Expert Systems with Applications 37 (2010) 6302–6309

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Forecasting stock exchange movements using neural networks: Empirical evidence from Kuwait

Mohamed M. Mostafa

New York Institute of Technology, Global Program in Bahrain, Manama, Bahrain

ARTICLE INFO

Keywords:
KSE
Neural networks
Forecasting

ABSTRACT

Financial time series are very complex and dynamic as they are characterized by extreme volatility. The major aim of this research is to forecast the Kuwait stock exchange (KSE) closing price movements using data for the period 2001–2003. Two neural network architectures: multi-layer perceptron (MLP) neural networks and generalized regression neural networks are used to predict the KSE closing price movements. The results of this study show that neuro-computational models are useful tools in forecasting stock exchange movements in emerging markets. These results also indicate that the quasi-Newton training algorithm produces less forecasting errors compared to other training algorithms. Due to their robustness and flexibility of modeling algorithms, neuro-computational models are expected to outperform traditional statistical techniques such as regression and ARIMA in forecasting stock exchanges' price movements.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction and related work

The increasing globalization of the financial markets has heightened the interest in emerging markets. In the Middle East there are 11 formal stock markets monitored by Standard and Poors Emerging Markets Database (Smith, 2007). Stock markets are usually classified as either developed or emerging. Three of the Middle East stock markets are categorized as developed: Kuwait, United Arab Emirates and Qatar. Middle East stock markets are, however, relatively small by world standards as the 11 formal stock markets in the region account for around 0.9% of world stock market capitalization (Smith, 2007).

Kuwait stock exchange (KSE) was officially established in 1984 after the crash of Almanakh (over-the-counter market). In the post liberation period (1992) KSE witnessed a lot of reforms among which were the government privatization program to sell its holdings of shares in local shareholding companies, the launching of mutual funds, and the emergence of institutional investors as the dominant players in the market (Al-Loughani & Chappell, 2000).

Throughout its history, KSE has been characterized by an irregularity in trades and price formation process. However, the Kuwaiti parliament has recently introduced some measures that permit foreign investors to trade in KSE (Al-Loughani & Chappell, 2000). This study aims to forecast the KSE movements using neural network (NN) models.

NN models have been successfully used in prediction or forecasting studies across many disciplines. One of the first successful applications of MLP is reported by Lapedes and Farber (1988). Using two deterministic chaotic time series generated by the logistic map and the Glass–Mackey equation, they designed an MLP that can accurately mimic and predict such dynamic non-linear systems. Another major application of MLP is in electric load consumption (e.g. Darbellay & Slama, 2000; McMenamin & Monforte, 1998). Many other problems have been solved by MLP. A short list includes air pollution forecasting (e.g. Videnova, Nedialkova, Dimitrova, & Popova, 2006), maritime traffic forecasting (Mostafa, 2004), airline passenger traffic forecasting (Nam & Yi, 1997), railway traffic forecasting (Zhuo, Li-Min, Yong, & Yan-hui, 2007), commodity prices (Kohzadi, Boyd, Kemlanshahi, & Kaastra, 1996), ozone level (Ruiz-Suarez, Mayora-Ibarra, Torres -Jimenez, & Ruiz-Suarez, 1995), student grade point averages (Gorr, Nagin, & Szczypula, 1994), forecasting macroeconomic data (Aminian, Suarez, Aminian, & Walz, 2006), financial time series forecasting (Yu & Lai Wang, 2009), advertising (Poh, Yao, & Jasic, 1998), and market trends (Aiken & Bsat, 1999).

Due to its good performance in noisy environments, the generalized regression neural network (GRNN) has been extensively used in various prediction and forecasting tasks in the literature. For example, Gaetz, Weinberg, Rzempoluck, and Jantzen (1998) analyzed EEG activity of the brain using a GRNN. Chtioui, Panigrahi, and Francl (1999) used a GRNN for leaf wetness prediction. Ibric, Jovanovi, Djuri, Paroj, and Solomun (2002) used a GRNN in the design of extended-release aspirin tablets. Cigizoglu (2005) employed a GRNN to forecast monthly water flow in Turkey. In this study, GRNN forecasting performance was found to be superior to the MLP and other statistical and stochastic methods. Kim and Lee

E-mail address: mmosta02@nyit.edu

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.02.091

(2005) used a GRNN-based genetic algorithm to predict silicon oxynitride etching. Hanna, Ural, and Saygili (2007) developed a GRNN model to predict seismic condition in sites susceptible to liquefaction. Shie (2008) used a hybrid method integrating a GRNN and a sequential quadratic programming method to determine an optimal parameter setting of an injection-molding process.

There is an extensive literature in financial applications of NNs (e.g. Harvey, Travers, & Costa 2000; Kumar & Bhattacharya, 2006). For example, Cao, Leggio, and Schniederjans (2005) used NNs to predict stock price movements for firms traded on the Shanghai stock exchange. The authors compared the predictive power using linear models to the predictive power of the univariate and multivariate NN models. Results showed that NN models outperform the linear models. These results were statistically significant across the sample firms and indicated that NN models are useful for stock price prediction. Kryzanowski, Galler, and Wright (1993) used NN models with historic accounting and macroeconomic data to identify stocks that will outperform the market. McGrath (2002) used market to book and price earnings ratios in a NN model to rank stocks based on the likelihood estimates. Ferson and Harvey (1993) and Kimoto, Asakawa, Yoda, and Takeoka (1990) used a series of macroeconomic variables to capture predictable variation in stock price returns. McNelis (1996) used the Chilean stock market to predict returns in the Brazilian markets. Yumlu, Gurgen, and Okay (2005) used various NN architectures to model the performance of Istanbul stock exchange over the period 1990–2002. Leigh, Hightower, and Modani (2005) used NN models and linear regression models to model the New York Stock Exchange Composite Index data for the period from 1981 to 1999. Results were robust and informative as to the role of trading volume in the stock market. Chen, Leung, and Daouk (2003) predicted the direction of return on market index of the Taiwan stock exchange using a probabilistic NN model. The results were then compared to the generalized methods of moments (GMM) with Kalman filter. From this literature survey we find that no previous studies have attempted to predict the movements of KSE. In this study we aim to fill this research gap through the application of MLP and GRNN models to forecast the movements of the KSE.

2. Methodology

2.1. Data

This study covers the time period of June 17, 2001 through November 30, 2003. Thus, the data set contained 612 data points in time series. This data set is quite similar in length to data sets in previous studies of similar nature. We used data for all listed companies traded on the KSE. The data consists of daily closing prices. The source of the closing price data is the KSE (www.Kuwaitse.com). Table 1 shows descriptive statistics of the open and closing prices of the data used in this study.

2.2. Multi-layer perceptron

MLP consists of sensory units that make up the input layer, one or more hidden layers of processing units (perceptrons), and one output layer of processing units (perceptrons). The MLP performs a functional mapping from the input space to the output space. An MLP with a single hidden layer having H hidden units and a single output, y, implements mappings of the form

$$y = F(W_0 + \sum_{h=1}^{H} W_h Z_h) \tag{1}$$

$$Z_h = F(\beta_{\text{oh}} + \sum_{j=1}^{n} \beta_{\text{jh}} Xj) \tag{2}$$

where Z_h is the output of the hth hidden unit, W_h is the weight between the hth hidden and the output unit, and W_0 is the output bias. There are N sensory inputs, Xj. The jth input is weighted by an amount β_j in the hth hidden unit. The output of an MLP is compared to a target output and an error is calculated. This error is back-propagated to the neural network and used to adjust the weights. This process aims at minimizing the mean square error between the network's prediction output and the target output.

MLP was first developed to mimic the functioning of the brain. It consists of interconnected nodes referred to as processing elements that receive, process, and transmit information. MLP consists of three types of layers: the first layer is known as the input layer and corresponds to the problem input variables with one node for each input variable. The second layer is known as the hidden layer and is useful in capturing non-linear relationships among variables. The final layer is known as the output layer and corresponds to the classification being predicted (Baranoff, Sager, & Shively, 2000). Fig. 1 represents the typical structure of MLP.

First of all the network has to be *trained* to produce the correct output with minimum error. To achieve the minimum error the network first has to be trained until it produces a *tolerable* error. This is how the training is done. Input is fed to the input nodes, from here the middle layer nodes take the input value and start to process it. These values are processed based on the randomly allocated initial weight of the links. The input travels from one layer to another and every layer process the value based on the weights of its links. When the value finally reaches the output node, the actual output is compared with the expected output. The difference is calculated and it is propagated backwards, this is when the links adjust their weights. After the error has propagated all the way back to first layer of middle level nodes, the input

Table 1
Descriptive statistics of KSE opening and closing prices.

Open		Close	
Mean	2550.021242	Mean	2553.124346
Standard Error	36.7838047	Standard Error	36.81831376
Median	2196.25	Median	2194.55
Mode	1746.7	Mode	1764.2
Standard Deviation	909.9810725	Standard Deviation	910.8347795
Sample Variance	828065.5523	Sample Variance	829619.9955
Kurtosis	-0.63889504	Kurtosis	-0.64911602
Skewness	0.891027464	Skewness	0.88577702
Range	2945.2	Range	2942.8
Minimum	1578.7	Minimum	1579
Maximum	4523.9	Maximum	4521.8
Count	612	Count	612

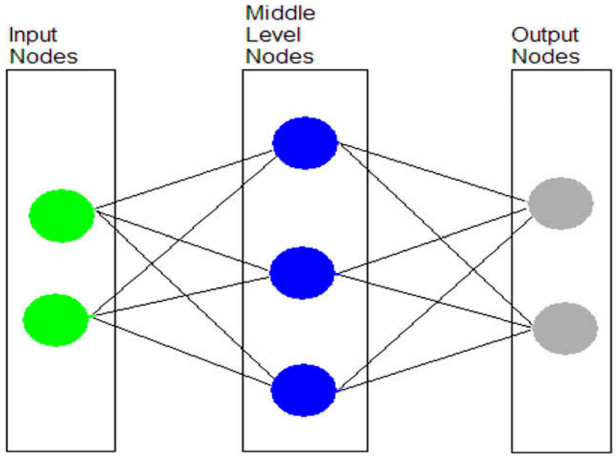

Fig. 1. General structure of MLP.

is again fed to the input nodes. The cycle repeats and the weights are adjusted over and over again until the error is minimized. The key here is the weight of different links. The weights of the links will decide the output value.

The MLP is the most frequently used neural network technique in pattern recognition (Bishop, 1999) and classification problems (Sharda, 1994). However, numerous researchers document the disadvantages of the MLP approach. For example, Calderon and Cheh (2002) argue that the standard MLP network is subject to problems of local minima. Swicegood and Clark (2001) claim that there is no formal method of deriving a MLP network configuration for a given classification task. Thus, there is no direct method of finding the ultimate structure for modeling process. Consequently, the refining process can be lengthy, accomplished by iterative testing of various architectural parameters and keeping only the most successful structures. Wang (1995) argues that standard MLP provides unpredictable solutions in terms of classifying statistical data.

2.3. Generalized regression neural network

GRNN was devised by Specht (1991), casting a statistical method of function approximation into a neural network form. The GRNN, like the MLP, is able to approximate any functional relationship between inputs and outputs (Wasserman, 1993). Structurally, the GRNN resembles the MLP. However, unlike the MLP, the GRNN does not require an estimate of the number of hidden units to be made before training can take place. Furthermore, the GRNN differs from the classical MLP in that every weight is replaced by a distribution of weight which minimizes the chance of ending up in local minima. Therefore, no test and verification sets are required, and in principle all available data can be used for the training of the network (Parojcic, Ibric, Djuric, Jovanovic, & Corrigan, 2005).

The GRNN is a method of estimating the joint probability density function (pdf) of x and y, giving only a training set. The estimated value is the most probable value of y and is defined by

$$E(y|x)y = \hat{y}(x) = \int_{-\infty}^{+\infty} yf(x,y)\mathrm{d}y \Big/ \int_{-\infty}^{+\infty} f(x,y)\mathrm{d}y \tag{3}$$

The density function $f\,(x,y)$ can be estimated from the training set using Parzen's estimator (Parzen, 1962)

$$f(x,y) = 1/(2\pi)^{(p+1)/2}\sigma^{(p+1)}1/n\sum_{i=1}^{n} \times \exp\left[-(x-x^i)^T(x-x^i)/2\sigma^2\right]\exp[-(y-y^i)^2/2\sigma^2] \tag{4}$$

The probability estimate $f\,(x,y)$ assigns a sample probability of width σ for each sample x^i and y^i, and the probability estimate is the sum of these sample probabilities (Specht, 1991). Defining the scalar function D_i^2

$$D_i^2 = (x - x_i)^T(x - x_i) \tag{5}$$

and assessing the indicated integration yields the following:

$$Y(x) = \sum_{i=1}^{n} Y^i \exp(-D_i^2/2\sigma^2) \Big/ \sum_{i=1}^{n} \exp(-D_i^2/2\sigma^2) \tag{6}$$

The resulting regression (6) is directly applicable to problems involving numerical data.

The first hidden layer in the GRNN contains the radial units. A second hidden layer contains units that help to estimate the weighted average. This is a specialized procedure. Each output has a special unit assigned in this layer that forms the weighted sum for the corresponding output. To get the weighted average from the weighted sum, the weighted sum must be divided through by the sum of the weighting factors. A single special unit in the second layer calculates the latter value. The output layer then performs the actual divisions (using special division units). Hence, the second hidden layer always has exactly one more unit than the output layer. In regression problems, typically only a single output is estimated, and so the second hidden layer usually has two units. Fig. 2 shows the general structure of the GRNN. The GRNN can be modified by assigning radial units that represent clusters rather than each individual training case: this reduces the size of the network and increases execution speed. Centers can be assigned using any appropriate algorithm (i.e., sub-sampling, K-means or Kohonen). Fig. 2 shows the General structure of the GRNN.

3. Results

3.1. MLP-based forecasting

There are many software packages available for analyzing MLP models. We chose NeuroIntelligence package (Alyuda Research Company, 2003). This software applies artificial intelligence techniques to automatically find the efficient MLP architecture. Typically, the application of MLP requires a training data set and a testing data set (Lek & Guegan, 1999). The training data set is used to train the MLP and must have enough examples of data to be representative for the overall problem. The testing data set should be independent of the training set and is used to assess the classification/prediction accuracy of the MLP after training. Following Lim and Kirikoshi (2005), an error back-propagation algorithm with

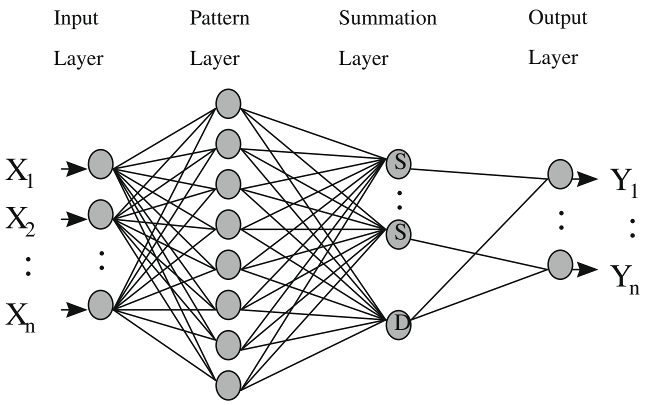

Fig. 2. General structure of the generalized regression neural network (GRNN).

weight updates occurring after each epoch was used for MLP training.

Fig. 3 shows the actual versus the fitted closing price values for the whole series using the quick propagation training algorithm. Fig. 4 shows the negative exponential decay in the error rate. As can be seen from Fig. 4, the best network was obtained after around 500 epochs (trials). This figure is quite similar to typical convergence of errors in MLP models (similar error graphs were obtained using other training algorithms).

Fig. 5 shows the actual versus the fitted closing price values for the whole series using the conjugate gradient descent training algorithm. Fig. 6 shows the actual versus the fitted closing price

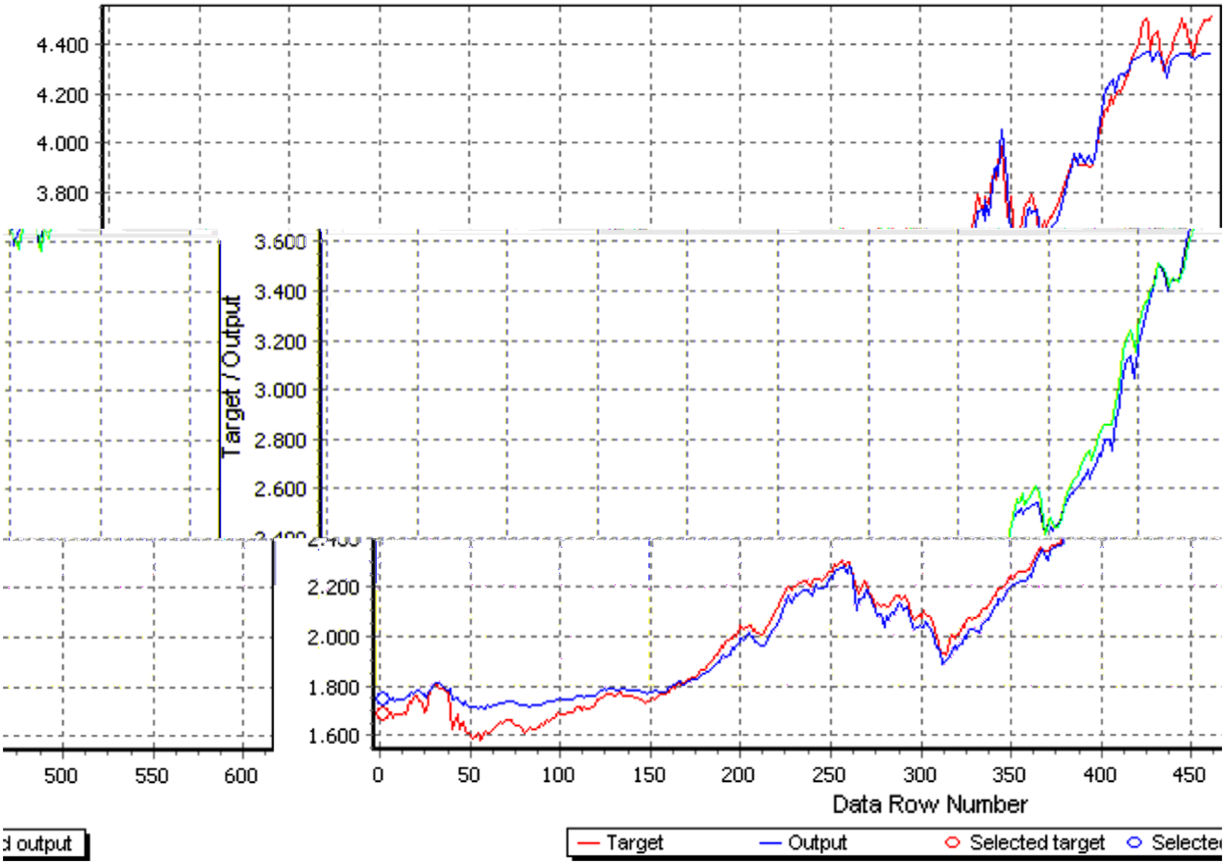

Fig. 3. Actual versus fit using the quick propagation training algorithm.

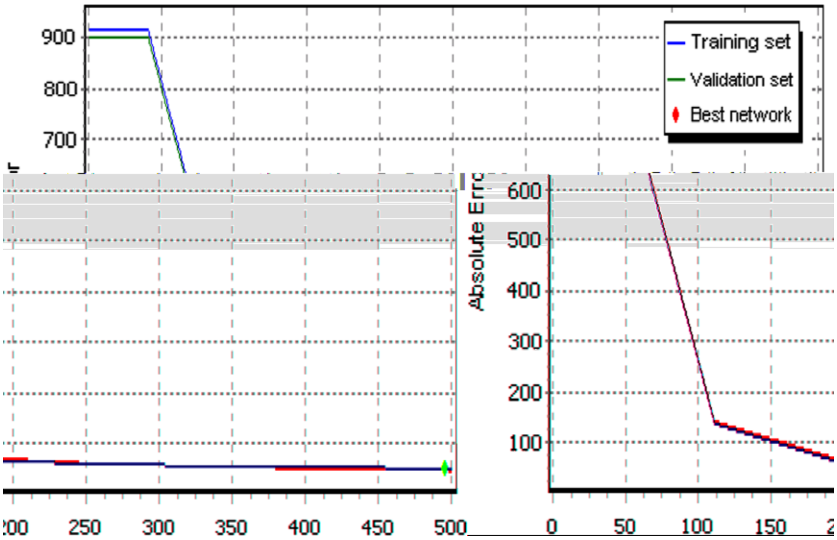

Fig. 4. Error conversion and best network error.

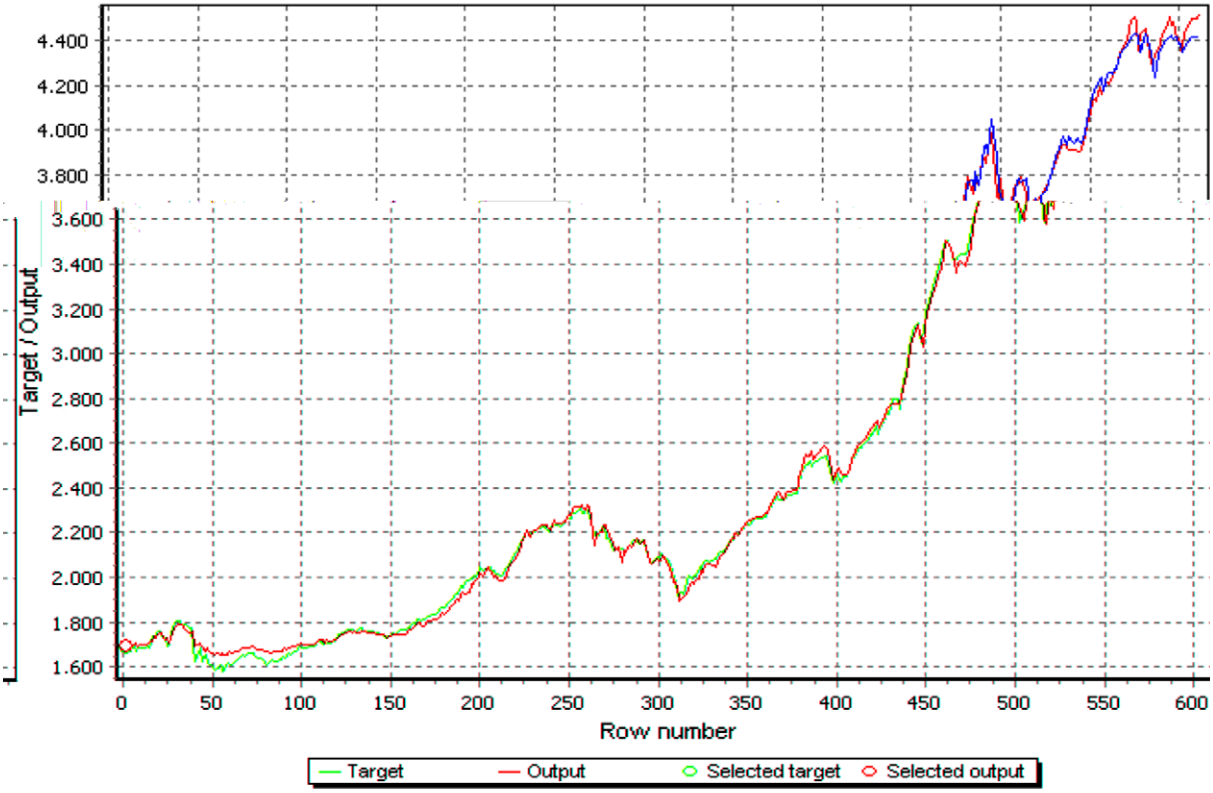

Fig. 5. Actual versus fit using the conjugate gradient descent training algorithm.

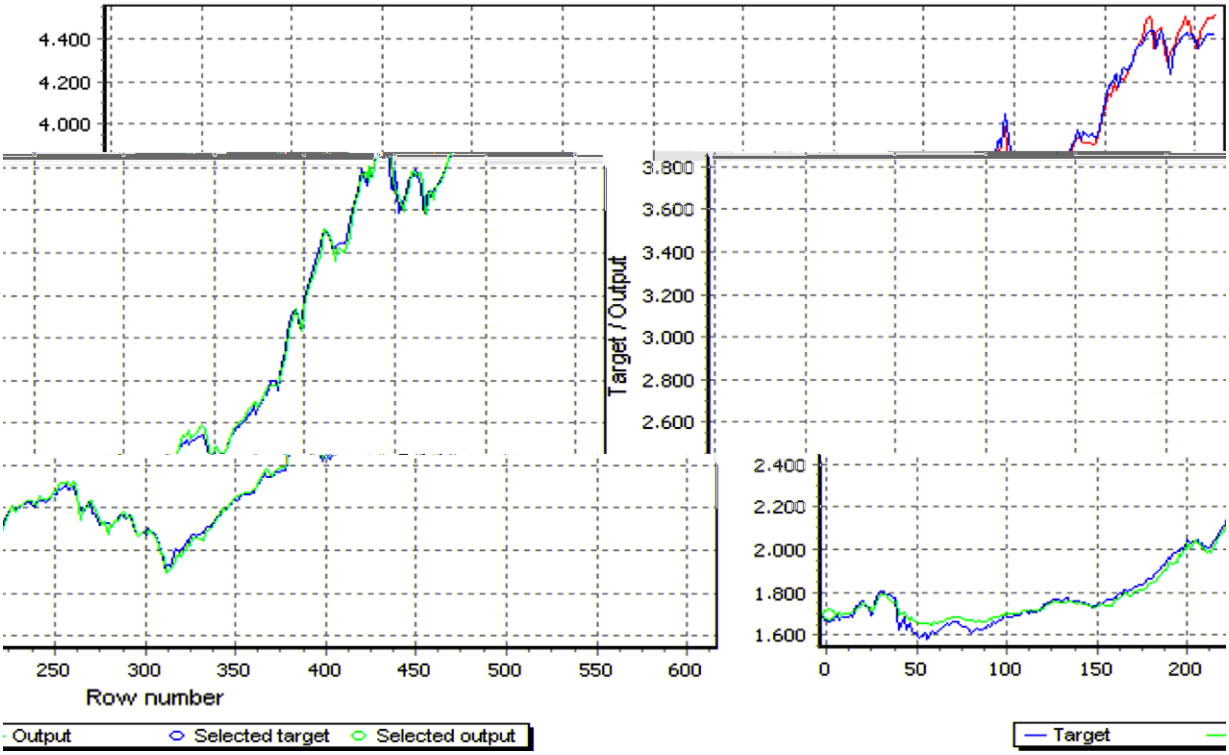

Fig. 6. Actual versus fit using the quasi-Newton training algorithm.

values for the whole series using the quasi-Newton training algorithm.

Based on the descriptive statistics of different training algorithms reported in Table 2 along with the plots of the training

Table 2
MLP training algorithms major statistics.

	Target	Output	AE	ARE
Quick propagation[a]				
Mean	2534.744	2537.564	48.450	0.021
SD	908.299	890.698	31.966	0.015
Min	1579.000	1708.253	0.336	0.000
Max	4521.800	4369.447	155.094	0.084
Conjugate gradient descent[b]				
Mean	2534.744	2535.314	26.598	0.010
SD	908.299	902.767	22.605	0.009
Min	1579.000	1653.851	0.334	0.000
Max	4521.800	4433.294	170.075	0.049
Quasi-Newton algorithm[c]				
Mean	2534.744	2535.313	24.956	0.010
SD	908.299	903.801	21.542	0.008
Min	1579.000	1646.779	0.303	0.000
Max	4521.800	4442.282	166.271	0.046

[a] Correlation = 0998; $R^2 = 0.996$.
[b] Correlation = 0999; $R^2 = 0.999$.
[c] Correlation = 0999; $R^2 = 0.999$.

Table 3
GRNN architecture.

Summary	
Net Information	
Configuration	GRNN Numeric Predictor
Training	
Number of Cases	490
Number of Trials	61
Reason Stopped	Auto-Stopped
% Bad Predictions (30% Tolerance)	0.0000%
Root Mean Square Error	24.51
Mean Absolute Error	17.13
Std. Deviation of Abs. Error	17.53
Testing	
Number of Cases	122
% Bad Predictions (30% Tolerance)	0.0000%
Root Mean Square Error	32.12
Mean Absolute Error	22.30
Std. Deviation of Abs. Error	23.12
Data Set	
Number of Rows	612

algorithms used, it seems that the quasi-Newton training algorithm produces less forecasting errors compared to the other two methods. This is also evident from the target vs. output graph and from the error dependence graph using the quasi-Newton training algorithm (Fig. 7).

3.2. GRNN-based forecasting

There are many computer software packages available for building and analyzing NNs. Because of its extensive capabilities for building networks based on a variety of training and learning methods, NeuralTools Professional package (Palisade Corporation, 2005) was chosen to conduct GRNN analysis in this study. This software automatically scales all input data. Scaling involves mapping each variable to a range with minimum and maximum values of 0 and 1. NeuralTools Professional software uses a non-linear scaling function known as the 'tanh', which scales inputs to a $(-1, 1)$ range. This function tends to squeeze data together at the low and high ends of the original data range. It may thus be helpful in reducing the effects of outliers (Tam, Tong, Lau, & Chan 2005). Table 3 shows the basic properties of the GRNN model used in this study. Figs. 8 and 9 show the error distribution and the predicted versus actual results using the GRNN network. These figures indicate the robustness of the GRNN and its normality of error distributions.

4. Implications, limitations and future research

Our results confirm the theoretical work by Hecht-Nielson (1989) who has shown that NNs can learn input–output relation-

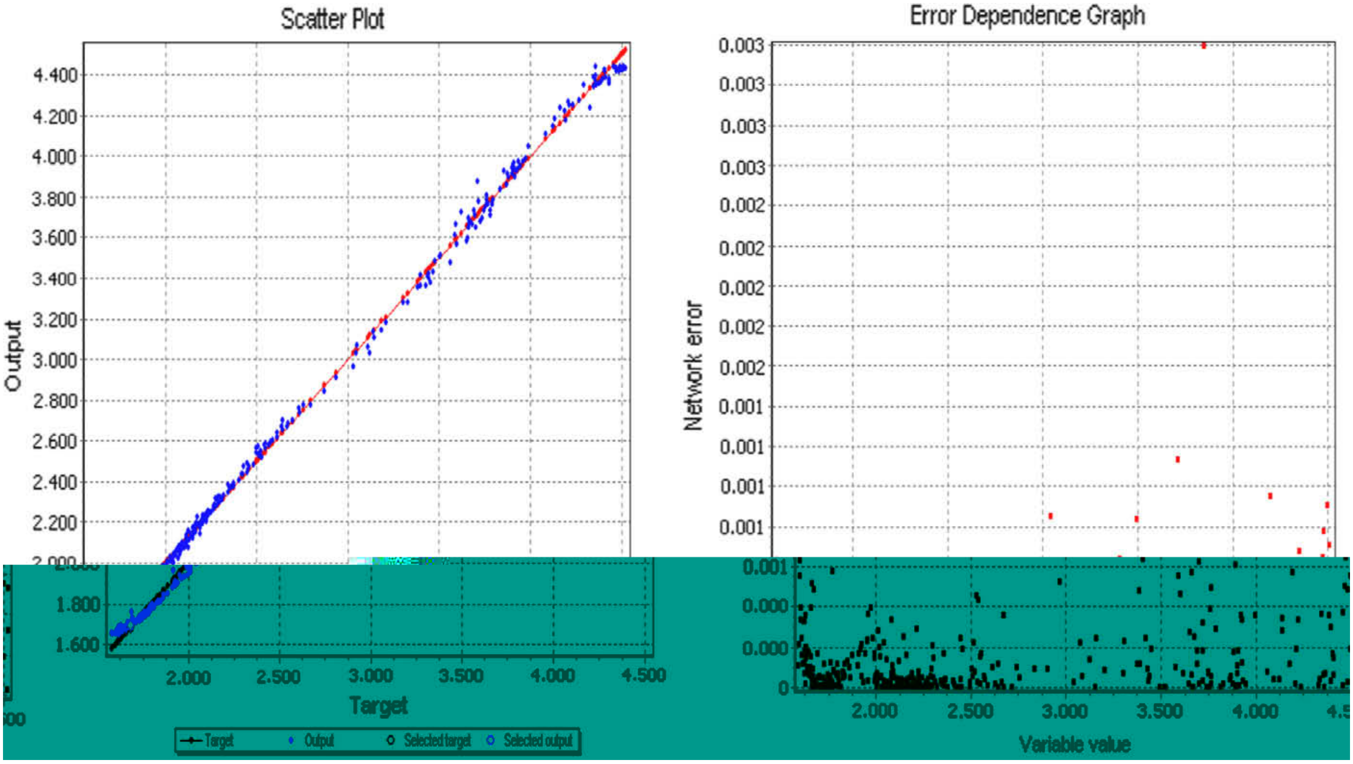

Fig. 7. Predicted Vs. actual and error dependence graph using the quasi-Newton training algorithm.

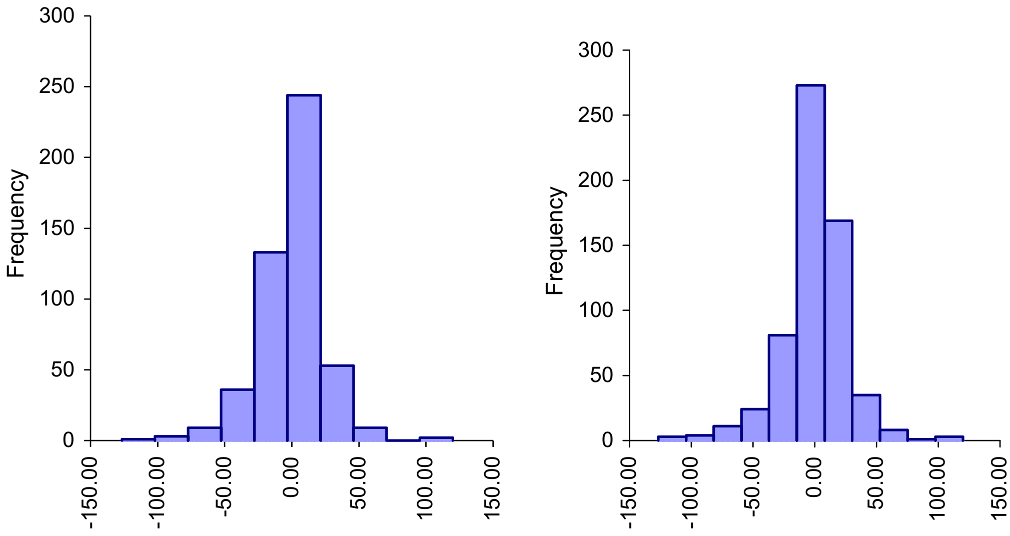

Fig. 8. GRNN histogram of residuals (training/testing).

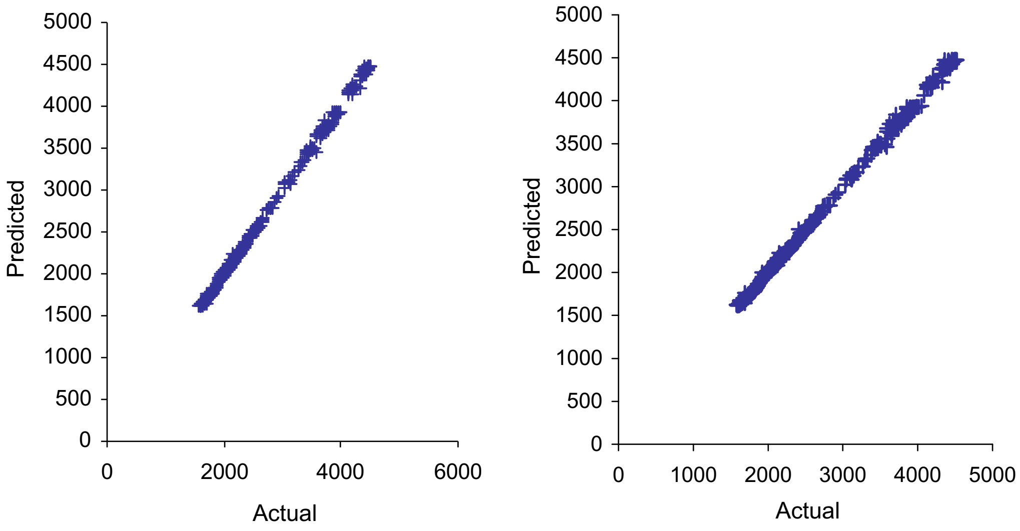

Figure 9. GRNN predicted versus actual values (training/testing).

ships to the point of making perfect forecasts with the data on which the network is trained. The good performance of the NN models in predicting KSE closing price movements can be traced to its inherent non-linearity. This makes an NN ideal for dealing with non-linear relations that may exist in the data. Thus, neurocomputational models are needed to better understand the inner dynamics of stock markets. Our results are also in line with the findings of other researchers who have investigated the performance of NN compared to other traditional statistical techniques, such as regression analysis, discriminant analysis, and logistic regression analysis. For example, in a study of credit-scoring models used in commercial and consumer lending decisions, Bensic, Sarlija, and Zekic-Susac (2005) compared the performance of logistic regression, neural networks and decision trees. The PNN model produced the highest hit rate and the lowest type I error. Similar findings have been reported in a study examining the performance of NN in predicting bankruptcy (Anandarajan, Lee, & Anandarajan 2001) and diagnosis of acute appendicitis (Sakai et al., 2007).

Despite the significant contributions of this study, it suffers from a number of limitations. First, despite the satisfactory performance of the NN models in this study, future research might improve the performance of the NN models used in this study by integrating fuzzy discriminant analysis and genetic algorithms (GA) with NN models. Mirmirani and Li (2004) pointed out that traditional algorithms search for optimal weight vectors for a neural network with a given architecture, while GA can yield an efficient exploration of the search space when the modeler has little *apriori* knowledge of the structure of problem domains. Second, future research might use other NN architectures such as self-organizing maps (SOMs) to classify movements in KSE. Due to the unsupervised character of their learning algorithm and the excellent visualization ability, SOMs have been recently used in myriad classification tasks. Examples include classifying cognitive performance in schizophrenic patients and healthy individuals (Silver & Shmoish, 2008), mutual funds classification (Moreno, Marco, & Olmeda, 2006), crude oil classification (Fonseca, Biscaya, de Sousa,

& Lobo 2006), and classifying magnetic resonance brain images (Chaplot, Patnaik, & Jagannathan 2006).

References

Aiken, M., & Bsat, M. (1999). Forecasting market trends with neural networks. *Information Systems Management, 16*, 42–49.

Al-Loughani, N., & Chappell, D. (2000). Modeling the day-of-the-week effect in the Kuwait stock exchange: A non-linear Garch representation. *Applied Financial Economics, 11*, 353–359.

Alyuda Research Company (2003). NeuroIntelligence User Manual (Version 2.1).

Aminian, F., Suarez, E., Aminian, M., & Walz, D. (2006). Forecasting economic data with neural networks. *Computational Economics, 28*, 71–88.

Anandarajan, M., Lee, P., & Anandarajan, A. (2001). Bankruptcy prediction of financially stressed firms: an examination of the predictive accuracy of artificial neural networks. *International Journal of Intelligent Systems in Accounting, Finance and Management, 10*, 69–81.

Baranoff, E., Sager, T., & Shively, T. (2000). A semi-parametric stochastic spline model as a managerial tool for potential insolvency. *Journal of Risk and Insurance, 67*, 369–396.

Bensic, M., Sarlija, N., & Zekic-Susac, M. (2005). Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. *Intelligent Systems in Accounting, Finance and Management, 13*, 133–150.

Bishop, C. (1999). *Neural networks for pattern recognition*. New York: Oxford University Press.

Calderon, T., & Cheh, J. (2002). A roadmap for future neural networks research in auditing and risk assessment. *International Journal of Accounting Information Systems, 3*, 203–236.

Cao, Q., Leggio, K., & Schniederjans, M. (2005). A comparison between Fama and French's model and artificial networks in predicting the Chinese stock market. *Computers and Operations Research, 32*, 2499–2512.

Chaplot, S., Patnaik, L., & Jagannathan, N. (2006). Classification of magnetic resonance brain images using wavlets as input to support vector machines and neural network. *Biomedical Signal Processing and Control, 1*, 86–92.

Chen, A., Leung, M., & Daouk, H. (2003). Application of neural networks to an emerging financial market: forecasting and trading the Taiwan stock index. *Computers and Operations Research, 30*, 901–923.

Chtioui, Y., Panigrahi, S., & Francl, L. (1999). A generalized regression neural network and its application for leaf wetness prediction to forecast plant disease. *Chemometrics and Intelligent Laboratory Systems, 48*, 47–58.

Cigizoglu, H. (2005). Generalized regression neural network in monthly flow forecasting. *Civil Engineering and Environmental Systems, 22*, 71–84.

Darbellay, G., & Slama, M. (2000). forecasting the short-term demand for electricity: Do neural networks stand a better chance. *International Journal of Forecasting, 16*, 71–83.

Ferson, W., & Harvey, C. (1993). The risk and predictability of international equity returns. *Review of Financial Studies, 6*, 527–566.

Fonseca, A., Biscaya, J., de Sousa, J., & Lobo, A. (2006). Geographical classification of crude oils by Kohonen self-organizing maps. *Analytica Chimica Acta, 556*, 374–382.

Gaetz, M., Weinberg, H., Rzempoluck, E., & Jantzen, K. (1998). Neural network classification and correlation analysis of EEG and MEG activity accompanying spontaneous reversals of the Necker cube. *Cognitive Brain Research, 6*, 335–346.

Gorr, W., Nagin, D., & Szczypula, J. (1994). comparative study of artificial neural network and statistical models for predicting student point averages. *International Journal of Forecasting, 10*, 17–34.

Hanna, A., Ural, D., & Saygili, G. (2007). Evaluation of liquefaction potential of soil deposits using artificial neural networks. *Engineering Computations, 24*, 5–16.

Harvey, C., Travers, K., & Costa, M. (2000). Forecasting emerging market returns using neural networks. *Emerging Markets Quarterly, 4*, 43–55.

Hecht-Nielson, R. (1989). Theory of the back-propagation neural network. In *International joint conference on neural networks* (pp. 593–605). Washington, DC.

Ibric, S., Jovanovi, M., Djuri, C., Paroj, J., & Solomun, L. (2002). The application of generalized regression neural network in the modeling and optimization of aspirin extended release tablets with Eudragit(r) RSPO as matrix substance. *Journal of Controlled Release, 82*, 213–222.

Kim, B., & Lee, B. (2005). Prediction of silicon oxynitride plasma etching using a generalized regression neural network. *Journal of Applied Physics, 98*, 1–6.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. In *Proceedings of the IEEE international conference on neural networks* (pp. 1–16).

Kohzadi, N., Boyd, M., Kemlanshahi, B., & Kaastra, I. (1996). A Comparison of artificial neural network and time series models for forecasting commodity prices. *Neurocomputing, 10*, 169–181.

Kryzanowski, L., Galler, M., & Wright, D. (1993). Using artificial neural networks to pick stocks. *Financial Analysts Journal, 49*, 21–27.

KSE (www.Kuwaitse.com), visited on November 10, 2008.

Kumar, K., & Bhattacharya, S. (2006). Artificial neural network vs. linear discriminant analysis in credit ratings forecast. *Review of Accounting and Finance, 5*, 216–227.

Lapedes, A., & Farber, R. (1988). How neural nets work? In D. Anderson (Ed.), *Neural information processing systems* (pp. 442–456). New York: American Institute of Physics.

Leigh, W., Hightower, R., & Modani, N. (2005). Forecasting the New York stock exchange composite index with past price and interest rate on condition of volume spike. *Expert Systems with Applications, 28*, 1–8.

Lek, S., & Guegan, J. (1999). Artificial neural networks as a tool in ecological modelling: an introduction. *Ecological Modeling, 120*, 65–73.

Lim, C., & Kirikoshi, T. (2005). Predicting the effects of physician-directed promotion on prescription yield and sales uptake using neural networks. *Journal of Targeting, Measurement and Analysis for Marketing, 13*, 158–167.

McGrath, C. (2002). Terminator portfolio. *Kiplinger's Personal Finance, 56*, 56–57.

McMenamin, J., & Monforte, F. (1998). Short term energy forecasting with neural networks. *Energy Journal, 19*, 43–52.

McNelis, P. (1996). A neural network analysis of Brazilian stock prices: Tequila effects vs. pisco sour effects. *Journal of Emerging Markets, 1*, 29–44.

Mirmirani, S., & Li, H. (2004). Gold price, neural networks and genetic algorithm. *Computational Economics, 23*, 193–200.

Moreno, D., Marco, P., & Olmeda, I. (2006). Self-organizing maps could improve the classification of Spanish mutual funds. *European Journal of Operational Research, 147*, 1039–1054.

Mostafa, M. (2004). Forecasting the Suez Canal traffic: A neural network analysis. *Maritime Policy and Management, 31*, 139–156.

Nam, K., & Yi, J. (1997). Predicting airline passenger volume. *Journal of Business Forecasting Methods and Systems, 16*, 14–17.

Palisade Corporation (2005). *NeuralTools professional user guide (version 1.0)*. Ithaca, New York: Palisade Corporation.

Parojcic, J., Ibric, S., Djuric, Z., Jovanovic, M., & Corrigan, O. (2005). An investigation into the usefulness of generalized regression neural network analysis in the development of level A in vitro–in vivo correlation. *European Journal of Pharmaceutical Sciences, 30*, 264–272.

Parzen, E. (1962). On the estimation of a probability density function and mode. *Annals of Mathematical Statistics, 33*, 1065–1076.

Poh, H., Yao, J., & Jasic, T. (1998). Neural networks for the analysis and forecasting of advertising impact. *International Journal of Intelligent Systems in Accounting, Finance and management, 7*, 253–268.

Ruiz-Suarez, J., Mayora-Ibarra, O., Torres -Jimenez, J., & Ruiz-Suarez, L. (1995). Short-term ozone forecasting by artificial neural network. *Advances in Engineering Software, 23*, 143–149.

Sakai, S., Kobayashi, K., Toyabe, S., Mandai, N., Kanda, T., & Akazawa, K. (2007). Comparison of the levels of accuracy of an artificial neural network model and a logistic regression model for the diagnosis of acute appendicitis. *Journal of Medical Systems, 31*, 357–364.

Sharda, R. (1994). Neural networks for the MS/OR analyst: An application bibliography. *Interfaces, 24*, 116–130.

Shie, J. (2008). Optimization of injection-molding process for mechanical properties of polypropylene components via a generalized regression neural network. *Polymers for Advanced Technologies, 19*, 73–83.

Silver, H., & Shmoish, M. (2008). Analysis of cognitive performance in schizophrenia patients and healthy individuals with unsupervised clustering models. *Psychiatry Research, 159*, 167–179.

Smith, G. (2007). Random walks in Middle Eastern stock markets. *Applied Financial Economics, 17*, 587–596.

Specht, D. (1991). A general regression neural network. *IEEE Transactions on Neural Networks, 2*, 568–576.

Swicegood, P., & Clark, J. (2001). Off-site monitoring systems for prediction bank underperformance: A comparison of neural networks, discriminant analysis, and professional human judgment. *International Journal of Intelligent Systems in Accounting, Finance and Management, 10*, 169–186.

Tam, C., Tong, T., Lau, T., & Chan, K. (2005). Selection of vertical framework system by probabilistic neural network models. *Construction Management and Economics, 23*, 245–254.

Videnova, I., Nedialkova, D., Dimitrova, M., & Popova, S. (2006). Neural networks for air pollution forecasting. *Applied Artificial Intelligence, 20*, 493–506.

Wang, S. (1995). The unpredictability of standard back propagation neural networks in classification applications. *Management Science, 41*, 555–559.

Wasserman, P. (1993). *Advanced methods in neural computing*. New York: Van Nostrand-Reinhold.

Yu, L., & Lai Wang, S. K. (2009). A neural-network-based nonlinear metamodeling approach to financial time series forecasting. *Applied Soft Computing, 9*, 563–574.

Yumlu, S., Gurgen, F., & Okay, N. (2005). A comparison of global, recurrent and smoothed-piecewise neural models for Istanbul stock exchange (ISE) prediction. *Pattern Recognition Letters, 26*, 2093–2103.

Zhuo, W., Li-Min, J., Yong, Q., & Yan-hui, W. (2007). Railway passenger traffic volume prediction based on neural network. *Applied Artificial Intelligence, 21*, 1–10.