيانات جديدة .