Neuro-Rough Trading Rules for Mining Kuala Lumpur Composite Index

Article · November 2008

3 authors:

Siti Mariyam Shamsuddin
Universiti Teknologi Malaysia
311 PUBLICATIONS   **1,562** CITATIONS

SEE PROFILE

Saiful Hafizah Jaaman
National University of Malaysia
46 PUBLICATIONS   **53** CITATIONS

SEE PROFILE

Maslina Darus
National University of Malaysia
581 PUBLICATIONS   **2,483** CITATIONS

SEE PROFILE

Neuro-Rough Trading Rules for Mining Kuala Lumpur Composite Index

Siti Mariyam Shamsuddin

*Faculty of Computer Science & Information System, Universiti Teknologi
Malaysia, Skudai, Johor, Malaysia*
E-mail: mariyam@utm.my

Saiful Hafizah Jaaman

*Centre for Modelling and Data Analysis, School of Mathematical Sciences
Faculty of Science & Technology, Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor, Malaysia*
E-mail: shj@ukm.my
Tel: 603-89213422, Fax: 603-89254519

Maslina Darus

*Centre for Modelling and Data Analysis, School of Mathematical Sciences
Faculty of Science & Technology, Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor, Malaysia*
E-mail: maslina@ukm.my
Tel: 603-89213423, Fax: 603-89254519

Abstract

Stock market plays a vital role in the economic performance. Typically, it is used to infer the economic situation of a particular nation. However, information regarding a stock market is normally incomplete, uncertain and vague, making it a challenge to predict the future economic performance. In order to represent the market, attending to granular information is required. In recent years, many researches in stock market prediction are conducted using diverse Artificial Intelligence approaches. These artificial applications have shown superior prediction results. As such, in this study, a prediction enhancement alleged as Neuro-Rough (NR) is proposed to forecast the Kuala Lumpur Stock Exchange Composite Index (KLCI) movements. NR hybridizes high generality of artificial neural network (ANN) and rules extraction ability of rough sets theory (RST) by demonstrating the capability of simplifying the time series data and dealing with uncertain information. Features of stock market data are extracted and presented in a set of decision attribute to the NR systems. The length of the stock market trend is used to assist the process of identifying the trading signals. A pilot experiment is conducted to discover the best discretization algorithm and ANN structure. NR is implemented in a trading simulation and its effectiveness is verified by analyzing the classifier output against the information provided in Bursa Malaysia's annual reports. The experiments using 10 years training and testing data reveal that NR achieves an accuracy of 70% with generated annual profit in trading simulation of 74.33%.

Keywords: Neuro-Rough, Artificial Neural Network, Granular Information, Stock Market Prediction

1. Introduction

Stock market is one of the places where people pursue their dreams of obtaining wealth by buying and selling shares. Thus, any knowledge of future information regarding the price behaviour of a particular share will definitely ensure large profits in the market. This motivates the development of numerous analytical methods to envisage and benefit from market's direction. In predicting the stock market movement, technical analysis has been the common approach employs, approximately 90% of major stock traders use this method in their investment analysis. Technical indicators such as the moving average, trading bands, Bollinger bands, volume, Moving Average Convergence / Divergence (MACD), Relative Strength Index (RSI) and others have been widely used to analyze the trend of market direction via chart presentations. As many traders use technical analysis to get insight into the market, this has invited the advancement of many applications to implement technical analysis, such as WinChart from Straits Index (M) Sdn Bhd [1] and MetaStock from Equis International Inc.

Contrarily, fundamental analysis is a more complex stock market prediction method involving an in-depth analysis on the performance and profitability of a company to determine its share price. This method assumes that current share (and future) price depends on its intrinsic value and anticipated return on investment. Though fundamental analysis assumes that new information about a company will affect the movement of its share price, fundamental analysis is difficult to implement for it requires real and reliable information of a company such as economic conditions, financial reports and company's competitive strength.

Moreover, over short period of time traditional time series forecasting provides reasonable accuracy in stock market prediction. This method attempts to model a nonlinear function by a recurrence relation derived from past values. For example the Box-Jenkins which is a complicated process of fitting data to appropriate model parameters. The equations used in this model contain past values of moving averages and prices. However, the conventional analysis methods discussed above are limited in their areas of application and time flexibility [2]. Hence, of late the artificial intelligence (AI) approaches are applied to stock market prediction as the problems and the mathematical calculations get even more complex. Such approaches involve genetic algorithm (GA), rough set theory (RST), artificial neural network (ANN) and fuzzy logic (FL).

In this research, the aim is to prove the feasibility of Neuro-Rough (NR) in predicting the trend of Kuala Lumpur Composite Index over time. By dividing the trend of KLCI close price into upward and downward patterns, features of each pattern such as the slope and signal to noise (SNR) are extracted. It is assumed that the strength of certain trend is determined by its slope and fluctuations with the aid of past trends (slopes). NR is a hybridized method which combines the generalization faculty of ANN and the rule reduction capability of RST. The extracted features of each pattern discussed above are divided into training and testing sets. Training set is applied to NR for generalization and rules reduction while testing sets are used to validate the NR classifier. NR classifier is employed in the trading simulation and the profit generated is compared against profits generated using other available stock market prediction tools that also employ knowledge based approach. The classification accuracy is taken into account to evaluate the efficiency of the proposed model.

2. Neuro-Rough (NR)

Artificial neural network (ANN) and rough set theory (RST) have both been employed in stock market prediction. Based on previous researches, both methods have shown its capability in this application. ANN has been used to predict the stock market for the past few years and is still being investigated by many researchers with the objective of achieving higher and accurate prediction. Researchers in the field study various architectures and develop many different new models. However, none of the available model can be exploited to various applications without sacrificing the architecture's structure.

ANN model is a practical model for a particular area, but not for other applications. The only drawback in ANN representation is its complexity in interpreting the learned solutions in the architecture. This situation has made it difficult to facilitate human inspection or understanding.
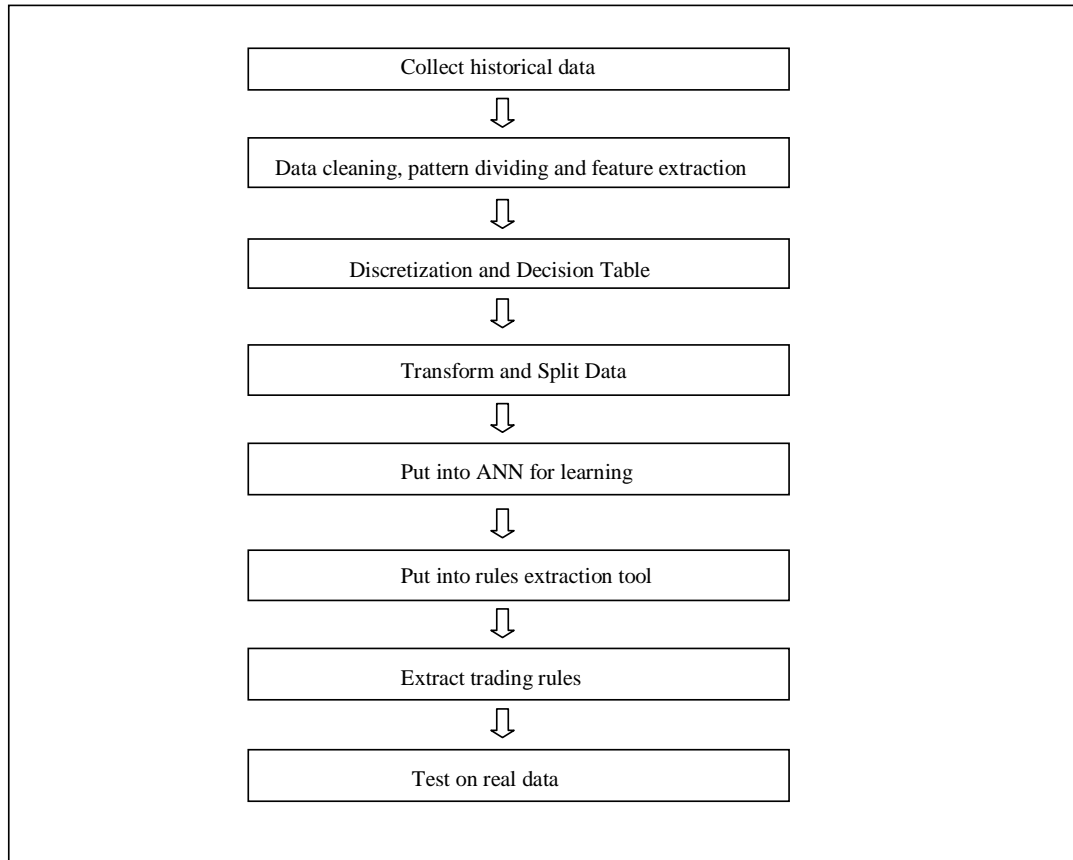
Besides ANN, RST has also been used in the application of stock market prediction. Like ANN, RST has also shown successful results with better accuracy. Many RST models have been developed for different areas. These include Rough Set Expert System (RSES) in financial investment and ProbRough in database marketing, RST dominance is in relation to business failure prediction, to name a few. RST is employed widely for it is based solely on the original data and is able to take on both qualitative and quantitative presentations providing results that are much easier to comprehend. Also existed are studies that combine both of these methods in view of the limitations in both architectures. For instance, RST discovers important facts hidden in the data and expressed them in natural language of decision rules overcoming the obstacle in ANN. Meanwhile, the strong ability of ANN to generalize has made it appropriate for the stock data mining of high noise. ANN is used to learn the information table then rough set is used to extract rules from the knowledge of neural network.

Based on the successful results given by RST, ANN or even the hybrid version of both methods in stock market prediction, thus it is worthy to explore on the hybridization of these techniques, that stock market prediction employing hybrid version of ANN and RST, the so-called NR is applicable and promising.

3. The Framework of the Model

NR is a hybridized method which combines the generalization faculty of ANN and the rule reduction capability of RST. The extracted features of each pattern discussed above are divided into training set and testing set. Training set is applied to NR for generalization and rules reduction while testing set is to validate the NR classifier. NR classifier is used in a trading simulation, and the generated profit is compared with the profit generated by available stock market prediction tools which also employ knowledge based approach. The classification accuracy is taken into account to evaluate the efficiency of the proposed model.

The framework of the proposed NR model in stock market data classification and analysis is illustrated in Figure 1. For predicting the KLCI returns, raw stock market data are collected consisting of KLCI daily close, high, low and open prices and its volume. These raw data are de-noised using low pass filter to remove the additive noise. After cleaning the raw data, it is partitioned into patterns based on the trend of the close price (*ie*. upward and downward). For each pattern, the identified features; the slope, SNR and length of the pattern are extracted. These extracted features form the information table in this study.

Figure 1: Proposed Problem Model of NR

```
┌─────────────────────────────────────────────────────────────┐
│                                                               │
│          ┌─────────────────────────────────────┐             │
│          │        Collect historical data      │             │
│          └─────────────────────────────────────┘             │
│                            ⇩                                  │
│          ┌─────────────────────────────────────┐             │
│          │ Data cleaning, pattern dividing and  │             │
│          │         feature extraction          │             │
│          └─────────────────────────────────────┘             │
│                            ⇩                                  │
│          ┌─────────────────────────────────────┐             │
│          │    Discretization and Decision Table │             │
│          └─────────────────────────────────────┘             │
│                            ⇩                                  │
│          ┌─────────────────────────────────────┐             │
│          │         Transform and Split Data     │             │
│          └─────────────────────────────────────┘             │
│                            ⇩                                  │
│          ┌─────────────────────────────────────┐             │
│          │         Put into ANN for learning    │             │
│          └─────────────────────────────────────┘             │
│                            ⇩                                  │
│          ┌─────────────────────────────────────┐             │
│          │        Put into rules extraction tool │             │
│          └─────────────────────────────────────┘             │
│                            ⇩                                  │
│          ┌─────────────────────────────────────┐             │
│          │           Extract trading rules      │             │
│          └─────────────────────────────────────┘             │
│                            ⇩                                  │
│          ┌─────────────────────────────────────┐             │
│          │            Test on real data         │             │
│          └─────────────────────────────────────┘             │
│                                                               │
└─────────────────────────────────────────────────────────────┘
```

Each derived attribute in the information table is discretized to simplify the knowledge representation. In this study, three discretization algorithms are selected; boolean reasoning algorithm, equal frequency binning and χ^2 algorithm. Consequently, the decision attribute is selected from the derived features and the resulting table is the decision table in RST. Prior to training and testing phase, each object in the decision table is transformed to feed into ANN. Subsequently, the training data are split from the testing data. ANN with backpropagation algorithm is utilized as the learning algorithm. Training objects are fed into ANN and the knowledge inside the training set is discovered. The ascertained knowledge enable the trained ANN to classify accurately the objects in the training dataset and any new objects in the testing phase. Objects learned by ANN are fed into ROSETTA (a roughest toolkit) for recognizing patterns and calculating reducts. Reducts or a minimal selection of attributes are calculated from training data to generate the trading rules.

These trading rules are exploited to classify the testing dataset. The trading rules can be used to explicate the knowledge learned by ANN from decision table. Besides, it can be applied to propose a trading decision based on the assessment of the generated rules. In this study, a pilot experiment with real KLCI data is examined to discover the suitable discretization algorithm, ANN network architecture, network training parameters and proper reduction algorithm.

In stock market prediction, the preprocessing stage can be regarded as the most essential part. The raw data from the stock market time series cannot be employed unless it has been preprocessed. Raw stock market data is noisy, vague and uncertain because of anonymous attributes due to non-trading days causing missing data. In this study, incomplete objects are eliminated from the stock market database.

In stock market time series database, the closing price of each trading day is composed from daily random fluctuations as well as long term trend. Therefore, it is necessary to denoise this data to

produce clean data. It is assumed that the raw data, $a_{raw}(n)$ is composed of long trend signal $a(n)$ and noise $e(n)$ with additive nature as shown in Equation 1 [3] [4]:

$$a_{raw}(n) = a(n) + e(n) \tag{1}$$

The cleaning operation produces $\hat{a}(n)$ to estimate the long-term signal $a(n)$. This signal is stable, deterministic, and influenced by relatively few factors.

For an investor, the length (duration) of a particular stock's trend is deemed important since it triggers a buying or a selling action for that particular stock. In order to find the lengths, this study employs the *turning point* of the data trend method which determines the end of the forward trend and the start of the second trend using the threshold, *d* as 5 days, which is the normal weekly trading period in Bursa Malaysia.

The next stage is to measure the intervals of different trends. These intervals can be approximated with linear function since the trend data in each interval is consistent. The length of the intervals and the target attribute are the most important features in this study as well as the slope and SNR. In $T_e = \{t_0, ..., t_{Ne}\}$, the length of interval is the length of the quotation, and will be the decision attribute in the rough set decision table. Length is calculated initially, and it is defined as the different of two sequential extremum that represent a particular interval [3] [4].

$$Length = t_{i+1} - t_i \text{ where } 0 \le i \le N_e - 1 \tag{2}$$

Then the Slope of the pattern is evaluated. For each interval, the initial and the final value of $a(n)$ are exploited to calculate the slope as in Equation (5) [3] [4].

$$\alpha_i = \frac{\hat{a}(t_{i+1}) - \hat{a}(t_i)}{t_{i+1} - t_i} \tag{3}$$

SNR is another important feature that articulates the fluctuations of the series data. A high SNR value indicates that the series is unstable and influenced by various parameters and different factors. Low values of SNR indicate stable series which are influenced by a limited number of factors. SNR is calculated using the Equation (6) and Equation (7) [3] [4].

$$SNR_i = \sqrt{\frac{\int_i^{i+1} \frac{\varepsilon^2(t)}{\hat{a}^2(t)}}{t_{i+1} - t_i}} \tag{4}$$

where,

$$\varepsilon(t) = a(t) - \hat{a}(t) \tag{5}$$

$a(t)$ is the original data and $\hat{a}(t)$ is the cleaned data.

The Slope, SNR and Length are the considerations in this study since these features are the input and output for the NR model.

3.1. Feature Discretization

Discretization is a process of quantizing continuous attributes. There is a very slim chance that a new object can be recognized by matching its attribute value vector with rows in decision table. Hence, real value attributes in decision table is discretized to achieve a higher quality of classification. To articulate the knowledge in terms of rules, attributes in the decision table are represented in discrete form. The advantages of employing discrete values include more concise representation and specification of the attributes, easier to use and comprehend which lead to improve prediction accuracy. This research adopts the Equal Frequency Binning and χ^2 algorithm discretization methods as described in the literature.

3.2. Decision Table

Stock market prediction involves forecasting future development of a time series based on the development of the time series up until the present time. The information of time sequences inside the data is essential. When applying RST, these time series data cannot be directly represented since RST disregard the time sequence in the database. Thus, the data needs to be converted into RST objects then preserve the interval relationship of each event / object. Baltzersen [5] presented two methods concerning this problem, the mobile window and columnizing methods. Mobile window method moves a window along the time series and the falling data points into the window are transferred to a RST object. In columnizing method, the time series are organized in columns such that each row represents an object and different point in time. Each column is an economic indicator or extracted feature [5].

3.3. Data Transformation and Partition

Data in discrete form is suitable for RST, but it is not the case of ANN. ANN deals with continuous value. Thus, each object in the decision table is transformed into binary (0 and 1) version that can be fed into ANN for generalization. Data transformation largely depends on the distribution of each attribute in the discretization stage. In this study, number of interval generated in the discretization process is used to determine the number of input and output nodes in ANN. The result of the data transformation is that instead of the interval for each attribute, it is transformed to take the value of 0 and 1 which are fed into ANN. Binary data that represents decision attribute is the target during ANN training process.

Training set is used for NR model development while testing set is adopted for evaluating the forecasting ability of NR model. Hence, objects in the decision table (both transformed and original) are divided into two sets; training and testing sets. The proposed NR model in this study uses 10 years KLCI data.

3.4. ANN Generalization

Discretization increases the generality of the objects in decision table. However, stock market time series data contain much noise therefore, ANN is used to improve the generalization of these objects. ANN network architecture consists of one or more hidden layers and different hidden nodes. To date, guideline for determination of these parameters are either heuristic or based on simulation derived from limited experiments [6]. Hence, in this study the number of hidden layers is fixed while the number of hidden nodes is randomly selected. In addition to the network architecture, learning rate and momentum need to be determined before ANN can be trained taking any value between 0 and 1. As noted, it is impossible to do an exhaustive search to find the best combinations of these parameters [6]. Theoretically, a learned ANN should be able to predict the objects that have been used in ANN training accurately since the knowledge have been captured. Thus, if the predicted output is different from the target output for certain objects, these objects are considered as awful objects and are removed from the decision table.

3.5. Standard Voting Classifier

A Standard Voting Classifier is developed for rules classification to predict the KLCI. The inputs of the classifier are exported rules from ROSETTA and the extracted features. For each testing object, the attribute value is matched with each antecedent rule in the rules set. A rule is fired if its antecedent is matched with the presented object. After the matching procedures are done, the election process is generated to decide the final decision value. In the election process, each obtained fired rules need to be cast with a number of votes in the decision value. The vote acts as the support for the particular decision value. These support values are equal to the number of objects in the training set that have the same antecedent for a particular decision class. After matching the entire rules, the decision value with

the highest vote is suggested to be the most likely decision value of the said object. Each object in the testing set is generated through this firing step and an election process is done to determine the final decision value. Once all the objects are classified, the results are compared with the real decision class of the testing objects and the accuracy of the generated rules is determined.

4. Experimental Results and Discussion

The prediction results for each data sample employing different reducts calculation and rules generation algorithm are summarized in Tables 2 and 4. The classification accuracy of each data sample is shown to be between 45% and 70%. In addition, results show that NR model can predict objects from different decision classes indicating that NR model is able to learn comprehensive knowledge produced by all objects in different decision classes. NR is not converged to only particular set of objects in the decision table.

Table 2:   Classification Accuracy of Discretization Algorithms

	Boolean Reasoning	EFB (4 bin)	Chi2
Matlab ANN	56.43%	62.14%	65.00%
Developed ANN	61.43%	54.29%	55.00%

Table 3:   The Number of Learned Objects for Discretization

	Boolean Reasoning	EFB (4 bin)	Chi2
Matlab ANN	90	81	99
Developed ANN	97	88	102

Table 4:   Best Network Architecture for Discretization

	Boolean Reasoning	**EFB (4 bin)**	**Chi2**
Matlab ANN	15-20-3 (learning rate 0.1, momentum 0.8)	12-14-15-3 (learning rate 0.1, momentum 0.8)	19-19-50-3 (learning rate 0.1, momentum 0.8)
Developed ANN	15-20-3 (learning rate 0.01, momentum 0.9)	12-14-15-3 (learning rate 0.01, momentum 0.9)	19-12-29-3 (learning rate 0.01, momentum 0.8)

4.1. Simulation for Buying and Selling

Assessing the performance of a stock market prediction system is not an easy task. Armano [8] mentioned that *percentage of classification accuracy do not have a direct economic meaning.* Furthermore, evaluation of any proposed model depends on the strategy of the trader or investor [7]. Even though the same underlying model is applied, different strategy may generate different profit.

Nevertheless, in order to verify the effectiveness of the NR model, a simulation of buying and selling of KLCI is done. NR model aims to predict the Length (duration) of KLCI identified trends. Last *et al* [4] stated that long interval or Length is an indication of a stable market. Thus, in this study, decision value of the classifier with shortest Length indicate an *unstable market trend* while decision value with longest Length means the market is in *a stable trend*. Using this information, trading strategy such as below can be constructed:

If decision="[17,)", then buy (long or short depend on the Slope)*
If decision="[,11)", then sell*
If decision="[11,17)", hold or no trading
If no decision from classifier, sell or no trading

If the *Slope* of the trend at a point is different from the *Slope when a trade is generated,* and this scenario lasts for two trading days consecutively an immediate sell signal is triggered. In this simulation, initial seed money of RM1000 is used for the investment on KLCI assuming no transaction cost. This simulation aims to view the profit made employing the NR model using KLCI data from January 14, 1994 to December 29, 2003 partitioning into training and testing data set. The final balance in the investment is the initial seed money plus any gains or any losses incurred for the generated trade during the simulation.

The highest annual profit using rules generated from each data sample is shown in Table 5. The highest final balance of the seed money is RM8408.15722 while the lowest final balance of the seed money at the end of the simulation is RM2872.2088.

Table 5:   The Highest Annual Profit Generated From Different Data Samples

	Boolean Reasoning	**EFB (4 bin)**	**Chi2**
Matlab ANN	29.5% (rules set from Holtes's IR)	50.29% (rules set from Dynamic exhaustive calculation)	74.33% (rules set from Johnson's algorithm)
Developed ANN	18.78% (rules set from Holtes's IR)	61.36% (rules set from Dynamic exhaustive calculation)	29.5% (rules set from Genetic algorithm)

5. Conclusion

In this research, an enhancement approach which combined ANN and RST is proposed to predict the movements of the Kuala Lumpur Composite Index. The so-called NR approach combines the high generalization of ANN and the rule reduction capability of RST. Granulation information for composite index is generated. The information granule is the objects in the decision table which consists of attributes Slope1, Slope2 and SNR. NR classifier is built to calculate the classification accuracy and used in trading simulation. In this study it is shown that NR is able to learn fully knowledge of KLCI historical data. By using ANN to generalize the data, reducts are calculated and rules are generated. The generated rules represent the real knowledge inside KLCI historical data without the influence of noise or daily random fluctuation. After discretization and ANN configuration, NR achieves an average of 65% classification accuracy. The highest classification accuracy among the generated rule sets is 70% which is the rules generated by Johnson's algorithm reduction. In the simulation for buying and selling operation, rule set with 70% classification accuracy has generated 74.33% of annual profit giving a profit of RM7408.1572 based on initial seed money of RM1000.

Results of classification and simulation have convinced that NR approach can learn knowledge in stock market time series and give profit to investors. It can be an alternative tool for investor to predict the stock market outcomes.

Acknowledgement

References

[1]   StraitsIndex.com. 2003. "KLSE composite index technical analysis online tutorial."
       http://www.straitsindex.com/etutorial/main.htm

[2]   Y.-Q. Zhang, S. Akkaladevi, G. Vachtsevanos, T. Y. Lin. 2002. "Granular neural web agents
       for stock prediction". *Soft Computing* 6; 406-413 © Springer-Verlag DOI 10.1007/S00500-002-
       0193-7.

[3]   Xiao-Ye Wang, Zheng-Ou Wang. 2002. "Stock Market Time Series Data Mining Based On
       Regularized Neural Network and Rough Set." Proceedings of the First International
       Conferences on Machine Learning and Cybernetics, Beijing.

[4]   Last M, Klein Y and Kandel A. 2001. "Knowledge Discovery in Time Series Database." IEEE
       Transactions on Systems, Man, and Cybernetics-part b, VAL31(1).

[5]   Baltzersen J.K. 1996. "An Attempt to Predict Stock Market Data." Diploma Thesis. Knowledge
       System Group, Department of Computer System and Telematics, Norwegian Institute of
       Technology.

[6]   Guoqiang Zhang, B. Eddy Patuwo, Michael Y. Hu. 1998. "Forecasting with artificial neural
       network: The state of the art." *International Journal of Forecasting* 14; 35-62.

[7]   Yao J.T., Tan C.L. and Poh H.L. 1999. "Neural Network for Technical Analysis: A Study on
       KLCI." *International Journal of Theoretical and Applied Finance,* Vol. 2 No. 2, 221-241.

[8]   G. Armano, A. Murru and F. Roli. 2002. "Stock Market Prediction by a Mixture of Genetic-
       Neural Experts." *International Journal of Pattern Recognition and Artificial Intelligence* Vol.
       16, No. 5; 501-526.