International Journal of Theoretical and Applied Finance
Vol. 2, No. 2 (1999) 221–241

NEURAL NETWORKS FOR TECHNICAL ANALYSIS: A STUDY ON KLCI

JINGTAO YAO, CHEW LIM TAN and HEAN-LEE POH
School of Computing, National University of Singapore,
Lower Kent Ridge Road Singapore 119260
E-mail: [yaojt, tancl]@comp.nus.edu.sg

Received 7 October 1998

This paper presents a study of artificial neural nets for use in stock index forecasting. The data from a major emerging market, Kuala Lumpur Stock Exchange, are applied as a case study. Based on the rescaled range analysis, a backpropagation neural network is used to capture the relationship between the technical indicators and the levels of the index in the market under study over time. Using different trading strategies, a significant paper profit can be achieved by purchasing the indexed stocks in the respective proportions. The results show that the neural network model can get better returns compared with conventional ARIMA models. The experiment also shows that useful predictions can be made without the use of extensive market data or knowledge. The paper, however, also discusses the problems associated with technical forecasting using neural networks, such as the choice of "time frames" and the "recency" problems.

Keywords: Neural network; financial analysis; stock market; prediction.

1. Introduction

People tend to invest in equity because of its high returns over time. Stock markets are affected by many highly interrelated economic, political and even psychological factors, and these factors interact with each other in a very complex manner. Therefore, it is generally very difficult to forecast the movements of stock markets.

Refenes *et al.* [24] indicate that conventional statistical techniques for forecasting have reached their limitation in applications with nonlinearities in the data set. Artificial neural network, a computing system containing many simple nonlinear computing units or nodes interconnected by links, is a well-tested method for financial analysis on the stock market. The research fund for neural network applications from financial institutions is the second largest [29]. For example, the America's Defense Department invests $400 million in a six-year project, and Japan has a $250-million ten-year-neural-computing project [9]. Neural networks have been shown to be able to decode nonlinear time series data which adequately describe the characteristics of the stock markets [17]. Examples using neural networks in equity market applications include forecasting the value of a stock index [13, 24, 32], recognition of patterns in trading charts [27], rating of corporate bonds [8], estimation of the market price of options [19, 16], and the indication of trading signals of selling and buying [5, 20], etc. Feed-forward backpropagation networks [25] as discussed in Sec. 3 are the most commonly used networks and meant for the widest variety of applications [31].

This paper shows that without the use of extensive market data or knowledge useful prediction can be made and significant paper profit can be achieved. It begins with the general discussion on the possibilities of equity forecasting in an emerging market. It is followed by a section on neural networks. Subsequently, a section is devoted to a case study on the equity forecasting in one of the largest emerging markets, pointing to the promises and problems of such an experiment. Finally, a conclusion which also discusses areas for future research is included at the end of the paper.

2. Forecasting the Stock Market

Prediction in stock market has been a hot research topic for many years. Generally, there are three schools of thought in terms of the ability to profit from the equity market. The first school believes that no investor can achieve above average trading advantages based on the historical and present information. The major theories include the *Random Walk Hypothesis* and the *Efficient Market Hypothesis* [21]. The Random Walk Hypothesis states that prices on the stock market wander in a purely random and unpredictable way. Each price change occurs without any influence by past prices. The Efficient Market Hypothesis states that the markets fully reflect all of the freely available information and prices are adjusted fully and immediately once new information becomes available. If this is true then there should not be any benefit for prediction, because the market will react and compensate for any action made from these available information. In the actual market, some people do react to information immediately after they have received the information while other people wait for the confirmation of information. The waiting people do not react until a trend is clearly established. Because of the efficiency of the markets, returns follow a random walk. If these hypotheses come true, it will make all prediction methods worthless. Taylor [28] provides compelling evidence to reject the random walk hypothesis and thus offers encouragement for research into better market prediction.

The research done here would be considered a violation of the above two hypotheses above for short-term trading advantages in, Kuala Lumpur Stock Exchange (KLSE for short hereafter), one of the most important emerging markets, which is considered by some Malaysian researchers such as Yong [33, 34] to be "less random" than the mature markets. In fact, even the stock market price movements of United States [12] and Japan [1] have been shown to conform only to the weak form of the efficient market hypothesis. Also, Solnik [26] studied 234 stocks from eight major European stock markets and indicated that these European stock markets exhibited a slight departure from random walk.

The second school's view is the so-called fundamental analysis. It looks in depth at the financial conditions and operating results of a specific company and the underlying behavior of its common stock. The value of a stock is established by analysing the fundamental information associated with the company such as accounting, competition, and management. The fundamental factors are overshadowed by the speculators trading. In 1995 US$1.2 trillion of foreign exchange swapped hands on a typical day [10]. The number is roughly 50 times the value of the world trade in goods and services which should be the real fundamental factor.

Technical analysis, on the other hand, assumes that the stock market moves in trends and these trends can be captured and used for forecasting. Technical analysis belongs to the third school of thought. It attempts to use past stock price and volume information to predict future price movements. The technical analyst believes that there are recurring patterns in the market behavior that are predictable. They use such tools as charting patterns, technical indicators and specialized techniques like Gann lines, Elliot waves and Fibonacci series [22]. Indicators are derived from price and trading volume time series. In most cases, there are five time series for a single share or market index. These five series are open price, close price, highest price, lowest price and trading volume. Analysts monitor changes of these numbers to decide their trading. There are several rules such as "When the 10-day moving average crosses above the 30-day moving average and both moving averages are in an upward direction it is the time to buy"; "When the 10-day moving average crosses below the 30-day moving average and both moving averages are directed downward it is time to sell", etc. used in trading floor. Unfortunately, most of the techniques used by technical analysts have not been shown to be statistically valid and many lack a rational explanation for their use [7]. As long as past stock prices and trading volumes are not fully discounted by the market, technical analysis has its value on forecasting.

To maximize profits from the stock market, more and more "best" forecasting techniques are used by different traders. Nowadays, traders no longer rely on a single technique to provide information about the future of the markets but rather use a variety of techniques to obtain multiple signals. Neural networks are often trained by both technical and fundamental indicators to produce trading signals.

Fundamental and technical analysis could be simulated in neural networks. For fundamental methods, retail sales, gold prices, industrial production indices, and foreign currency exchange rates, etc. could be used as inputs. For technical methods, the delayed time series data could be used as inputs. In this paper, a technical method which takes not only the delayed time series data as inputs but also the technical indicators.

3. Neural Network and its Usage in the Stock Market

3.1. *Neural networks*

A neural network is a collection of interconnected simple processing elements. Every connection of neural network has a weight attached to it. The backpropagation algorithm [25] has emerged as one of the most widely used learning procedures for multi-layer networks. The typical backpropagation neural network usually has an input layer, some hidden layers and an output layer. Figure 1 shows a one-hidden-layer neural network. The units in the network are connected in a feedforward manner, from the input layer to the output layer. The weights of connections have been given initial values. The error between the predicted output value and the actual value is backpropagated through the network for the updating of the weights. This is a supervised learning procedure that attempts to minimize the error between the desired and the predicted outputs.

The output value for a unit j is given by the following function:

$$O_j = G\left(\sum_{i=1}^{m} w_{ij}x_i - \theta_j\right), \tag{3.1}$$

where x_i is the output value of the ith unit in a previous layer, w_{ij} is the weight on the connection from the ith unit, θ_j is the threshold, and m is the number of units in the previous layer. The function $G(\,)$ is a sigmoid hyperbolic tangent function:

$$G(z) = \tanh(z) = \frac{1 - e^{-z}}{1 + e^{-z}} \tag{3.2}$$

$G(\,)$ is a commonly used activation function for time series prediction in backpropagation networks [5, 17].

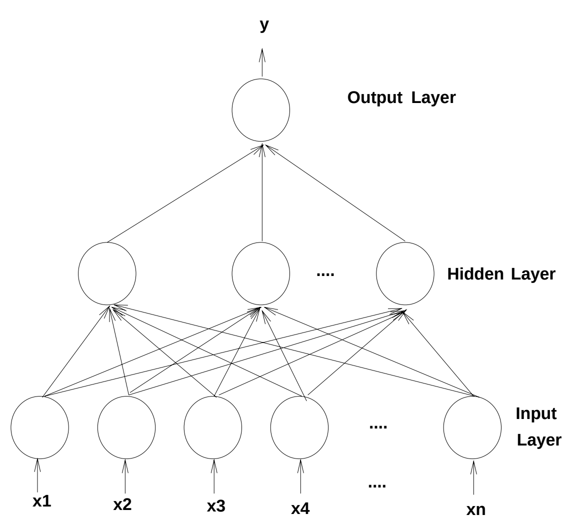

Fig. 1. A neural network with one hidden layer.

3.2. *Time series forecasting with neural networks*

Based on the technical analysis, past information will affect the future. So, there should be some relationship between the stock prices of today and the future. The relationship can be obtained through a group of mappings of constant time interval. Assume that u_i represents today's price, v_i represents the price after ten days. If the prediction of a stock price after ten days could be obtained using today's stock price, then there should be a functional mapping u_i to v_i, where

$$v_i = \Gamma_i(u_i)\,. \tag{3.3}$$

Using all $(u_i,\, v_i)$ pairs of historical data, a general function $\Gamma(\,)$ which consists of $\Gamma_i(\,)$ could be obtained.

$$v = \Gamma(u)\,. \tag{3.4}$$

More generally, $\vec{u}$ which consists of more information in today's price could be used in function $\Gamma(\,)$. Neural networks can simulate all kinds of functions, so they also can be used to simulate this $\Gamma(\,)$ function. The $\vec{u}$ is used as the inputs to the neural network.

There are three major steps in the neural network based forecasting proposed in this research: preprocessing, architecture, and postprocessing. In *preprocessing*, information that could be used as the inputs and outputs of the neural networks are collected. These data are first normalized or scaled in order to reduce the fluctuation and noise. In *architecture*, a variety of neural network models that could be used to capture the relationships between the data of inputs and outputs are built. Different models and configurations using different training, validation and forecasting data sets are experimented. The best models are then selected for use in forecasting based on such measures as out-of-sample hit rates. Sensitive analysis is then performed to find the most influential variables fed to the neural network. Finally, in *postprocessing*, different trading strategies are applied to the forecasting results to maximize the capability of the neural network prediction.

3.3. *Measurements of neural network training*

The *Normalized Mean Squared Error* (NMSE) is used as one of the measures to decide which model is the best. It can evaluate and compare the predictive power of the models. The definition of NMSE is

$$\text{NMSE} = \frac{\sum_k (x_k - \hat{x}_k)^2}{\sum_k (x_k - \bar{x}_k)^2}\,, \tag{3.5}$$

where x_k and $\hat{x}_k$ represent the actual and predicted values respectively, and $\bar{x}_k$ is the mean of x_k. Other evaluation measures include the calculation of the correctness of signs and gradients. Sign statistics can be expressed as

$$\text{Sign} = \frac{\sum s_k}{N}\,, \tag{3.6}$$

where N represents the number of patterns in a testing set and s_k is a segment function which can be expressed as

$$s_k = \begin{cases} 1 & x_k \hat{x}_k > 0 , \\ 1 & x_k = \hat{x}_k = 0 , \\ 0 & \text{otherwise} . \end{cases} \tag{3.7}$$

Here Sign represents the correctness of signs after normalization. Similarly, directional change statistics can be expressed as

$$\text{Grad} = \frac{\sum g_k}{N} , \tag{3.8}$$

where

$$g_k = \begin{cases} 1 & (\hat{x}_{k+1} - x_k) = 0 \text{ and } (x_{k+1} - x_k) = 0 , \\ 1 & (x_{k+1} - x_k)(\hat{x}_{k+1} - x_k) > 0 \ , \\ 0 & \text{otherwise} . \end{cases} \tag{3.9}$$

NMSE is one of the most wildly used measurements. It represents the fit between the neural network predictions and the actual targets. However, a prediction that follows closely the trend of the actual target would also result in a low NMSE. For pattern recognition, it is a very important signal. We argue that although NMSE is a very important signal for pattern recognition, it may not be the case for trading in the context of time series analysis. We can use a simple example to explain our argument.

The "Target" line in Fig. 2 is representive of a real financial time series though the figures are rounded for simplicity in illustration. In addition, four forecasting time series are artificially created as shown in Table 1 based on the pattern of the "Target" line according to the following criteria. "MissBig" series is a forecasting which always fits the target except for the big changes in the "Target" line. "MissSmall" series is a forecasting which always fits the target except for the small changes. "Trend" series which is also shown in Fig. 2 is a forecasting which accentuates the trend and is thus always correct in terms of trend. "Versus" series is a forecasting which is always wrong in terms of trend.

The performance of all the four forecasts with respect to the "Target" is shown in Table 2. "MissBig" and "MissSmall" series are the best in terms of goodness of fit in all four measure, i.e. Mean Squared Error(MSE), NMSE, Average Error(AE)

Table 1. Values of the five time series.

Target	10	11	8	9	6	7	6	12	11	12	11	12	6	7	6	8
Trend	10	16	5	10	4	8	5	18	9	13	10	13	5	8	7	8
MissBig	10	11	8	9	10	7	5	12	11	12	11	12	13	7	6	8
MissSmall	10	11	8	9	6	5	6	12	11	12	13	12	6	7	6	8
Versus	10	9	10	7	8	6	9	8	11	10	11	10	11	7	8	8

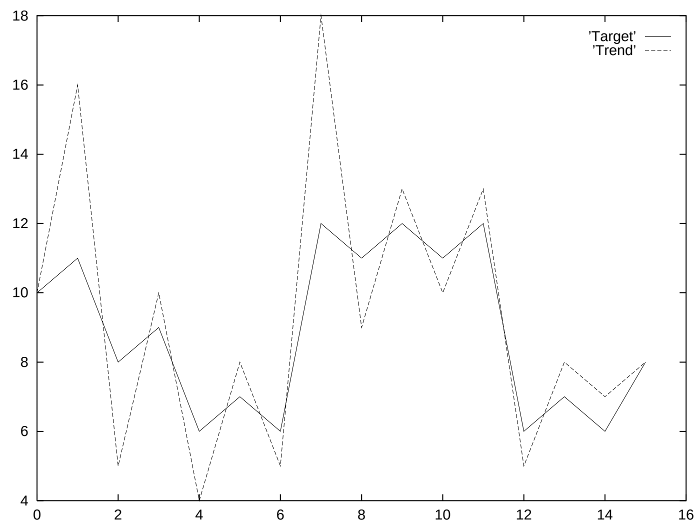

Fig. 2. The Target time series and Trend forecasting series.

Table 2. Statistics of Target series its forecasts. MSE: Mean Sqiared Error; AE Average Error; AAE: Absolute Average Error; After: Seed Money (initially 1000 units) left after all trading.

	MSE	NMSE	AE	AAE	Grad	After	Correct Trading
Target	0.0000	0.000000	0.0000	0.0000	100%	5345.45	100%
Trend	5.4375	1.014577	−0.4375	1.6875	100%	5345.45	100%
MissBig	4.1250	0.769679	−0.6250	0.7500	73.3%	1309.09	63.6%
MissSmall	0.5000	0.093294	0.0000	0.2500	73.3%	3300.00	63.6%
Versus	4.9375	0.921283	−0.0625	1.6875	0%	149.65	0%

and Absolute Average Error(AAE). To test the profit, we assume that a 1000 unit seed money is at hand before trading and a Strategy 2 (Eq. (4.13) to be discussed in Sec. 4) is used in trading. The goodness of fit happens to be the worst for "Trend" series. But its profit is as good as that of "Target" series which means that the forecast follows exactly its target. The reason for the good performance of this series is that its forecast is always correct in trend and thus the trading based on them will always be right. It shows that Grad is a better indicator of the quality of forecasting for trading purposes as we can profit in the two "Miss" series. Grad can show how good the forecast trend is, which is very useful for trading purposes. Similarly, when the inputs are the changes of levels instead of the actual levels, then Sign also points to the accuracy of the forecast trends, and therefore it could be useful for trading. As the aim of financial forecasting is to maximize profit, we suggest that NMSE should not be used as the unique criterion of forecasting performance.

4. A Case Study on the Forecasting of the KLCI

The Kuala Lumpur Composite Index (KLCI) is calculated on the basis of 86 major Malaysian stocks. It is capitalization-weighted by paäsche formula and has a base level of 100 as of 1977. It may be regarded as the Malaysian Dow Jones Index. As of March 13, 1995, 492 companies have been listed in KLSE. It has only ten years of history, so there are not enough fundamental data that could be used for forecasting. Besides, the KLSE is considered a young and speculative market, where investors tend to look at price movements, rather than the fundamentals.

Due to the high returns in emerging markets, investors are attracted to enhance their performance and diversify their portfolios [11]. KLSE is considered the second largest non-Japan Asia market in terms of capitalization (US$ 202.8 billion). KLSE is a typical emerging market in Asia and hence the research on this market may contribute to the global investment.

In this paper, a technical method is adopted which takes not only the delayed time series data as inputs but also the technical indicators. Neural networks are trained to approximate the market values which may reflect the thinking and behavior of some stock market traders, or so to speak.

Forecasting of stock indices is to find the nonlinear dynamic regularities between stock prices and historical indices together with trading volumes times series. Due to the nonlinear interaction among these variables, it is very difficult to find the regularities but the regularities do exist. This research is aimed to find the hidden relationship between technical indicators and future KLCI through a neural network model.

Different indicators are used as the inputs to a neural network and the index of stock is used to supervise the training process, in order to discover implicit rules governing the price movement of KLCI. Finally, the trained neural network is used to predict the future levels of the KLCI. The technical analysis method is used commonly to forecast the KLCI, the buying and selling point, turning point, and the highest, lowest point, etc. When forecasting by hand, different charts will be used by analysts in order to predict the changes of stocks in the future. Neural networks could be used to recognize the patterns of the chart and the value of index.

4.1. *Data choice and pre-processing*

The daily data from Jan 3, 1984 to Oct 16, 1991 (1911 data) are used on the first trial. Figure 3 shows the basic movement of the stock exchange composite index.

Technical analysts usually use indicators to predict the future. The major types of indicators are *moving average* (MA)*, momentum* (M)*, Relative Strength Index* (RSI) *and stochastics* (%K)*, and moving average of stochastics* (%D). These indicators can be derived from the real stock composite index. The target for training the neural network is the actual index.

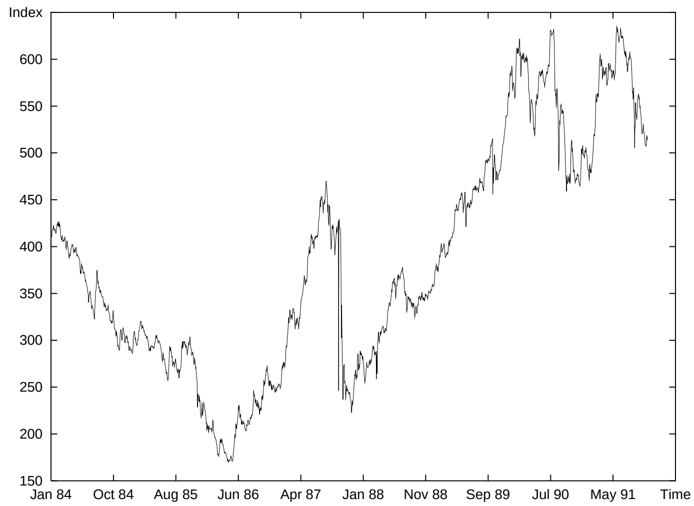

Fig. 3. Daily stock price of KLCI.

The inputs to the neural network model are I_{t-1}, I_t, MA_5, MA_{10}, MA_{50}, RSI, M, %K and %D. The output is I_{t+1}. Here I_t is the index of tth period, MA_j is the moving average after jth period, and I_{t-1} is the delayed time series. For daily data, the indicators are calculated as mentioned above. Other indicators are defined as follows,

$$\mathrm{M} = \mathrm{CCP} - \mathrm{OCP}\,, \tag{4.1}$$

where
CCP = current closing price,
OCP = old closing price for a predetermined period (5 days),

$$\mathrm{RSI} = 100 - \frac{100}{1 + \frac{\Sigma(\text{positive changes})}{\Sigma(\text{negative changes})}} \tag{4.2}$$

$$\%\mathrm{K} = \frac{\mathrm{CCP} - \mathrm{L9}}{\mathrm{H9} - \mathrm{L9}} 100\,, \tag{4.3}$$

where
L9 = the lowest low of the past 9 days,
H9 = the highest high of the past 9 days,

$$\%\mathrm{D} = \frac{\mathrm{H3}}{\mathrm{L3}} 100\,, \tag{4.4}$$

where
H3 = the three day sum of (CCP − L9)
L3 = the three day sum of (H9 − L9)

Indicators can help traders identify trends and turning points. Moving average is a popular and simple indicator for trends. Stochastic and RSI are some simple indicators which help traders identify turning points.

In general, the stock price data have bias due to differences in name and spans. Normalization can be used to reduce the range of the data set to values appropriate for inputs to the activation function being used. The normalization and scaling formula is

$$y = \frac{2x - (\max + \min)}{\max - \min}\,, \tag{4.5}$$

where
x is the data before normalizing,
y is the data after normalizing.

Because the index prices and moving averages are in the same scale, so the same maximum and minimum data are used to normalize them. The max is derived from the maximum value of the linked time series, and the same applies to the minimum. The maximum and minimum values are from the training and validation data sets. The outputs of the neural network will be rescaled back to the original value according to the same formula.

4.2. *Nonlinear analysis of the KLCI data*

Statistics characteristics of KLCI series are analysed first before applying it to neural network models. Table 3 shown mean, maximum, minimum, variance, standard deviation, average deviation, skewness and kurtosis of KLCI for the period from January 1984 to October 1991.

Figure 4 shows the graph of the KLCI represented as logarithmic return $\ln(\mathrm{I}_{t+1}/\mathrm{I}_t)$ for the defined period, where I_t is the index value at time t. It shows that the data is very noisy which makes forecasting very difficult.

The high standard deviation of returns indicates that the risk in this emerging market is higher than in developed merchant markets.

The rescaled range analysis(R/S analysis) [15, 21] is able to distinguish a random series from a non-random series, irrespective of the distribution of the underlying series. In this paper, it is used to detect the long-memory effect in the KLCI time series over a time period. R captures the maximum and minimum cumulative deviations of the observations x_t of the time series from its mean (μ), and it is a

Table 3. Statistics results of KLCI.

Min	Mean	Max	Stdev	Var	Avedev	Skew	Kurt
169.83	383.89	821.77	123.06	15,143.49	104.58	0.36	−0.89

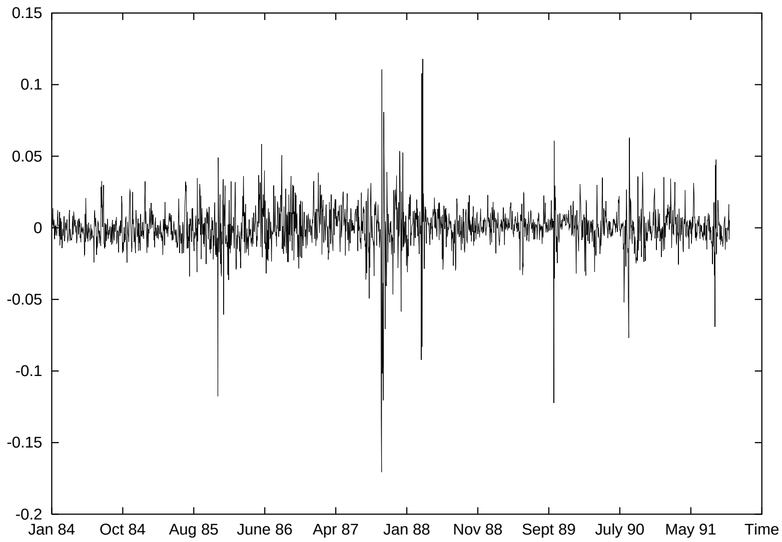

Fig. 4. Logarithmic returns of KLCI daily data.

function of time (the number of observations is N):

$$R_N = \max_{1 \le t \le N} [x_{t,N}] - \min_{1 \le t \le N} [x_{t,N}] \,, \tag{4.6}$$

where $x_{t,N}$ is the cumulative deviation over N periods and defined as follows:

$$x_{t,N} = \sum_{u=1}^{t} (x_u - \mu_N) \,, \tag{4.7}$$

where μ_N is the average of x_u over N periods.

The R/S ratio of R and the standard deviation S of the original time series can be estimated by the following empirical law: $R/S = N^H$ when observed for various N values. For a value of N, the Hurst exponent can be calculated by

$$H = \log(R/S)/\log(N) \,, \qquad 0 < H < 1 \,, \tag{4.8}$$

and the estimate of H can be found by calculating the slope of the log / log graph of R/S against N using regression.

The Hurst exponent H describes the probability that two consecutive events are likely to occur. The type of series described by $H = 0.5$ is random, consisting of uncorrelated events. A value of H different from 0.50 denotes the observations that are not independent. When $0 \le H < 0.5$, the system is an antipersistent or ergodic series with frequent reversals and high volatility. Despite the prevalence of the mean reversal concept in economic and financial literature, only few antipersistent series have been found. For the third case ($0.5 < H \le 1.0$), H describes a persistent or trend-reinforcing series which is characterized by long memory effects. The strength of the bias depends on how far H is above 0.50. The lower the value of H, the more noise there is in the system and the more random-like the series is.

The value of Hurst exponent for the KLCI time series was found to be 0.88 which indicates a long-memory effect in the time series. Hence, there exist possibilities for conducting time series forecasting in the KLCI data. As suggested by Edgar E. Petter [21], a further confirmation of the validity of this claim can be obtained by randomly scrambling the original series. Two of such scrambled series result Hurst exponent to be 0.61 and 0.57, respectively. This drop of H shows that scrambling destroyed the long memory structure. In other words, a long memory component does exist in the KLCI time series.

4.3. *Neural network model building*

Historical data are divided into three parts; training, validation and testing sets. The training set contains two-thirds of the collected data, while the validation and the testing sets contain two-fifteenths and three-fifteenths, respectively. A model is considered good if the error for out-of-sample testing is the lowest compared with the other models. If the trained model is the best one for validation and also the best one for testing, one can assume that it is a good model for future forecasting.

A practical approach of model selection used in this study can be described as follows. The neural network is trained by training data set to find the general pattern of inputs and outputs. To avoid overfitting, the hold-out validation set is used as cross-validation and then a "best" model is picked up. This model is then used as a forecasting model applied to out-of-sample testing data set. The data are chosen and segregated in time order. In other words, the data of the earlier period are used for training, the data of the later period are used for validation, and the data of the latest time period are used for testing. This method may have some *recency problems*. Using the above rule, the neural network is only trained using data up till the end of 1988. In forecasting the index after November 1991, the neural network is "forced" to use knowledge up till 1988 only. Hence, another method where the data are randomly chosen is designed to circumvent this problem.

After experimenting with the choice of data, a very good testing result may not predict well. On the other hand, a model which is trained with randomly chosen data may predict well even with average testing results.

In theory, a neural network model that fits any kind of functions and data could be built. They have been shown to be universal approximators of functions and their derivatives mathematically [14, 30]. The main consideration when building a suitable neural network for the financial application is to make a trade-off between convergence and generalization. It is important not to have too many nodes in the hidden layer because this may allow the neural network to learn by example only and not to generalize [2].

According to Beale and Jackson [3], a network with one hidden layer can model any continuous function. Depending on how good we want to approximate our function, we may need tens, hundreds, thousands, or even more neurons. In practice, people would not use one hidden layer with a thousand neurons, but prefer more hidden layers with fewer neurons doing the same job.

Freisleben [13] achieved the best result with the number of hidden nodes being equal to a multiple number k of the number of inputs n minus one, as denoted in Eq. (4.9).

$$\text{No\_of\_hidden\_nodes} = (k * n) - 1\,. \tag{4.9}$$

But Refenes [24] achieved the best configuration in terms of the trade-offs between convergence and generalization and obtained a conveniently stable network which is a 3-32-16-1 configuration.

There are two more formulas which appeared in the discussion of neural network news group [6].

$$\text{No\_of\_hidden\_nodes} = \sqrt{\text{input} * \text{output}} \tag{4.10}$$

$$\text{No\_of\_hidden\_nodes} = \ln\left(\text{No\_of\_nodes\_in\_previous\_layer}\right) \tag{4.11}$$

Of course there is probably no perfect rule of thumb and the best thing to do is to use a cross validation set to obtain the optimal generalization point. An iterative process is adopted beginning with one node in a hidden layer and working up until a minimum error in the test set is obtained. We adopt a simple procedure of deciding the number of hidden nodes which is also determined by the number of nodes in the input or preceding layer. For a single hidden layer neural network, the number of nodes in the hidden layer being experimented are in the order of $n2, n2{\pm}1, n2{\pm}2, \ldots$, where $n2$ stands for half of the input number. The minimum number is 1 and the maximum number is the number of input, n, plus 1. In the case where a single hidden layer is not satisfactory, an additional hidden layer is added. Then another round of similar experiments for each of the single layer networks are conducted and now the new $n2$ stands for half of the number of nodes in the preceding layer. For five inputs and one output neural networks, the architectures of the neural network to be experimented are, 5-3-1, 5-2-1, 5-4-1, 5-1-1, 5-5-1, 5-6-1, 5-3-2-1, 5-3-1-1, 5-3-3-1, 5-3-4-1, ..., for instance. For each neural network architecture, different learning and momentum rates are also experimented until a satisfactorily low NMSE is reached for the validation data set. The resultant network is then considered the best candidate of forecasting model for this architecture to test the forecastability using out-of-sample testing data.

Primary sensitive analysis is conducted for input variables. Models are built in an attempt to discover which of these variables influence the output variable. As a rule of thumb to determine whether a variable is relevant, the network was run for numerous times, each time omitting one variable. If the results before and after omitting a variable are the same or even better it can be inferred that this

Table 4. The best results for six different models. (Arch: architecture of the neural network; α: learning rate; η: momentum rate; NMSE: normalized mean squared error; Grad: correctness of gradients; and Sign: correctness of signs.)

Architecture	α	η	NMSE	Grad(%)	Sign(%)
5-3-1	0.005	0.0	0.231175	67	77
5-4-1	0.005	0.0	0.178895	85	86
5-3-2-1	0.005	0.1	0.032277	78	83
6-3-1	0.005	0.1	0.131578	82	66
6-5-1	0.005	0.0	0.206726	78	89
6-4-3-1	0.005	0.1	0.047866	75	96

variable probably does not contribute much to producing the outcome. Such variables include M_{20}, M_{50}, %K, %D among others. On the contrary, if the results of the network deteriorate significantly after the variable has been left out, then the variable has probably big influences on the outcome. Such variables include I_t, RSI, M and MA_5. The findings with respect to %K, %D was somewhat surprising as they are known to be important technical factors among traders. To investigate this further, more sensitive analysis [23] should be conducted. Five important factors, namely, I_t, MA_5, MA_{10}, RSI and M are chosen as five inputs. In addition, I_{t-1} is chosen also for some models. The models constructed have configurations such as 5-3-1, 5-4-1, 5-6-1, 5-3-2-1, 6-3-1, 6-5-1, 6-5-1, and 6-4-3-1.

The NMSE is used as the basic performance metric for validation data set. As mentioned earlier, NMSE should not be used as the unique criterion. We select not only one best performance neural network but many similar level of networks as our candidates of forecasting model. The NMSE level for the validation set of these models ranges from 0.005433 to 0.028826. The configurations and performance in out-of-sample testing data for different models using daily data are shown in the Table 4. Figure 5 is the result of the prediction using out-of-sample testing data.

4.4. *Paper profits using neural network predictions*

As mentioned earlier, the evaluation of the model depends on the strategy of the traders or investors. To simulate these strategies, a small program was developed. Assume that a certain amount of seed money is used in this program. The seed money is used to buy a certain number of indexed stocks in the right proportion when the prediction shows a rise in the stock price. To calculate the profit, the major blue chips in the KLCI basket are bought or sold at the same time. To simplify the calculation, assume that the aggregate price of the major blue chips is the same as the KLCI. A method used is to go long when the neural network model predicts that the stock price will rise. Then the basket of stocks will be held at hand until the next turning point that the neural network predicts. The results obtained are shown in Table 5.

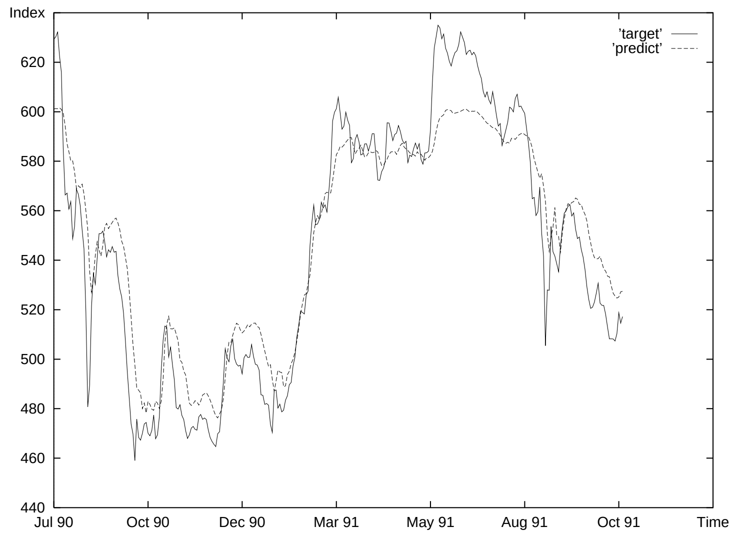

Fig. 5. Daily stock price index prediction of KLCI (Out of sample data: from July 30, 1990 (horizontal scale 0) to Oct 16, 1991 (303)).

Table 5. Paper profit for different models (Return-1: Annual return using entire data set, Return-2: annual return of Strategy 1 using out of sample *daily* data set, Return-3: annual return of Strategy 2 using out of sample *daily* data set).

Architecture	Return-1(%)	Return-2(%)	Return-3(%)
5-3-1	38.42	9.04	6.36
5-4-1	40.14	11.91	11.88
5-3-2-1	48.89	22.94	22.94
6-3-1	42.48	12.74	15.45
6-5-1	36.48	10.24	5.37
6-4-3-1	47.05	26.02	22.93

There are two kinds of trading strategies used in this study. One uses the difference between predictions, and another uses the difference between the predicted and the actual levels.
Strategy 1:

$$\text{if}(\hat{x}_{t+1} - \hat{x}_t) > 0\,, \text{ then } buy \text{ else } sell \tag{4.12}$$

Strategy 2:

$$\text{if}(\hat{x}_{t+1} - x_t) > 0\,, \text{ then } buy \text{ else } sell\,. \tag{4.13}$$

Here x_t is the actual level at time t, where $\hat{x}_t$ is the prediction of the neural networks. The best return based on the out of sample prediction is obtained with the 6-4-3-1 model using the above simulation method. The annual return rate is approximately 26%. (See Table 5.)

If the prediction generated by the entire training and testing data are used, then the annual return rate would be approximately 47% . The reason for the different results of the different strategies lies in the forecasting errors. Assume that the error of the prediction at time t is δ_t, then the following equation holds:

$$x_t = \hat{x}_t + \delta_t \,. \tag{4.14}$$

Which strategy is more accurate depends on $|\delta_{t+1} - \delta_t|$ and $|\delta_{t+1}|$. If δ_{t+1} and δ_t have different signs, then Strategy 1 is better than Strategy 2. Otherwise Strategy 2 is better than Strategy 1.

The transaction cost will be considered in real trading. In this paper, 1% of the transaction cost was included in the calculation. In some stock markets, the indices are traded in the derivative markets. However, the KLCI is not traded anywhere. Therefore the indexed stocks were bought or sold in proportional amounts in this paper. In a real situation, this might not be possible as some indexed stocks may not be traded at all on some days. Besides, the transaction cost of a big fund trading, which will affect the market prices was not taken into consideration in the calculation of the “paper profit”. To be more realistic, a certain amount of the transaction cost will have to be included in the calculation.

4.5. *Benchmark return comparison*

There are three benchmarks for the simulated profit. Benchmark 1 uses a passive investment method: to buy the index on the first day of testing period(July 30, 1990) and sell it on the last day of this period (Oct 16, 1991). The annual return for Benchmark 1 is −14.98%, and it is calculated as follows:

$$\text{Return} = \left(\frac{\text{index2}}{\text{index1}}\right)^{\frac{12}{n}} - 1 \,, \tag{4.15}$$

where
index1 = index on first the testing day,
index2 = index on last the testing day,
n = No. of months in testing period.

Benchmark 2 is to save the seed money at the beginning and withdraw it at the end earning interest. (The monthly interest rates ranged between 6.0% and 7.6%.) The annual return for Benchmark 2 is 7.98%, and is. calculated as follows:

$$\text{Return} = \left[\prod_{j=\text{Jul 90}}^{\text{Oct 91}} \left(1 + \frac{\text{int}_j}{12}\right)\right]^{\frac{12}{n}} - 1 \,, \tag{4.16}$$

where
int_j = interest rate in jth month of testing period.

Benchmark 3 is the trend following method which is described as follows:

$$\begin{aligned} &\text{if} \quad (x_{t-1} > x_{t-2}) \cap (x_{t-2} > x_{t-3}) \\ &\quad \text{then} \ \ \textit{buy} \ \ x_t \\ &\quad \text{elseif} \ \ (x_{t-1} < x_{t-2}) \cap (x_{t-2} < x_{t-3}) \\ &\qquad \text{then} \ \ \textit{sell} \ x_t \\ &\qquad \text{else} \ \ \textit{hold} \ x_t \,, \end{aligned} \tag{4.17}$$

where the x_t is the index of day t.

Table 6 gives the comparison of returns using different benchmarks. It shows that using neural networks can achieve a significant profit compared with the benchmarks.

4.6. *Comparison with ARIMA*

Autoregressive Integrated Moving Average (ARIMA) Model was introduced by George Box and Gwilym Jenkins [4] in 1976. The Box–Jenkins methodology provided a systematic procedure for the analysis of time series that was sufficiently general to handle virtually all empirically observed time series data patterns. ARIMA(p,d,q) is the general form of ARIMA models. Here p stands for the order of the autoregressive process, d presents the degree of differencing involved, and q is the order of the moving average process. To compare the forecasting results of the neural networks, a number of ARIMA models were built. Table 6 is the results of ARIMA models together with the above benchmarks of different trading strategies.

The entire data set was used as fitting data for the ARIMA models. In other words, the data forecast by ARIMA were already used in the fitting stage of ARIMA model building. The results obtained from the ARIMA models should be compared with Return-1 in Table 5. In other words, the ARIMA models should deliver worse out-of-sample forecasting returns than the ARIMA results indicated in Table 6. The ARIMA(1,0,1) and ARIMA(2,0,2) models are similar to the neural network models studied in this paper.

5. Discussion

A very small NMSE does not necessarily imply good generalization. The sum of the NMSE of the three parts of data (training, validation and testing) must be kept small, not just the training NMSE alone. Sometimes having small NMSEs for testing and validation is more important than having small NMSE for training.

Further, better testing results are demonstrated in the period near the end of the training sets. This is a result of the "recency" problem.

For better forecasting, different data are also used to test other models. One method is to use the entire monthly data to train the neural network, and to

Table 6. Comparison of returns using benchmarks and ARIMA.

Model	Gradient (%)	Return (%)
Strategy 1 *	75	26.02
Strategy 2 *	85	25.81
Benchmark 1		−14.98
Benchmark 2		7.98
Benchmark 3		8.12
ARIMA(0,0,1)	62.83	1.60
ARIMA(1,0,0)	71.71	19.11
ARIMA(1,0,1)	48.68	19.11
ARIMA(1,1,1)	65.79	15.13
ARIMA(2,0,0)	66.78	15.13
ARIMA(2,0,2)	62.83	15.71
ARIMA(1,1,0)	66.12	15.13
ARIMA(0,1,1)	65.79	16.72
ARIMA(0,1,0)	48.68	19.11
ARIMA(2,1,2)	65.13	14.47

(*: out-of-sample testing)

use the weekly data for testing and validation. Although the weekly data and the monthly have different noises and characteristics, the experiments show that good results could also be obtained from different data sets.

There are tradeoffs for testing and training. The behavior of an individual could not be forecast with any degree of certainty, on the other hand, the behavior of a group of individuals could be forecast with a higher degree of certainty. In other words, a large number of uncertainties produce a certainty. In this case, one should not say it is the best model unless he has tested it, but once one has tested it one has not trained enough. One of the aims of this paper is to find a "best" forecasting model for KLCI data. In order to train neural networks better, all the data available should be used. The problem is that we have no data to test the "best" model. In order to test the model, we partition the data into three parts. The first two parts are used to train (and validate) the neural network while the third part of data is used to test the model. But the networks have not been trained enough as the third part is not used in training. In considering other performance metric, we trained several networks as candidates of the "best" forecasting model. There are two approaches for using the forecasting result. One is the best-so-far approach which choose a best model using paper profit as performance metric, e.g. 5-3-2-1 or 6-4-3-1 as forecasting model. As we have not trained the networks enough, we propose a so-called committee approach which select some of the best so far model as a forecasting committee. Using the committee approach in real future forecasting, we will not base on one (the best) model's forecasting result but the committee's recommendation. A real trading action is based on the majority of committee member's forecasts.

Other measures include the correctness of signs and the correctness of gradients. The choice of the testing criteria, be it NMSE, or sign, or gradient, depends on the trading strategies. In the stock market, the gradient is very important for traders. Sometimes, they even need not know what the actual level of the index is.

In this research, the index of the stock market is predicted. In some markets, the index futures can be bought or sold. The practitioner can use the neural network forecast as a tool to trade the futures. But in KLSE the index futures are not traded. A possible way to use the forecasting results in practice is to trade the stocks which are the most active and highly correlated with the index. Choosing the most active stocks can ensure that the practitioners can buy or sell the stocks at a certain level.

There are four challenges beyond the choice of either technical or fundamental data for using neural network to forecast the stock prices. First, the inputs and outputs of the neural networks have to be determined and preprocessed. Second, the types of neural networks and the activation functions for each node have to be chosen. Third, the neural network architecture based on the experiment with different models has to be determined. Finally, different measures to evaluate the quality of trained neural networks for forecasting have to be experimented with.

6. Conclusion and Future Research

This paper reports an empirical work which investigates the usefulness of artificial neural networks in forecasting the KLCI of Malaysian stocks. The performance of several backpropagation neural networks applied to the problem of predicting the KLSE stock market index was evaluated. The delayed index levels and some technical indicators were used as the inputs of neural networks, while the current index level was used as output. With the prediction, significant paper profits were obtained for a chosen testing period of 303 trading days in 1990/91.

The same technical analysis was also applied to weekly data. However, the results were not as impressive as those obtained using the daily data. This is attributed to the high volatility of the KLSE market.

The significance of this research is as follows:

- It shows that useful prediction could be made for KLCI without the use of extensive market data or knowledge.
- It shows how a 26% annual return could be achieved by using the proposed model. The annual return for passive investment and bank savings were −14.98% and 7.98% respectively for the same period of time. The excess return of 18% could be achieved by deducting interests that could be obtained from the bank. The return of 26% also compared favorably with the ARIMA models. Thus, the results shows that there is practical implication for an index linked fund to be set up in the cash market or for the index to be traded in the derivative market.

- It highlights the following problems associated with neural network based time series forecasting:
  - (i) the hit rate is a function of the time frame chosen for the testing sets;
  - (ii) generalizability of the model over time to other period is weak;
  - (iii) there should be some recency trade offs.

To improve neural networks' capabilities in forecasting, a mixture of technical and fundamental factors as inputs over different time periods should be considered. Sensitivity analysis should be conducted which can provide pointers to the refinement of neural network models. Last but not least, the characteristics of emerging markets such as KLCI should be further researched to facilitate better modeling of the market using neural networks. The forecasting results can then be applied to the trading of index linked stocks under consideration of the transaction costs.

Acknowledgments

Parts of this article have been presented at a seminar at National University of Singapore, International Conference On Neural Networks in the Capital Markets and IEEE International Conference on Neural Networks. We would like to thank participants for their helpful comments and invaluable discussions with them. The authors are grateful to the anonymous referees whose insightful comments enabled us to make significant improvements.

References

[1] J. S. Ang and R. A. Pohlman, *A note on the price behavior of Far Eastern stock*, J. Int. Business Studies, Spring/Summer (1978).
[2] E. B. Baum and D. Hassler, *What size net gives valid generalization*? Neural Computation **1** (1989) 151–160.
[3] R. Beale and T. Jackson, *Neural Computing: An Introduction*, Adam Hilger (1990).
[4] G. E. PBox and G. M. Jenkins, *Time series analysis: forecasting and control*, Holden-day, San Francisco (1976).
[5] A. J. Chapman, *Stock market reading systems through neural networks: developing a model*, Int. J. Appl. Expert Systems **2**(2) (1994) 88–100.
[6] comp.ai.neural-nets, Internet Newsgroup (1995–1996).
[7] D. R. Coulson, *The Intelligent Investor's Guide to Profiting from Stock Market Inefficiencies*, Probus Publi. Co. (1987).
[8] S. Dutta and S. Shekhar, *Bond rating: A non-conservative application of neural networks*, IEEE Int. Conf. on Neural Networks (1990).
[9] *More in a cockroach's brain than your computers dream of*, The Economist (April 15 1995).
[10] *Mahathir, Soros and the currency market*, The Economist, (Sept. 27 1997).
[11] V. Errunza, *Emerging markets: some new concepts*, J. Portfolio Management (Spring 1994).
[12] E. F. Fama, *The behavior of stock market prices*, J. Business (Jan. 1965) 34–105.

[13] B. Freisleben, *Stock market prediction with backpropagation networks*, Industrial and Engineering Applications of Artificial Intelligence and Expert System. 5th International Conference, Paderborn, Germany (June 1992) 451–460.

[14] K. Hornik, M. Stinchcombe and H. White, *Multilayer feedforward networks are universal approximators*, Neural Networks **2**(5) (1989) 359–366.

[15] H. E. Hurst, *Long term storage of reservoirs*, Transactions of the American Society of Civil Engineers **116** (1951).

[16] J. M. Hutchinson, Andrew Lo and T. Poggio, *A nonparametric approach to pricing and hedging derivative securities via learning networks*, J. Finance **49** (July 1994) 851–889.

[17] A. Lapedes and R. Farber, *Nonlinear signal processing using neural networks*, IEEE Conference on Neural Information Processing System — Natural and Synthetic (1987).

[18] R. M. Levich and L. R. Thomas, *The merits of active currency risk management: evidence from international bond portfolios*, Financial Analysts J. (Sept-Oct 1993).

[19] E. Y. Li, "*Artificial neural networks and their business applications*", Information and Management **27** (1994) 303–313.

[20] S. Margarita, *Genetic neural networks for financial markets: some results*, ECAI'92, Vienna, Austria (1992) 211–21.

[21] E. E. Peters, *Chaos and order in the capital markets: A new view of cycles, prices, and market volatility*, John Wiley & Sons Inc. (1991).

[22] T. Plummer, *Forecasting Financial Markets: A Technical Analysis and the Dynamic of Price*, New York (1991).

[23] H.-L. Poh, J. T. Yao and T. Jašić, *Neural networks for the analysis and forecasting of advertising and promotion impact*, Int. J. Intelligent Systems in Accounting, Finance and Management **7** (1998).

[24] A. N. Refenes, A. Zapranis and G. Francis, *Stock performance modeling using neural networks: a comparative study with regression models*, Neural Network **5** (1994) 961–970.

[25] D. E. Rumelhart and J. L. McClelland, *Parallel distributed processing: Explorations in the micro-structure of cognition*, volume 1, The MIT Press (1986) 318–362.

[26] B. H. Solnik, *Note on the validity of the random walk for European stock prices*, J. Finance (Dec. 1973).

[27] T. Tanigawa and K. Kamijo, *Stock price pattern matching system: dynamic programming neural network approach*, IJCNN'92, Vol. 2, Baltimore, Maryland (June 1992).

[28] S. Taylor, *Modeling Financial Time Series*, John Wiley & Sons (1986).

[29] R. R. Trippi and E. Turban (eds), *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-world Performance*, Irwin Professional Pub. (1996).

[30] H. White, *Artificial Neural Networks: Approximation and Learning Theory*, Blackwell (1992).

[31] B. K. Wong *A bibliography of neural network business applications research: 1988-September 1994*, Expert Systems **12**(3) (1995).

[32] J. T. Yao and H.-L. Poh, *Equity forecasting: a case study on the KLSE index*, Neural Networks in Financial Engineering, Proc. 3rd Int. Conf. on Neural Networks in the Capital Markets, Oct 1995, London, eds. A.-P N. Refenes, Y. Abu-Mostafa, J. Moody and A. Weigend, World Scientific (1996) 341–353.

[33] O. Yong, *Behavior of the Malaysian Stock Market*, Penerbit Universiti Kebangsaan Malaysia (1993).

[34] O. Yong, *The Malaysian Stock Market and You*, Leeds Publi., Malaysia (1995).