Financial Prediction using Inductive Models

Turki Y. Abdalla ***Abdulkareem. Y. Abdalla*** ***Lamya A. Dexon****

***College of engineering , University of Basrah**

****College of science, University of Basrah**

Abstract

Financial prediction is an example of a prediction problem which is challenging due to small sample sizes, high noise, non-stationary, and non-linearity. Neural networks have been frequently used in financial prediction because of their ability to deal with uncertain, fuzzy, or insufficient data. Despite that, neural networks(NN) have limitations; they still require a significant amount of a priori information about the model structure. Group Method of Data Handling (GMDH) is an inductive approach which attempts to overcome the subjectiveness of neural networks based on the principle of self-organization. We have developed an algorithm inspired from the evolutionary manner of conventional GMDH to generate an inductive model based on using multilayer perceptron that can avoid some of GMDH problems like the exhaustive computations on candidate Adalines and the increasing number of Adalines in the following layers.

1. Inductive and deductive modeling

The capability of induction is fundamental for human thinking. It is the next human ability that can be utilized in soft-computing, besides that of learning and generalization. The induction means gathering small pieces of information, combining it, using already collected information in the higher abstraction level to get complex overview of the studied object or process. Inductive modeling methods utilize the process of induction to construct models of studied systems [1].

The inductive methods have a similar concept to that of evolution introduced by Holland where a number of solutions are created and an external criterion plays the role of finding the fittest [2].

Inductive methods are objective, they can produce a self organization models On the other hand, Deductive methods are subjective since they are totally based on detailed instructions from the person performing the modeling as well as his/her idea about the object behavior [3].

2. Neural Networks in Financial Prediction

Many attempts have been made to predict the behavior of bonds, currencies, stocks, stock markets, or other economic markets . These attempts were encouraged by various evidences that economic markets do not behave randomly, but rather perform in a chaotic manner [4-8].

Neural networks (NNs) have become very important method for finance and investing applications because of their ability to deal with uncertain, fuzzy, or insufficient data which fluctuate rapidly in very short periods of time [9-11].

NNs differ from conventional techniques in that the analyst is not required to specify the nature of the relationships involved; the analyst simply identifies the inputs and the outputs. In addition, the NNs' main strength lies in its ability to vary in complexity, from a simple parametric model to a highly flexible, nonparametric model [11,12].

Despite the small number of assumptions in NNs in comparison to statistical methods, they still require a significant amount of a priori information about the model's structure. Experts should decide on the quality and quantity of input arguments, the number of hidden layers and neurons as well as the form of their activation function [11].

3. GMDH Neural Networks

GMDH neural network is an inductive approach which attempts to overcome the subjectiveness of neural networks based on the principle of self-organization. An inductive approach is similar to neural networks but is unbounded in nature, where the independent variables of the system are shifted in a random way and activated so that the best match to the dependent variables is ultimately selected [12].

An idea behind GMDH algorithms which developed by Ivakhnenko (1966) is based on an evolution principle introduced by Holland which implies the *generation* and *selection* of the *candidate neurons* and an external criterion plays the role of finding the fittest [13].

The GMDH network is a kind of multilayered networks composed by a suitable combination of an Adalines which is illustrated in Figure 1.

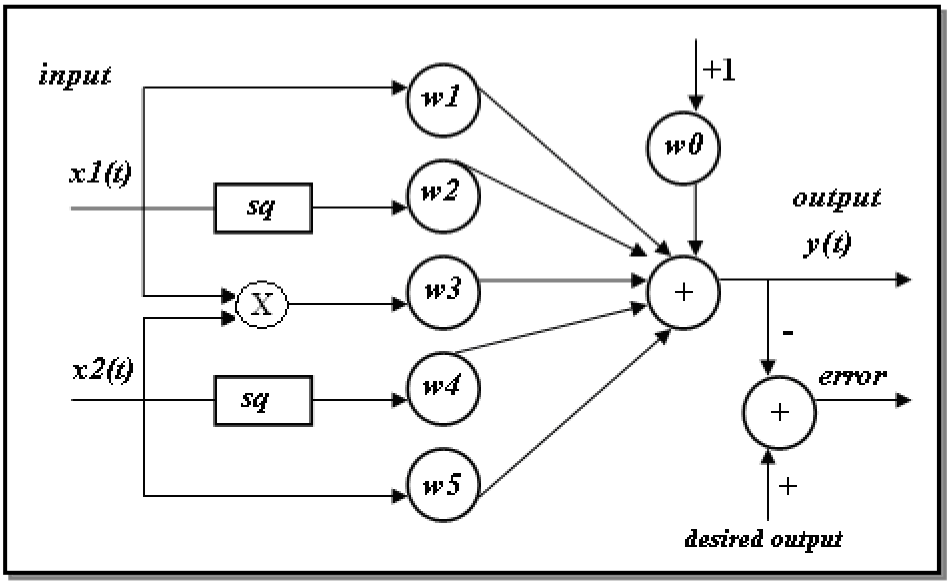

Figure 1: structure of Adaline with a bi-quadratic polynomial function

The output variable y(t) is expressed as follows:

$y(t) = w_0 + w_1 x_1(t) + w_2 x_1^2(t) + w_3 x_1(t) x_2(t) + w_4 x_2^2(t) + w_5 x_2(t)$ ..(1)

Where w_i are weights, and are determined by the least squares method and x_1, x_2 are the inputs to the unit.

Let $W = [w_0\ w_1\ w_2\ w_3\ w_4\ w_5]^T$ and $X = [1\ x_1\ x_1^2\ x_1x_2\ x_2^2\ x_2]^T$

The Widrow – Hoff delta rule adopted for training W is as follows [9, 10]:

$W_{k+1} = W_k + \beta\ (X_k / |X_k|^2)(y_d^k – W_k^T X_k)$ ..(2)

Where y_d^k is the desired output at time k, β is the learning rate with the range (0,1) and $|X_k|$ is the square of length of input vector. The application of equation (2) causes W to be modified so the difference between the desired and actual outputs will reduce. The network structure is determined by the following steps [13-16]:

Step l: separate the experimental data set into a training data set and a selection data set;

Step 2: create a layer of N Adalines based on the number of inputs. Every pair of inputs produces an Adaline. A combination C_2^i Adalines are created if there are i input variables;

Steps 3: use the training data to train all Adalines in the created layer and estimate the weighing coefficients for each Adaline with the least squares method;

Step 4: input the selection data set to the network and obtain the error of all trained Adalines in the layer. Assign a threshold; keep the Adalines whose errors are below the threshold and use them to make the next layer;

Step 5: If the smallest error of the current layer is larger than that of the previous layer or the current layer has only one Adaline, stop the training and trim the network.

Step 6: test the performance of the trained network with evaluation data set. The evaluation data could be a combination of the training and selection data sets or a completely new data set.

The schematic diagram of a GMDH network with selection factor F=0.5 is shown in Figure 2.

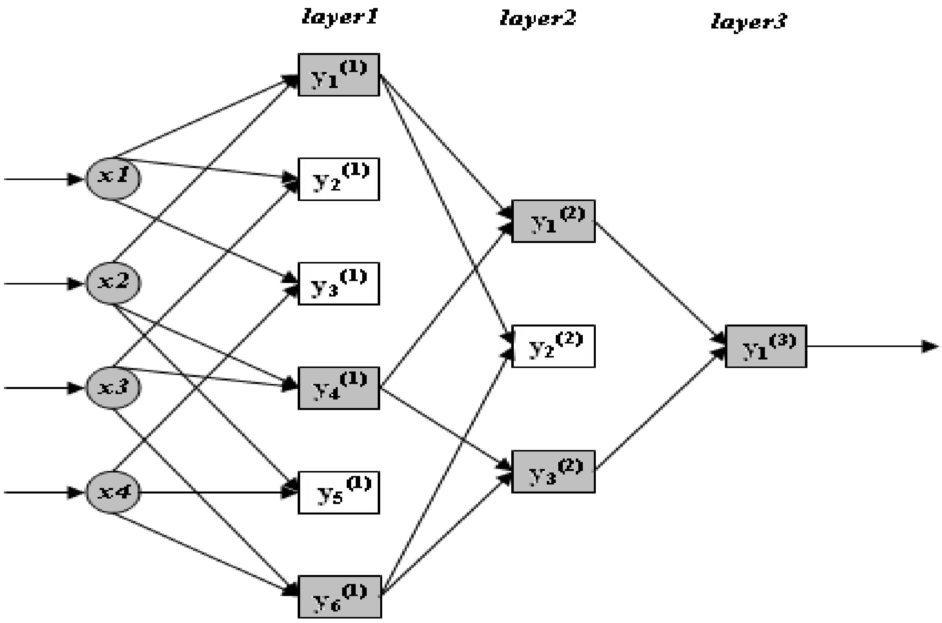

Figure 2 : The structure of neural network grown by GMDH algorithm

The Adalines that were selected after trimming process at the layers are depicted in Figure 2 as the gray boxes. A resulting network is the three-layer network consisting of six Adalines and four input nodes.

4. The Proposed Inductive Model

An idea behind our algorithm is to select the neurons one-by-one and then add them to a binary network according to an external criterion . Only the neuron with the best selection error is added to an inductive network where the independent variables of the system are shifted in a random way and activated so that the best match to the dependent variable is ultimately selected. Following such a procedure there is a gradual increase of complexity and the optimum model is found which has a binary tree structure that is shown in Figure 3.

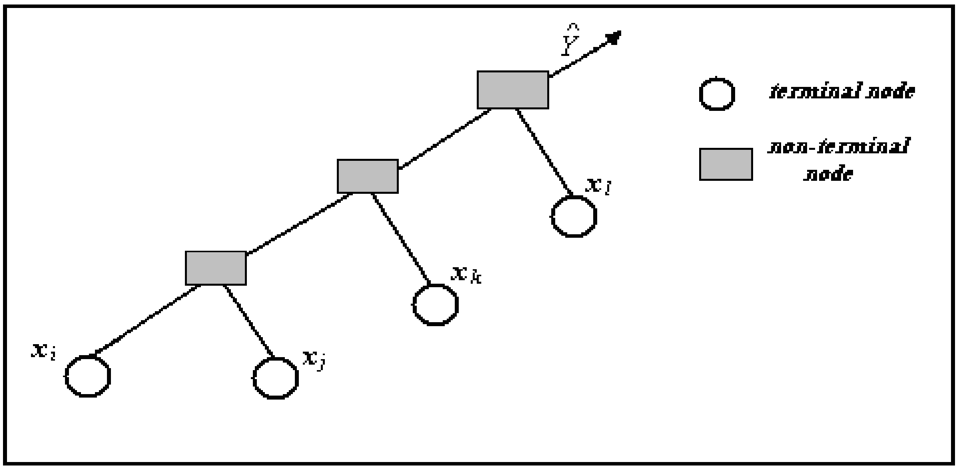

Figure 3: A binary tree with four terminals

In Figure 3, the binary tree takes four terminals or inputs $\{x_i, x_j, x_k, x_l\}$ to produce a predicted $\hat{Y}$, Where, i, j, k, l are in an arbitrary order. Each non-terminal node is performed a transfer function to estimate the output which is re-entered as an input in the following non-terminal node. The network is growing in an evolutionary principle; each time one input is added to the network until some a specific condition is verified.

Collected data are distributed between training and selection sets. Training set is used during training stage to estimate weight coefficients while the selection set is used to measure the generalization ability of the trained network to unseen data.

Deductive Neural Networks have the design problem. Also, these neural networks cannot provide an automated selection of essential input variables and all these omissions solved by trial and error experiments. Our idea is to exploit the good training ability of these deductive neural networks in building an inductive model (shown in Figure 3) where each non terminal node is a Multi-Layer Perceptron (MLP). The algorithm can be summarized as follows:

Step1: ***initialization:*** a population of **N** individuals (chromosomes) is randomly created. Each chromosome represents a proposed binary network. The chromosome has the form and the features as explained in Figure 4.

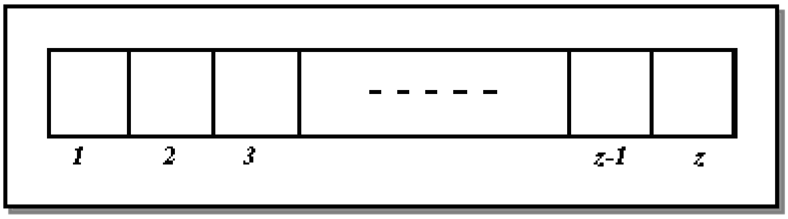

Figure 4: The form of the networks chromosome

Where:

The integer i=1,2 ….,z : means the ***i*** th input variable . The i^{th} input is not active if the corresponding gene contains zero, otherwise it is active.

Step2: **N** networks are constructed corresponding to **N** chromosomes then trained using training data set and a suitable training algorithm.

Step3: ***evaluation:*** a selection stage is performed by testing the generalization ability of all trained network to unseen data (selection data).

Step4: ***Reproduction:*** Genetic Algorithm operators of Rank selection, Uniform crossover, One-point mutation, evaluation and Replacement is repeatedly executed until a stopping condition is satisfied.

Figure 5 displays the network structure of this proposed inductive model. Each building unit in the proposed model is MLP trained using Back- Propagation learning algorithm; it has two inputs; hidden layer with one or more hidden neuron and one output neuron represents the predicted output.

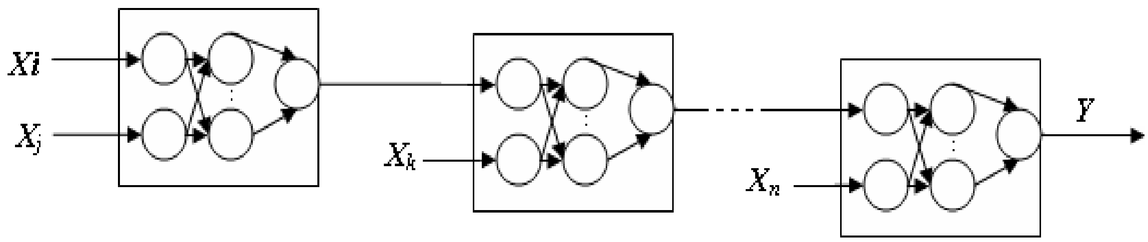

Figure 5: MLP inductive model network

5. Application of the proposed models for stock market Prediction

The value of a stock market price is established by analyzing the fundamental information associated with the market and affected it. The fundamental information consists of several input parameters which are used as input/output for our proposed prediction models. The input variables are:

1. Listed companies: number of listed companies in the market within a time period.

2. Exchange rates: the average of exchange rates between the local currency and US dollar within a period.

3. Market capitalization: Sum of the capitalizations of all listed companies in the market within a time period.

4. Value traded: the overall value of stocks that traded in the market within a time period.

5. Shares traded: sum of the shares (stocks) that traded in the market within a time period.

6. Days traded: number of the traded days in the stock market within a time period.

7. Turnover (TOR): is the result of division the value traded on the market capitalization in a time period multiplied by 100.

TOR = (value traded / capitalization) * 100

8. Capitalization average: is the result of the division of the Gross Domestic Product Average (GDPA) of the country to its market capitalization.

Capitalization average = GDPA / capitalization

The output variable is the price of the market's stock and it is computed as:

Stock price = value traded / shares traded.

A Saudi stock market is taken as a case study. The application data consist of the monthly averaged for the input parameters described above. There are 141 trading months within the period from January 1996 to September 2007 taken from the website www.amf.org [17].

The input and the output data sets to the networks are preconditioned by normalizing the data. The purpose of normalizing the data is to modify the variable levels to a reasonable value. Normalizing data entails manipulating it to be in the range between 0 and 1. The networks are trained utilizing the normalized data sets. In our work data are first preprocessed by log function then normalization is done by dividing each data item on the greatest input value.

A Genetic Algorithm population of 7 binary individuals (proposed networks) is randomly initialized. The algorithm that is described in earlier is executed with Genetic parameters of 0.9 for crossover and 0.1 for mutation. The network that is shown in Figure 6 is the final result of the algorithm execution.

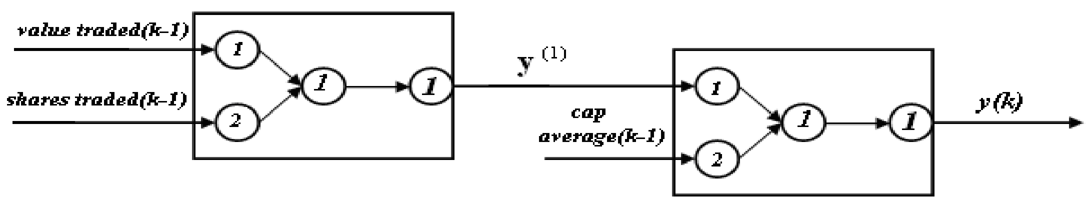

Figure 6: Trained MLP based inductive network

Trained MLP inductive model consists of two MLP networks; each one of them consists of 2 input units, 1 hidden and 1 output neurons. The first and the second MLP networks are trained by 8000 and 6000 training cycles respectively. The adopted learning rate was 0.45 and the momentum was 0.5. The prediction ability of the trained

MLP inductive model for stock market application was tested by applying it with the overall data and the test results for the prediction are shown in Figure 7.

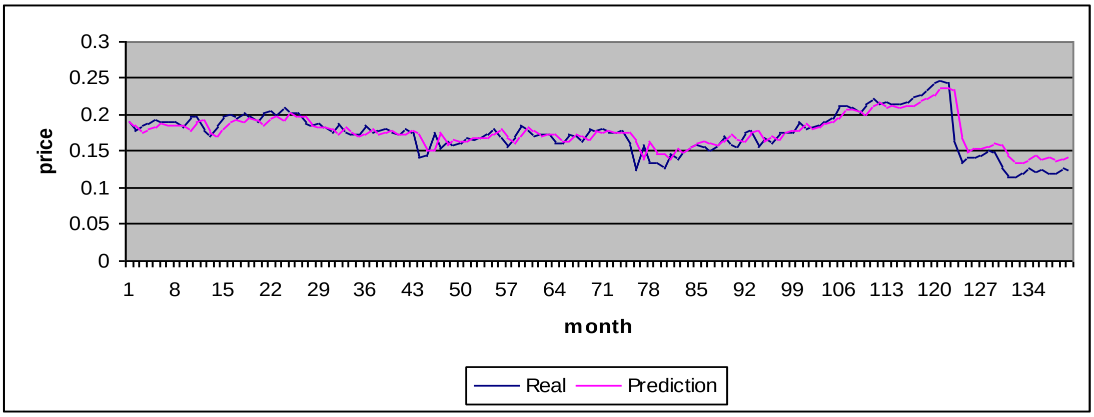

Figure 7: MLP based inductive model prediction results

6. Conclusion

MLP inductive model gives good prediction results For comparison, the SE errors at the evaluation stage are computed for the proposed model and some other types of NNs as shown in Table 1.

$$\mathbf{SE} = \sum_k (y_t(k) - y(k))^2$$

Where: $y_t(k)$ is the desired output and y (k) is the network output at time k

Table 1: SE of the proposed model for stock market prediction application

No	proposed method	SE
1	Bilinear GMDH NN	0.211835
2	Biquadratic GMDH NN	0.069382
3	Modified Biquadratic GMDH NN	0.174637
4	MLP network	0.112403
5	MLP inductive model	0.02435

References:

[1] P. Kordik, M. Snorek, M. Genyk-Berezovskyj, " Hybrid Inductive Models: Deterministic Crowding Employed", 2004.

[2] H-S Park, B-J Park, H-K Kim, and S-K Oh, " Self-Organizing Polynomial Neural Networks Based on Genetically Optimized Multi-Layer Perceptron Architecture", International Journal of Control, Automation and Systems, vol. 2, no. 4, pp. 423-434, 2004.

[3] L. Anastasakis, and N. Mort, "The Development of Self-Organization Techniques in Modelling: A Review of the Group Method of Data Handling (GMDH)", 2001.

[4] K. Chakraborty, M. Kishan, K. M. Chlukuri, and R. Sanjay, "Forecasting the Behavior of Multivariate Time Series Using Neural Networks", Neural Networks, 5,96 1-970, 1992.

[5] R. Tsay, "Analysis of Financial Time Series", Wiley Inter science, 2005

[6] M. E. Malliaris, "Modeling the Behavior of the S&P500 Index: A Neural Network Approach", Proc. 10th IEEE Conference on AI for Applications, 86-90, 1994.

[7] J. Yang, and N. Gradjoevic , "Non-linear, non-parametric, non-fundamental exchange rate forecasting". Journal of Forecasting 25(4), 2006.

[8] AK Dhamija, and VK Bhalla, "Financial Time Series Forecasting: Comparison of Neural Networks and ARCH Models" ,International Research Journal of Finance and Economics , Issue 49, (2010).

[9] E. Schoeneburg, "Stock Price Prediction Using Neural Networks: A Project Report", Neurocomputing, vol. 2, 1990, pp. 17-27.

[10] S. Simon, and A. Raoot, "Accuracy driven artificial neural networks in stock market prediction", International Journal on Soft Computing (IJSC) Vol.3, No.2, May 2012.

[11] N. W. Tan, "A Hybrid Financial Trading System Incorporating Chaos Theory, Statistical and Artificial Intelligence/Soft Computing Methods", Queensland Finance Conference 1999.

[12] G. R. Weckman, and R. Agarwala, "Identifying Relative Contribution of Selected Technical Indicators in Stock Market Prediction", 2003.

[13] A. G. Ivakhnenko, and J-A. Muller, "Recent Developments of Self-Organising Modelling in Prediction and Analysis of Stock Market", 2000.

[14] M. Caudill, and C. Butler, "Understanding Neural Networks: Computer Explorations", 2000.

[15] V. Schetinin, J. Schult, and A. Brazhnikov, "Neural Network Techniques for visual mining clinical electroencephalograms", 2005.

[16] A. Sakaguchi, and T. Yamamoto, "A Study on System Identification Using GA and GMDM Network", IEEE 2003.

[17] [86] Arab Monetary Foundation, www.amf.org.