Department of Social Systems and Management

Discussion Paper Series

No. 1131

Prediction of Stock Returns Using the Artificial Neural Network: Evidence from Developed and Developing Countries

by

Jennifar Alam Jun Nagayasu

September 2005

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

Prediction of Stock Returns Using the Artificial Neural Network: Evidence from Developed and Developing Countries

Jennifar Alam Jun Nagayasu*
Graduate School of Systems and Information Engineering
University of Tsukuba

September, 2005

Abstract

Using the artificial neural network (ANN) model, this study examines the predictability of stock returns in developed and developing countries, and evaluates its forecasting performance with those of a benchmark model (ARIMA). Some evidence is obtained that, in the long-term, the ANN produces more accurate forecasts while the ARIMA seems more suitable for short-term prediction. Furthermore, based on trading simulation exercises, very convincing evidence is obtained of the ANN generating accurate market turning signals.

JLE classification: C450, C530, G150
Keywords: Artificial neural network, financial forecasting, market timing signals, trading simulations

*Corresponding author: Jun Nagayasu, Mailing address: Graduate School of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki, 305-8573 Japan. The earlier version of this paper was presented at the University of Tsukuba. We would like to thank Eiji Fujii, Tatsuyoshi Miyakoshi, Isao Shoji, Yuichiro Kanazawa and Islam Mohammad Tanvir for constructive comments on the earlier version. The first author also thanks the Japan International Cooperation Agency for financial support provided for her study in Japan.

1. Introduction:

Acceptance of financial markets as being governed by purely random forces leads to the efficient market hypothesis (EMH). Originated by the Gaussian random walk model of Bachelier (1913) and later developed by Fama(1960), the EMH has been a central assumption in finance literature for a number of decades. The EMH can be expressed as a linear model and states that realized prices incorporate and reflect all relevant information. Therefore, there is no scope for making profits from analysis of historical market prices.[1]

However, recent studies (i.e., Brock *et al* 1992) provide evidence of the predictability of stock returns, indicating that it is indeed possible to make profits, and furthermore they confirm the non-linearity of stock returns. Therefore, an alternative approach known as artificial neural networks (ANN), which can capture non-linearity, attracted much interest from financial economists in the late Eighties/early Nineties.[2]

Due to the lack of a training algorithm for the multi-layer neural network which (will be explained in the next section) at that time, research using the ANN was quite limited. However, its application to forecasting started to take on a new shape after 1986 when the back-propagation algorithm was introduced (Rumelhart *et al* 1986).[3] One of the first applications of the ANN in forecasting was performed by Lapedes and Farber (1987), and since then it has been applied to research in finance. For example, Gencay (1988) and Gencay and Stegnos (1998) found strong predictability of Dow Jones Industrial Index returns using moving average trading rules as inputs to neural networks. Komoda and Mizuno *et al* (1998) developed a modular neural network for technical analysis of the Topix index using some technical indices as inputs and compared the result with a statistical model based on discriminant analysis. In their study, the ANN yields a better result for a buying decision but not in the case of a selling decision. Similarly, Qi (1999) analyzed financial and economic data to generate recursive predictions of stock returns and found that the ANN provides better in-sample and out-of-sample forecasts than a linear model. Furthermore, Jasic and Wood (2004) measured the profitability of stock index returns of four major stock markets, S&P 500, DAX, TOPIX and FTSE, using an ANN and compared it with a benchmark linear autoregressive (AR(1)) model. Their result indicated strong evidence of out-of-sample predictability in each case compared to the linear model.

[1] This is the weak form of the efficient market hypothesis.

[2] The ANN is based on the biological nervous systems and can perform extraordinary complex computations in the real world without recourse to explicit quantitative operations. This was first used in the fields of cognitive science and engineering.

[3] Werbos (1974, 1988) first formulated the back propagation and found that ANNs trained with back propagation outperform traditional statistical methods such as regression.

In the light of such literature, we model the stock return behaviors of several countries using the ANN, predict stock returns, and compare their forecasting performance with that of a benchmark time-series model (ARIMA). This work differs from previous studies in several aspects. First of all, the forecasting performance of the ANN and ARIMA is conducted both in the short- and long-time horizons. While previous research has frequently employed the AR(1) as a benchmark, a more robust model (ARIMA) is used in this paper. Secondly, when using a transfer function in network architecture, this paper uses the g*tan h transfer function*h which is more flexible than the conventional *sigmoid transfer function*h. Thirdly, in addition to the stock returns of developed countries, this paper analyzes returns in emerging markets (BSE 30 (India) and KLSE (Malaysia)) to test the conformity of the predictive performance of the ANN. Analysis of emerging markets is in sharp contrast to previous research which focused exclusively on financial data in developed economies. Finally,unlike previous papers which only relied upon the size of forecast errors for judging predictive performance, this paper uses a wide range of real life trading performance measures to evaluate the economic potential of the predictive capabilities of the ANN and the ARIMA.

2.Artificial Neural Networks (ANN)

The ANN is a multivariate, non-linear, and nonparametric inference technique that is data driven and model free (Azoff 1994). *Multivariate* refers to neural network inputs comprising many different variables whose interdependency and causative influences are exploited in predicting the future behavior of a sequence. *Non parametric* and *model free* describe, in the statistical sense, the fact that no predetermined parameters are required to specify the mapping model. The ANN is trained for the adaptation of *free parameters* to discover any possible relationship driven and shaped only by input data. The *free parameters* are weights associated with the signal communication lines between neurons. The ANN does not depend on assumptions such as normality and stationarity which are the preconditions for traditional statistical models.

The ANN used in this study is a Multi-Layer Perception network (MLP). In the MLP, neural network inputs are connected to one or more neurons or nodes in the input layer, and those nodes are connected to further layers until they reach the output layer. If the input to the network are observations, X_i, presented by a vector of input variables $(x_1, x_2, \cdots, x_k)$ then the jth neuron of the hidden layer receives the activation signal, s_j, given by:

$$s_j = g\left(\sum_{i=1}^{k} w_{ji}x_i + \theta_{0j}\right) \tag{1}$$

wheew $w_{ji}(j = 0, 1, \ldots, n$ and $i = 0, 1, \ldots, k)$ is a matrix of the weights from inputs to hidden layer units, θ_{0j} is a biased term representing a threshold value, and g is the activation function. To address the value of x within the range of -1 to +1, this paper uses the hyperbolic tangent function (tan h function):

$$g(x) = \tan h(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2}$$

The advantage to this approach is that, empirically the tan h function often gives rise to faster convergence of the training network than the logistic function (Bishop 2000). For the output layer, the final output can be defined by:

$$o_p = f\left(\sum_{j=1}^{n} \omega_{jp} s_j + v_{0p}\right) \tag{3}$$

Here, $\omega_{jp}(j = 0, 1, \ldots, n)$ is a vector of coefficients from the hidden nodes to the output node and v_{0p} is the biased term. Here the activation function $f(\cdot)$ is a linear transfer function. Combining the layers, the output of a feed-forward neural network can be written as:

$$o_p = f\left[v_{0p} + \sum_{j=1}^{n} \omega_{jp} g\left(\theta_{0j} + \sum_{i=1}^{k} w_{ji} x_i\right)\right] \tag{4}$$

The next step is to train the network with the aim of minimizing errors and fitting the model with time series data. For this purpose, the training algorithm used here is a gradient descent algorithm called the error back-propagation (Rumelhart *et al* 1986). The error back-propagation is a gradient descent method for training the weights of a multi-layer neural network. For a given problem, there is a set of training vectors X such that for every vector $x \in X$, there is an associated desired output vector $d \in D$, where D is the set of desired outputs associated with the training vectors in X. Let the instantaneous error E_p be defined as:

$$E_p = \frac{1}{2}\left(d_p - z_p\right)^T \left(d_p - z_p\right) = \frac{1}{2}\sum_{k=1}^{N}\left(d_{k,p} - z_{k,p}\right)^2 \tag{5}$$

where $d_{k,p}$ is the kth component of the pth desired output z_p when the pth training exemplar x_p is input to the multi-layer perceptron. In the back-propagation, the change of weight is proportional to the gradient of this error:

$$\Delta w_{ji}^{t} = -\alpha \frac{\partial E_p}{\partial w} + \eta \Delta w_{ji}^{t-1} \tag{6}$$

where the learning rate (α) is some small positive number between 0 and 1, the momentum factor (η) is also a small positive number (between 0 and 1), and Δw_{ji}^{t-1} is a change in the weight computed at time t.

To model the stock returns using time series data, the input units are lagged variables of length, d, $(x_{t-1}, x_{t-2}, \cdots, x_{t-d})$ where x_t is the observations of the time series at time t and a time-delay term,τ, which is an interval between the time of input observations and prediction output. The rationale of using the lag value and time-delay is that they can explain the full geometric structure of the non- linear system (Jasic *et al* 2004).

3. Data

Our data set comprises the daily closing prices of the S&P 500 (New York, USA), Nikkei225 (Tokyo, Japan), FTSE 100 (London, UK), BSE 30 (Mumbai, India), and KLSE Composite index (Kuala Lumpur, Malaysia) obtained from Yahoo-finance (www.yahoo.com/finance). For each series, 1,200 sample observations are taken up to June 30, 2004. Since the number of working days in a year differs among the countries, the closing date of the sample was fixed at June 30, 2004 and the previous 1200 working days' stock market data have been taken as the sample data. Therefore, for the same sample size, the sample period varies slightly for different countries (See Table 1).

Among 1,200 observations, the first 1,000 are used for model estimation for the in-sample analysis, and the last 200 observations are reserved for the out-of-sample forecasting. As input data, the return series are obtained by taking the first difference of log prices: $r_t = \ln(p_t) - \ln(p_{t-1})$ where p_t is a stock price at time t.[4] Figures 1 and 2 represent the graphical presentation of the index prices and the log return series.

<Figures 1 and 2>

Table 1 provides the descriptive statistics for each market. The return series have a negative mean (except the KLSE), indicating declining stock prices during the sample periods. All series have negative skewness, which implies that the distribution of stock returns is asymmetric and that the left tail of the distribution is fatter than the right tail. Most returns have a kurtosis of less than 3 (as required by the normal distribution) which indicates that the data are platykurtic i.e. the distributions of stock returns that are simultaneously less peaked and have thinner tails. This indicates that, return series are not concentrated on a single sharp peak

[4]Logarithmic transformation is useful for data which can take both small and large values and is characterized by an extended right hand tail distribution. Logarithmic transformations also convert multiplicative or ratio relationships to additive which is believed to simplify and improve neural network training.

value. For the BSE 30 and KLSE, kurtosis is greater than 3 indicating a leptokurtic distribution and suggesting that return series are concentrated on a peak value and have fat tails.

<Table 1>

More formal statistical tests are carried out in order to analyze time-series properties of return data. The Kolmogorov-Smirnov test statistics have been calculated in order to check whether or not the stock returns are normally distributed. We find that all series rejected the hypothesis of normality at the 1% significance level (Table 2). Similarly, the Ljung-Box Q statistics have been calculated to test the non-linearity of the stock returns series (see Tsay 2002). This is a portmanteau statistic for detecting departures of stock returns from zero-autocorrelations. The autocorrelation pattern is evident from the significant Ljung-Box Q statistics for the returns in all markets (Table 3). Similarly, the Ljung-Box Q statistics for squared returns are significantly large and thus confirm the presence of heteroschedasticity in the return series.

<Tables 2 & 3>

Therefore, the return series present the properties of non-linearity, non-normality and heteroschedasticity, which make data modeling and forecasting a formidable task, and thus suggest the necessity of considering non-linear models for describing the observed characteristics in such return series and for making sensible out-of-sample forecasts.

Thus, this paper considers non-linear models that can handle these properties of stock return data, and investigates which approach, the ANN or the ARIMA, is compatible with high frequency stock return data, and gives reliable prediction results.

4. FORECASTING PERFORMANCE

4.1 Model specifications

There are several important elements in modeling an ANN. The most critical step in building such a model is deciding the number of hidden layers. It is the number of hidden layers and the number of nodes that provide the network with its ability to capture the dynamics of the time-series data. In practice, a neural network with one or two hidden layers and a sufficient number of hidden neurons is capable of approximating any continuous function (Azoff 1994).[5] Zhang (1994) reports that

[5]Increasing the number of hidden layers increases the computation time and the danger of over-fitting which leads to a superior in-sample fit but a poor out-of-sample forecasting performance (Kaastra and Boyd 1996).

networks with two hidden layers can model the underlying data structure and make accurate predictions.[6] Thus, we consider, first, the ANN with a maximum of two hidden layers, and then using the Root Mean Squared Error (RMSE) as a criterion for finding the optimal network, the ANN with two hidden layers is found to be resulted in smaller training errors. [7]

The other critical point in building a network is to determine the number of neurons in the input and hidden layer. In most cases, the number of input nodes corresponds to the dimension of the input vector used to forecast the future values. In a time-series forecasting problem, the number of input nodes corresponds to the number of lagged observations used to discover the underlying pattern in a time series. Considering these, this study considers two to five input nodes to incorporate the weekly lag length (5days) of stock market movements in building the neural network simulation model. After trial and error during the training phase, in the case of the Nikkei225 and S&P500, 4 input nodes are found to be optimal, while for the FTSE100, BSE30 & KLSE, the lowest error term resulted from using 3 input nodes.

To train the network, initial weights are selected randomly, and moderate learning rates, α of 0.5 and momentum value and η of 0.3, are used. Kaastra and Boyd (1996) commented that the greater the number of weights relative to the size of its training sets, the greater the ability of the network to memorize the idiosyncrasies of individual observations. As a result, the generalization of the training set is lost and the model becomes of little use for forecasting. Therefore, this paper follows a fundamental statistical modeling rule that specifies that for N sets of observations, the degree of freedom in the model should not exceed $N^{1/2}$. Given the size of 1,000 observations for the training set, the upper bound of the weight has been taken as a maximum of 31. Keeping this upper bound limit and considering the lag and time delay effect, the following network architectures are initially considered: 4-(4+3)-1 (i.e. 4 inputs, 4 and 3 units in two hidden layers respectively and 1 output and weight: $4{\times}4{+}4{\times}3{+}3{\times}1 = 31), 4-(3+2)-1, 3-(4+3)-1, 3-(4+2)-1, 2-(3+2)-1$, and $2 - (4 + 3) - 1$. Our criterion for the model performance is to minimize the RMSE between the output and the teaching value. In the training phase, the number of iterations was varied from 500 to 1,000. Finally, through trial and error, the ANN specifications summarized in Table 4 are employed in the subsequent part of our study.

<Tables 4>

[6] Cybenko (1988) and Lapeds and Farber (1988) argue more strongly that a network does not need more than one hidden layer to solve most problems including forecasting.

[7] Detailed results are available on request.

4.2 Forecasting Performance Based on the RMSE, MAE, and Sign and Gradient of Returns

Four criteria have been used to make comparisons of the forecasting ability of the ANN model and the ARIMA model. First of all, the forecasting performance has been evaluated by the two traditional methods of error calculation, the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Tables 5 and 6 present the comparative out-of-sample forecasting performance of these models. From these tables, the following conclusions can be obtained. In terms of error-based criteria (the RMSE and MAE), the ANN model outperforms the ARIMA marginally in long-run prediction, but in the short-term, the ARIMA model yields smaller RMSE and MAE values. One exception is the KLSE where the ANN performs better in the short-term.

<Tables 5 and 6>

However, given the uncertainty associated with the exact values of future stock returns, a correct signal for the direction of the stock movements could be more useful information for making an investment decision. This is so since for traders and market analysts, the market turning point and direction is the most important element of the forecast. Therefore, this study also uses two directional statistics defined by Jasic et al (2004). These statistics are the Sign of Returns which measures the percentage of accuracy in predicting the direction of changes in the price level, and the Gradient of Returns which calculates the percentage of accuracy in predicting the direction of changes in the returns or gradient of returns.

When directional criteria are employed, more convincing evidence is obtained in favor of the ANN (Tables 5 and 6). Indeed, both in short- and long-run forecasting, the ANN gives significantly better values of sign and direction statistics. This implies that the ANN produces more reliable market timing signals than the ARIMA, and underscores the importance of modeling non-linearity in the stock return function.

Now let us see whether these prediction results reflect any economic values in the context of stock trading simulation exercises.

4.3 Forecasting Performance Based on Stock Market Trading Simulations

In addition to the forecasting performance criteria used in the previous section, the economic significance of forecasting performance is also examined here by using a simple trading strategy based on the forecasting results of the ANN and ARIMA. With the objective of evaluating trading simulation results, some standard performance measures used in the fund management industry are calculated with the prediction results of both models in the short- and long-terms. More specifically, the criteria used in this section are 1) the number of winning trades, 2) the number of losing trades, 3) cumulative returns, 4) average daily returns, 5) annualized returns, 6) annualized volatility, 7) the Sharp ratio, and 8) the ideal profit ratio. The definitions of these trading measures are provided in Appendix I.[8]

The comparative results of the trading simulations of the ANN and ARIMA for market returns have been presented in Tables 7 and 8. In these tables, the results are presented under two headings: long-term simulations (200 days) and short-term ones (20 days), and are shown in nominal values of local currencies and as a percentage of the average index price of the defined period.

<Tables 7 & 8>

A comparison of the forecasting performance between the ANN and ARIMA models can be made on the basis of nominal values expressed in local currencies in these tables. Our results suggest that the relative performance of the ANN is quite impressive in most markets for all time horizons. The ANN produces the greater number of winning trades, and higher cumulative, average daily and annualized profits. The superior performance of ANN trading simulations is also evident from the ideal profit ratio which measures the ratio of the sum of correctly predicted returns against the sum of all returns of that period. On the basis of the volatility measure and the Sharp ratio also, the ANN yields better results. Furthermore, convincing evidence is obtained of the solid performance of the ANN particularly in long-run exercises, which is consistent with our results based on error calculation measures. In short, out of 10 cases examined, there is only one case where the ARIMA has outperformed the ANN. This is the case of FTSE100 where short-term profits from the ANN are less profitable compared with those from the ARIMA.

In the case of individual markets, the highest cumulative, individual and annualized returns are obtained for the Nikkei 225 followed by the BSE 30, FTSE 100, S&P 500, and KLSE in our long-term simulations. This is due to the fact that the highest success rate for the sign statistic is obtained for the Nikkei 225 (82.5%). In terms of profitability ratios (the Sharp and ideal profit ratios) also, the ANN performs best for the Nikkei 225 index, but the S&P 500 and FTSE have a better result than the BSE 30 because of their relatively low volatility. Based on our findings from the BSE 30 and KLSE, the ANN is found to perform well in modeling the stock return series in developing economies.

[8] The only limitation of these measures is that transaction costs have not been accounted for in the models. However, since electronic trading downsizes considerably transaction costs, this study considers that transaction costs will not have a significant effect on evaluation measures.

5. CONCLUSION

The financial industry has been a prime area for application of the ANN, with the latest innovations rapidly assimilated to drive the competitive edge. Therefore, it is important to examine empirically whether techniques based on the ANN outperform the traditional forecasting methods. The motivation of this paper was to ascertain whether the ANN adds any extra value in providing information that is useful for market prediction by benchmarking their results against those achieved with a simpler and more conventional modeling technique (ARIMA).

The ANN has powerful pattern recognition capabilities and predictive abilities with a high degree of accuracy. In terms of error based criteria (RMSE and MAE), both ANN and ARIMA models generate very similar results in their forecasting ability, but the encompassing profit signals and the trading strategy measures have proved that the ANN based trading simulation model results in more significant economic values than the ARIMA. The reason may be because non-linear or chaotic elements, which cannot be captured completely by the linear ARIMA model, may be found in significant number in stock returns. Furthermore, our findings suggest that non-linearity is more significant in the long-term, and thus the ANN has an edge over the long-term forecasting. Overall, our results indicate that the predictive performance of the market turning point is considerably more reliable in the ANN based simulation model. The results from this study are in line with other contemporary studies showing the significant predictability of stock market data using non-linear modeling.

References

Akaike, H., 1974, *"A New Look at the Statistical Model Identification",* IEEE Trans. Automat. Control, 19, 716-72.

Azoff, E M., *Neural Network Time Series Forecasting of Financial Markets,* John Wiley & Sons,1994.

Bishop, M Christopher, *Neural Network for Pattern Recognition,* Oxford University Press,2000.

Brock, W. Lakonishok and J. LeBaron, B.,1992, *Simple technical Trading rule and stochastic properties of stock returns,* Journal of Finance, 47,1731-64.

Dunis, C. L. and Jalilov J., *Neural Network Regression and Alternative Forecasting* Techniques for Predicting Financial Variables, 2001.

Fama, E.F.,1976, *Efficient capital markets: a review of theory and empirical work,* Journal of Finance, 25, 383-417.

Gately, E., *Neural Network for Financial Forecasting,* John Wiley & Sons, 1996.

Gencay, R.,1998, *The predictability of security returns with simple technical trading rules,* Journal of Empirical Finance, 5, 347-359.

Gencay, R. and Stegnos, T.,1998, *"Moving average rules, volume and predictability of security returns with feedforward networks,* Journal of Forecasting, 17, 401-14

George E. P. Box, Gwilym M. Jenkins and Gregory C Renisel, *gTime Series Analysis: Forecasting* & *Controlh third edition,* Prentice Hall, Englewood Cliffs.

Jasic, T. and Wood, D.,2004, *The profitability of daily stockmarket indices based on neural network predictions,* Applied Financial Economics, 14, 285-297.

Kaastra, I., and Boyd, M.,1996, *Designing Neural Network for forecasting financial and economic time series,* Neurocomputing,10 ,215-236.

Hirotaka Mizuno, Mitsuo Kosaka, Hiroshi Yajima,and Norihisa Komoda,1998, *Application of Neural Network to Technical Analysis of Stock Market Prediction,* Studies in Informatic and Control, Vol.7, No.2,111-120

Lapedes, A. and Farber Robert, *Nonlinear Signal Processing using neural networks,* Prediction and system modeling, 1987.

Michael Pokorny,1987, *An Introduction to Econometrics,* Basil Blackwell, New York.

Qi, M.,1999, *Nonlinear Predictability of stock returns using financial and economic variables,* Journal of Business and Economic Statistics, 17, 419-29.

Rosaria Silipo, International Computer Science Institute, Berkely, USA Rumelhart, D.E., J.L. McCleland, and the PDP Research Group (Eds), *Learning internal representations by error propagation.*In Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations, pp. 318-362, Cambridge, MA : MIT Press, 1986.

Schwarz, G.,1978,*"Estimating the Dimension of a Model",* Ann. Stat., 6, 461-464.

Masters, T, *Practical Neural Network Recipes in* C++, Academic Press, New York, 1993.

Tsay, R S, *Analysis of Financial Time Series,* John Wiley & Sons, 2002.

Wong, B. K. and Selvi, Y.,1998, *Neural Network Applications in Finance: A review and analysis of literature,* Information and Management, 34,129-139.

Werbos P.J.,1974, *Beyond Regression: new tools for prediction and analysis in the behavioural sciences,* Ph.D. thesis, Harvard University, Boston, MA.

Zhang, G. Patuwo, B.E. Hu, M. Y.,1998, *Forecasting with artificial neural networks: The state of the art,* International Journal of Forecasting 14, 35-62.

Appendix 1. Trading Simulation Performance Measures

Performance Measure	Description	Performance Measure	Description
Winning trades (WT)	WT=Number of $R_t > 0$	*Losing trade (LT)*	LT=Number of $R_t < 0$
Cumulative Return	$R^C = \sum_{t=1}^{N} R_t$	*Average Daily Return*	$\bar{R}_t = \frac{1}{N}\sum_{t=1}^{N} R^C$
Annualized Return	$R^A = 252 \times \bar{R}_t$	*Annualized Volatility*	$\sigma^A = 252 \times \sqrt{\frac{\sum_{t=1}^{N}(R_t - \bar{R}_t)^2}{N-1}}$
Ideal Profit Ratio	$IP = \sum_{t=1}^{N} R_t S_{stat} / \sum_{t=1}^{N} R_t$	*Sharp Ratio*	$Sharp \quad Ratio = \frac{R^A}{\sigma^A}$

Note: R_t is the stock returns.

Table 1: **Descriptive Statistics of Stock Return Series**

	Nikkei225	S&P500	FTSE100	BSE30	KLSE
Sample Period	18/11/99- 30/06/04	20/08/99- 30/06/04	28/09/99- 30/06/04	27/08/99- 30/06/04	09/08/99- 30/06/04
Mean	-0.2E-3	-0.1E-3	-0.3E-3	-0.1E-4	0.1E-3
Median	-0.1E-3	-0.1E-3	-0.2E-3	0.1E-2	0.1E-3
Std Deviation	0.006	0.013	0.013	0.017	0.009
Skewness	-0.243	-0.121	-0.094	-0.549	-0.171
Kurtosis	1.317	1.541	1.923	4.065	4.909
Minimum	-0.031	-0.060	-0.056	-0.118	-0.028
Maximum	0.025	0.056	0.059	0.079	0.025
No. of obs	1200	1200	1200	1200	1200

Note: Data source is Yahoo Finance.

Table 2: **Kolmogorov-Smirnov Test**

Index	Test Statistic	P-value
Nikkei 225	0.037	0.003
S&P 500	0.039	0.001
FTSE 100	0.041	0.000
BSE 30	0.059	0.000
KLSE	0.728	0.000

Note: The Kolmogorov-Smirnov test examines the null hypothesis of normality.

Table 3: **The Ljung-Box Statistics for the Return and Squared Returns**

	Nikkei225	S&P500	FTSE100	BSE30	KLSE
For Returns					
Q(4)	1.463*	3.644*	27.696*	13.330*	51.409*
Q(8)	2.512*	8.428*	34.444*	22.567*	57.751*
Q(12)	6.692*	17.982*	43.634*	27.032*	61.528*
Q(16)	8.838*	22.914*	53.549*	31.275*	79.608*
Q(20)	10.694*	29.420*	65.738*	34.948*	80.462*
For Squared Returns					
Q(4)	42.509*	144.659*	340.113*	245.469*	146.688*
Q(8)	75.082*	242.847*	682.584*	280.767*	148.540*
Q(12)	81.836*	310.707*	885.585*	293.040*	149.906*
Q(16)	89.457*	348.099*	1025.121*	308.715*	159.673*
Q(20)	102.138*	394.333*	1118.077*	319.438*	167.360*

Note: * denotes statistical significance at 5% level.

Table 4: **Selected ANN for Stock Return Series**

Index Name	Number of Neuron (input-hidden-output)	Time-delay	Teaching Error (root mean squared error)
Nikkei225	3-(4+3)-1	$\tau = 3$	0.005
S&P500	3-(4+3)-1	$\tau = 4$	0.010
FTSE100	4-(4+3)-1	$\tau = 4$	0.010
BSE30	4-(4+3)-1	$\tau = 4$	0.011
KLSE	4-(4+3)-1	$\tau = 4$	0.004

Note: The number of neurons is that of nodes in the input, output, and hidden layers. The time delay τ indicates an interval between time of input observations and prediction output. The teaching error is the lowest root mean square error generated in the trial and error training phase of neural network, on the basis of which optimum ANN model is selected.

Table 5: **Long-Term Out-of-Sample Results for the ANN Vs. ARIMA Models**

Model	Architecture	RMSE	MAE	Sign Statistics	Direction Statistics
Nikkei 225					
ANN	3-4+3-1,τ =3	0.005858	0.004353	82.5%	87.0%
ARIMA(p, d, q)	(2, 1, 3)	0.005955	0.004511	52.0%	72.0%
S&P500					
ANN	3-4+3-1, τ =4	0.007249	0.005621	72.5%	79.0%
ARIMA(p, d, q)	(4, 1, 2)	0.007512	0.005860	54.5%	78.0%
FTSE100					
ANN	4-4+3-1, $\tau = 4$	0.006972	0.005204	72.5%	84.5%
ARIMA(p, d, q)	(1, 1, 3)	0.007018	0.005388	50.0%	79.5%
BSE30					
ANN	4-4+3-1, τ =4	0.018550	0.013179	71.0%	73.5%
ARIMA(p, d, q)	(2, 1, 1)	0.018953	0.013613	57.5%	71.0%
KLSE					
ANN	4-4+3-1,τ =4	0.003663	0.002860	65.5%	75.0%
ARIMA(p, d, q)	(4, 1, 1)	0.003595	0.002847	52.0%	70.5%

Note: The Root Mean Squared Error (RMSE) is calculated as
RMSE= $\sqrt{(1/N)\sum_{t=1}^{N}(P_t - A_t)^2}$ where, P_t is the predicted value for time t, A_t is the actual value at time t and N is the number of observations. The Mean Absolute Error (MAE) is obtained as MAE=$(1/N)\sum_{t=1}^{N}|P_t - A_t|$. Given a set m comprising N pairs of the actual values A_t and the predicted values P_t at time t, the sign statistic is:
$S_{stat} = (1/N)\sum_{k\in m} a_k$ where $a_k = 1$ if $A_t P_t > 0$ or $A_t = P_t = 0$ and $a_k = 0$ otherwise. Finally, the direction statistic can be defined as: $D_{stat} = (1/N)\sum_{k\in m} b_k$ where $b_k = 1$ if $(A_t - A_{t-1})(P_t - P_{t-1}) >= 0$ or $b_k = 0$ otherwise. The lag order for the ARIMA model is determined by the Akaike Information criterion.

Table 6: **Short-Term Out-of-Sample Results for the ANN Vs. ARIMA Models**

Model	Architecture	RMSE	MAE	Sign Statistics	Direction Statistics
Nikkei 225					
ANN	3-4+3-1,τ =3	0.005474	0.004326	70.0%	80.0%
ARIMA(p, d, q)	(2, 1, 3)	0.005417	0.004158	60.0%	80.0%
S&P500					
ANN	3-4+3-1, τ =4	0.009821	0.008238	80.0%	85.0%
ARIMA(p, d, q)	(4, 1, 2)	0.009790	0.008089	55.0%	75.0%
FTSE100					
ANN	4-4+3-1, τ = 4	0.008557	0.006765	65.0%	80.0%
ARIMA(p, d, q)	(1, 1, 3)	0.008393	0.006473	45.0%	75.0%
BSE30					
ANN	4-4+3-1, τ =4	0.017490	0.015220	70.0%	75.0%
ARIMA(p, d, q)	(2, 1, 1)	0.016840	0.014600	70.0%	75.0%
KLSE					
ANN	4-4+3-1,τ =4	0.002129	0.001711	85.0%	95.0%
ARIMA(p, d, q)	(4, 1, 1)	0.002966	0.002555	45.0%	80.0%

Note. See Table 5.

Table 7: **Long-Term Trading Simulation Performance Measures**

	Nikkei225		S&P500		FTSE100		BSE30		KLSE	
	ANN	ARIMA	ANN	ARIMA	ANN	ARIMA	ANN	ARIMA	ANN	ARIMA
No of winning trades	160	104	145	109	145	100	142	115	131	104
No of losing trades	40	96	55	91	55	100	58	85	69	96
Cumulative returns	13400	1362	638	59	2152	239	4859	443.46	233	195.7
Average daily returns	67	7	3.2	0.3	10.8	1.19	24.30	2.22	1.16	0.98
Annualized returns	16883	1716	804	74	2712	301	6123	559	293.35	246.58
Annualized volatility	2076	2332	119	129	460	491	1501	1550	106.09	106.56
Sharp ratio	8.13	0.74	6.79	0.586	5.89	0.61	4.08	0.36	2.77	2.31
Ideal profit ratio(%)	60.28	6.14	49.89	4.61	45.33	5.03	34.04	3.11	22.03	18.52

Note: See the definition of each performance measures in Appendix I.

Table 8: **Short-Term Trading Simulation Performance Measures**

	Nikkei225		S&P500		FTSE100		BSE30		KLSE	
	ANN	ARIMA	ANN	ARIMA	ANN	ARIMA	ANN	ARIMA	ANN	ARIMA
No of winning trades	14	12	16	11	13	9	14	14	17	9
No of losing trades	6	8	4	9	7	11	6	6	3	11
Cumulative returns	982	475	116	30	35.5	36.3	519	347	49.96	9.04
Average daily returns	49	24	5.8	1.5	1.78	1.82	26	17	2.35	0.452
Annualized returns	12368	5988	1458	384	447.3	457.4	6542	4368	591.7	113.90
Annualized volatility	2007	2125	134	162	575	575	1113	1157	62.49	72.89
Sharp ratio	5.86	2.82	10.87	2.37	0.78	0.80	5.88	3.78	9.47	1.56
Ideal profit ratio(%)	46.5	21.35	69.17	18.2	6.48	6.63	40.7	27.17	62.33	12

Note: See the definition of each performance measures in Appendix I.

Figure 1. Stock Price Data

Nikkei225

S&P500

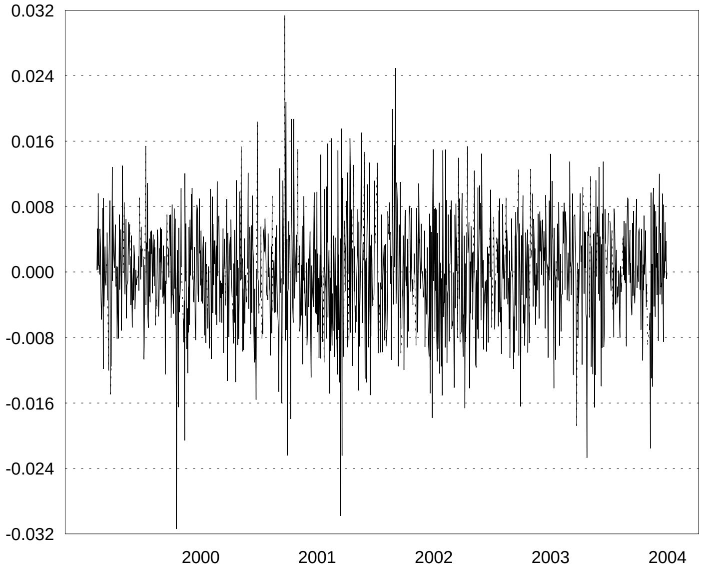
FTSE100

BSE30

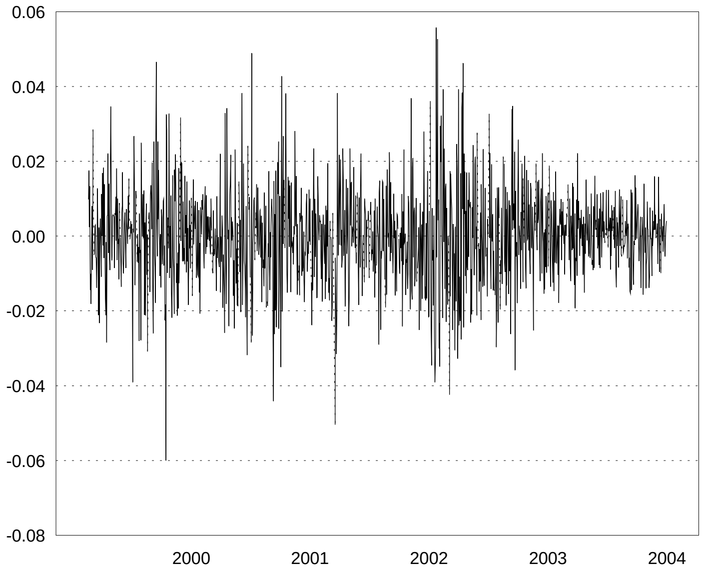
KLSE

Figure 2. Log Stock Return Data

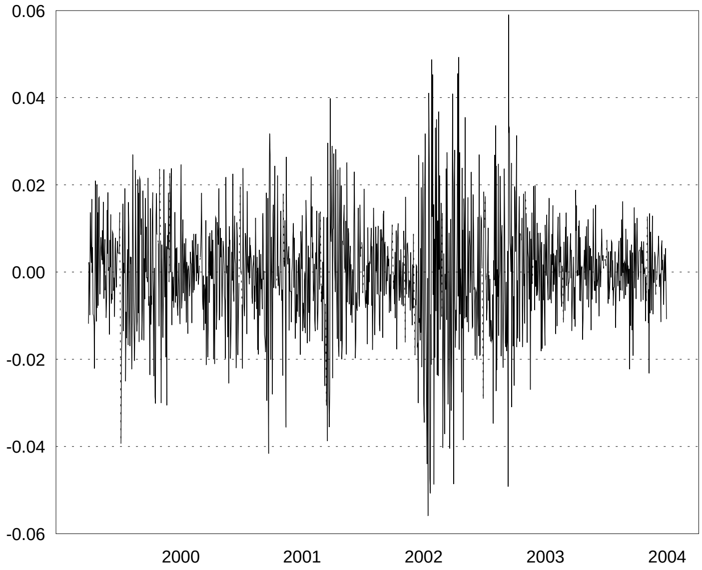
Nikkei225



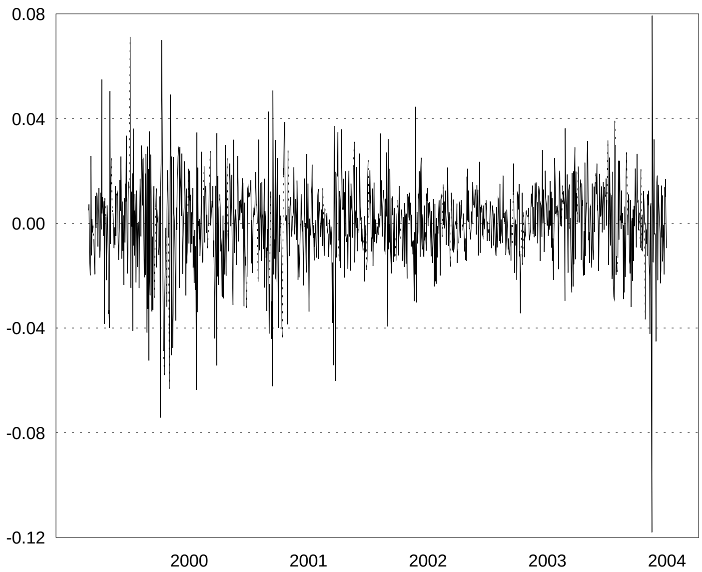
S&P500



FTSE100



BSE30



KLSE