ELSEVIER

Expert Systems with Applications 29 (2005) 927–940

Expert Systems with Applications

www.elsevier.com/locate/eswa

The use of data mining and neural networks for forecasting stock market returns

David Enke*, Suraphan Thawornwong

Laboratory for Investment and Financial Engineering, Smart Engineering Systems Lab, Intelligent Systems Center, University of Missouri, Rolla, MO 65409-0370, USA

Abstract

It has been widely accepted by many studies that non-linearity exists in the financial markets and that neural networks can be effectively used to uncover this relationship. Unfortunately, many of these studies fail to consider alternative forecasting techniques, the relevance of input variables, or the performance of the models when using different trading strategies. This paper introduces an information gain technique used in machine learning for data mining to evaluate the predictive relationships of numerous financial and economic variables. Neural network models for level estimation and classification are then examined for their ability to provide an effective forecast of future values. A cross-validation technique is also employed to improve the generalization ability of several models. The results show that the trading strategies guided by the classification models generate higher risk-adjusted profits than the buy-and-hold strategy, as well as those guided by the level-estimation based forecasts of the neural network and linear regression models.

Keywords: Stock return forecasting; Data mining; Information gain; Neural networks; Trading strategies

1. Introduction

Over the past two decades many important changes have taken place in the environment of financial markets. The development of powerful communication and trading facilities has enlarged the scope of selection for investors (Elton and Gruber, 1991). Traditional capital market theory has also changed and methods of financial analysis have improved (Poddig and Rehkugler, 1996). Forecasting stock return or a stock index is an important financial subject that has attracted researchers' attention for many years. It involves an assumption that fundamental information publicly available in the past has some predictive relationships to the future stock returns or indices. The samples of such information include economic variables such as interest rates and exchange rates, industry specific information such as growth rates of industrial production and consumer price, and company specific information such as income statements and dividend yields. This is opposed to the general perception of market efficiency (Fama, 1970). In fact, the efficient market hypothesis states that all available information affecting the current stock values is constituted by the market before the general public can make trades based on it (Jensen, 1978). Therefore, it is impossible to forecast future returns since they already reflect all information currently known about the stocks. This is still an empirical issue because there is considerable evidence that markets are not fully efficient, and it is possible to predict the future stock returns or indices with results that are better than random (Lo and MacKinlay, 1988).

Recently, Balvers, Cosimano, and McDonald (1990), Breen, Glosten, and Jagannathan (1990), Campbell (1987), Fama and Schwert (1977), Fama and French (1988, 1989), Ferson (1989), Keim and Stambaugh (1986), and Schwert (1990) among others, provide evidence that stock market returns are predictable by means of publicly available information such as time-series data on financial and economic variables, especially those with an important business cycle component. These studies identify that such variables as various interest rates, monetary growth rates, changes in industrial production, and inflation rates are statistically important for predicting a portion of the stock returns. However, most of the above studies attempting to

* Corresponding author. Tel.: +573-341-4565; fax: +573-341-6567
E-mail address: enke@umr.edu (D. Enke).

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2005.06.024

capture the relationship between the available information and the stock returns rely on simple linear regression assumptions. There is no evidence to support the assumption that the relationship between the stock returns and the financial and economic variables is perfectly linear. This is due to the fact that there exists the significant residual variance of the actual stock return from the prediction of the regression equation. Therefore, it is possible that nonlinear models are able to explain this residual variance and produce more reliable predictions of the stock price movements (Mills, 1990; Priestley, 1988).

Since many of the current modeling techniques are based on linear assumptions, a new kind of financial analysis that considers the nonlinear analysis of integrated financial markets needs to be considered. Even though there exists a number of non-linear regression techniques, most of these techniques require that the non-linear model must be specified before the estimation of parameters can be determined. One non-linear modeling technique that may overcome these problems involves the use of neural networks (Hill, O'Connor, and Remus, 1996). In fact, neural networks offer a novel technique that does not require a pre-specification during the modeling process because they independently learn the relationship inherent in the variables. This is especially useful in security investment and other financial areas where much is assumed, and little is known about the nature of the processes determining asset prices (Burrell and Folarin, 1997). Neural networks also offer the flexibility of numerous architecture types, learning algorithms, and validation procedures. As a result, the discovery and use of non-linearity in financial market movements and analysis to produce better predictions of future stock returns or indices has been greatly emphasized by various researchers and financial analysts during the last few years (see Abhyankar, Copeland, and Wong, 1997). Current studies that reflect an interest in applying neural networks to answer future stock behaviors include Chenoweth and Obradovic (1996), Desai and Bharati (1998), Gencay (1998), Leung, Daouk, and Chen (2000), Motiwalla and Wahab (2000), Pantazopoulos et al. (1998), Qi and Maddala (1999), and Wood and Dasgupta (1996).

To this end, it has been found that stock trading driven by a certain forecast with a small forecasting error may not be as profitable as trading guided by an accurate prediction of the sign of stock return (Aggarwal and Demaskey, 1997; Leung et al., 2000; Maberly, 1986; Wu and Zhang, 1997). Nonetheless, having an accurate prediction of a certain stock or stock index return still has numerous benefits. Given the existence of a vast number of articles addressing the predictabilities of stock market return, most of the proposed models relied on various assumptions and often employ a particular series of input variables without justification as to why they were chosen. Obviously, a systematic approach to determine what inputs are important is necessary. In regard to this, the present paper will begin with the discussion of the methodology for data selection and then introduce an information gain data mining technique for performing the variable relevance analysis. Two neural network approaches that can be used for classification and level estimation will also be briefly reviewed in the third section, followed by a discussion of the neural network models, including the generalized regression, probabilistic, and multi-layer feed-forward neural networks that were developed to estimate the value (level) and classify the direction (sign) of excess stock returns on the S&P 500 stock index portfolio. In addition, the five-fold cross validation and early stopping techniques were also implemented in this study to improve the generalization ability of the feed-forward neural networks. The resulting data selection and model development, empirical results, and discussion and conclusion will then be presented, respectively. Finally, the data sources and descriptions are given in the Appendix.

2. Methodology for data selection

In general, large-scale deterministic components, such as trends and seasonal variations, should be eliminated from the inputs since the network will attempt to learn the trend and use it in the prediction (Nelson et al., 1999; Pantazopoulos et al., 1998). Therefore, the data collected in this study, excluding *DIV*, *T1*, *SP*, *DY*, and *ER*, were seasonally adjusted allowing the networks to concentrate on the important details necessary for an accurate prediction (the source and definition of all the variables are given in the Appendix). In addition, due to the lag associated with the publication of macroeconomic indicators as mentioned by Qi and Maddala (1999), certain data, particularly *PP*, *IP*, *CP*, and *M1*, were included in the base set with a two-month time lag while the rest of the variables were included with a one-month time lag. Constructing the data in this manner ensures that the forecasting models using these variables will be similar to real-world practice. Specifically, only observable, but not future data were employed as inputs to the forecasting models. As a result, these time lags were used throughout the experiment to maintain realistic situations when data are gathered.

In this study, the differences $[P_t - P_{t-1}]$ of variables were provided to the networks so that different input variables can be compared in terms of change to the monthly stock returns, since the level changes of the variable may be more meaningful to the models than the original values when forecasting a financial time series. Monthly data from March 1976 to December 1999, for a total of 286 periods and for each of 31 financial and economic variables, were collected and analyzed. These variables, including PP_{t-1}, CP_{t-1}, IP_{t-1}, $M1_{t-1}$, $T3_t$, $T6_t$, $T12_t$, $T60_t$, $T120_t$, $CD1_t$, $CD3_t$, $CD6_t$, AAA_t, BAA_t, DIV_t, $T1_t$, SP_t, DY_t, $TE1_t$, $TE2_t$, $TE3_t$, $TE4_t$, $TE5_t$, $TE6_t$, $DE1_t$, $DE2_t$, $DE3_t$, $DE4_t$, $DE5_t$, $DE6_t$, and $DE7_t$, were primarily employed to predict the level and to classify the sign of the excess stock returns (ER_{t+1}) on the S&P 500 index portfolio. These data consisted of a mixture of the variables conducted by various researchers, including Desai and Bharati (1998), Leung et al. (2000), Motiwalla and Wahab (2000), and Qi and Maddala (1999). However, two variables often used in the literature, long-term treasury rates and commercial papers, were not applicable due to the fact that the 30-year treasury rate provided by the Federal Reserve Board of Governors started from February 1977, while the series of commercial papers had been discontinued because of a change in methodology in September 1997. Therefore, several financial instruments, such as CD and T-bill rates with additional maturities were included to supplement unavailable data in this study.

While uncertainty in selecting the predictive variables to forecast stock returns still exits, as can be observed from a variety of input variables used in a recent literature survey, several techniques such as regression coefficients (Qi and Maddala, 1999), autocorrelations (Desai and Bharati, 1998), backward stepwise regression (Motiwalla and Wahab, 2000), and genetic algorithms (Motiwalla and Wahab, 2000) have been employed by a few studies to perform variable subset selection. In addition, several researchers, such as Leung et al. (2000), Gencay (1998), and Pantazopoulos et al. (1998), subjectively selected the subsets of variables based on empirical evaluations. None of these studies, however, have incorporated all available variables previously mentioned in the literature to uncover input data that may be effective in predicting stock returns. Obviously, many of variables gathered in these studies may be irrelevant or redundant to the prediction of stock returns. In fact, leaving out relevant variables or keeping irrelevant variables may be detrimental, causing confusion to the neural network models. Besides, the use of many variables would require a neural network that contains an excessive number of neurons and possibly many hidden layers. It is fundamentally impossible to train such a network if few data are available during its modeling. Unfortunately, there is no consistent method that has been used to pick out the useful variables in stock return forecasting. This may be due to the fact that the behavior of this data is not well known.

One alternative that can be used to extract valuable information and knowledge from large amounts of data involves the use of data mining (Han and Micheline, 2000). Specifically, there have been many studies in some areas of data mining (i.e., machine learning, fuzzy logic, statistics, and rough set theories) on variable relevance analysis for data understanding. This relevance analysis, therefore, may be performed on the financial data with the aim of removing any irrelevant or redundant variables from the learning process. The general idea behind variable relevance analysis is to compute some measures that can be used to quantify the relevance of variables hidden in a large data set with respect to a given class or concept description. Such measures include information gain, the Gini index, uncertainty, and correlation coefficients. For this study, an inductive learning decision tree algorithm that integrates an information gain analysis technique with a dimension-based data analysis method was selected as it can be effectively used for variable subset selection (Han and Micheline, 2000). The resulting method removes the less information producing variables and collects the variables that contain more information. Therefore, it may be the most appropriate data mining technique to perform variable subset selection when the usefulness of the data is unknown. While using the information gain analysis technique, the predicted directions of excess stock returns were used as class distributions for the experiment. The resulting variables with the high information gain were chosen as the relevant input variables provided to the neural network models. The following paragraphs give a brief introduction to the information gain calculation. It is recommended that readers who are interested in full details of the information gain algorithm should refer to Quinlan (1993).

Let S be a set consisting of s data samples. Suppose the class label variable has m distinct values defining m distinct classes, C_i (for $i=1, 2,\ldots, m$). Let s_i be the number of samples of S in class C_i. The expected information needed to classify a given sample is given by:

$$I(s_1, s_2, s_3, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

where p_i is the probability that an arbitrary sample belongs to class C_i and is estimated by s_i/s. Note that a log function to the base 2 is used since the information is encoded in bits. Let variable A have v distinct values denoted in order from small to large values as $\{a_1, a_2, a_3,\ldots, a_v\}$. Any split value lying between a_i and a_{i+1} will have the same effect of dividing the samples into those whose value of the variable A lies in $\{a_1, a_2, a_3,\ldots, a_i\}$ and those whose value is in $\{a_{i+1}, a_{i+2}, a_{i+3},\ldots, a_v\}$. However, the midpoint of each interval is usually chosen as the representative split. It is defined as $(a_i+a_{i+1})/2$. Thus, there are $v-1$ possible splits on A, all of which are examined. Note that examining all $v-1$ splits is necessary to determine the highest information gain of A.

Variable A can therefore be used to partition S into 2 subsets, $\{S_1, S_2\}$, where S_j contains those samples in S that have values $\{a_1, a_2, a_3,\ldots, a_i\}$ or $\{a_{i+1}, a_{i+2}, a_{i+3},\ldots, a_v\}$ of A. Let S_j contain s_{ij} samples of class C_i. The expected information based on this partitioning by A, also known as the "entropy" of A, is given by:

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + s_{2j} + \ldots + s_{mj}}{s} I(s_{1j}, s_{2j}, \ldots, s_{mj}).$$

The term $(s_{1j}+s_{2j}+\ldots+s_{mj})/s$ acts as the weight of the j^{th} subset and is the number of samples in the subset (i.e., having value a_j of A) divided by the total number of samples in S. Note that for a given subset S_j,

$$I(s_{1j}, s_{2j}\ldots, s_{mj}) = -\sum_{i=1}^{m} p_{ij} \log_2(p_{ij})$$

where $p_{ij} = s_{ij}/|S_j|$ and is the probability that a sample in S_j belongs to class C_i. The information gain obtained by this partitioning of the split on A is defined by:

$$\mathrm{Gain}(A) = I(s_1, s_2, s_3, \ldots, s_m) - E(A).$$

In this approach to relevance analysis, the highest information gain for each of the variables defining the samples in S can be obtained. The variable with the highest information gain is considered the most discriminating variable of the given set. By computing the information gain for each variable, a ranking of the variables can be obtained. Finally, the relevant threshold is determined to select only the strong relevant variables to be used in the forecasting models.

In this study, each of the neural network models was compared against a linear regression model, as well as a buy-and-hold strategy. For all models, the data set used in this study was divided into two periods: the first period runs from March 1976 to October 1992 for a total of 200 months while the second period runs from November 1992 to December 1999 for a total of 86 months. The former was used for determining the specifications of the models and parameters of the forecasting techniques. The latter was reserved for out-of-sample evaluation and comparison of performances among the forecasting models.

3. Neural network models

The theory of neural network computation provides interesting techniques that mimic the human brain and nervous system. A neural network is characterized by the pattern of connections among the various network layers, the numbers of neurons in each layer, the learning algorithm, and the neuron activation functions. Generally speaking, a neural network is a set of connected input and output units where each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to correctly predict or classify the output target of a given set of input samples. Given the numerous types of neural network architectures that have been developed in the literature, three important types of neural networks, including the generalized regression, probabilistic, and multi-layer feed-forward neural networks were implemented in this study to compare their predictive ability against the classical linear regression model. The following three subsections give a brief introduction of these three neural network models.

3.1. Multi-layer feed-forward neural network

Multi-layer feed-forward neural networks have been widely used for financial forecasting due to its ability to correctly classify and predict the dependent variable (Vellido, Lisboa, and Vaughan, 1999). For each training sample, the input variables are fed simultaneously into a layer of units making up the input layer. The weighted outputs of these units are, in turn, fed simultaneously to a second layer of units known as a hidden layer. The hidden layer's weighted outputs can be input to another hidden layer, and so on. The weight outputs of the last hidden layer are input to units making up the output layer which issues the network's prediction for a given set of samples.

Backpropagation is by far the most popular neural network algorithm that has been used to perform training on the multi-layer feed-forward neural networks. It is a method for assigning responsibility for mismatches to each of the processing elements in the network by propagating the gradient of the activation function back through the network to each hidden layer down to the first hidden layer. The weights are then modified so as to minimize the mean squared error between the network's prediction and the actual target. Since the feed-forward neural networks are well known, the network structures and backpropagation algorithms are not described in this paper. However, readers who are interested in greater detail can refer to Rumelhart and McClelland (1986) for a comprehensive explanation of the backpropagation algorithm used to train multi-layer feed-forward neural networks.

During neural network modeling, Malliaris and Salchenberger (1993) suggest that validation techniques are required to identify the proper number of hidden layer nodes, thus avoid under-fitting (too few neurons) and over-fitting (too many neurons) problems. Generally, too many neurons in the hidden layers, hence, too many connections, produce a neural network that memorizes the data and lacks the ability to generalize. One approach that can be used to avoid over-fitting is n-fold cross-validation (Peterson, St Clair, Aylward, and Bond, 1995). A five-fold cross-validation, which was used in this experiment, can be described as follows: The data sample is randomly partitioned into five equal-sized folds and the network is trained five times. In each of the training passes, one fold is omitted from the training data and the resulting model is validated on the cases in that omitted fold, which is also known as a validation set. The first period (200 months) of the data set is used for the five-fold cross-validation experiment, leaving the second period for truly untouched out-of-sample data. The average root-mean squared error over the five unseen validation sets is normally a good predictor of the error rate of a model built from all the data.

Another approach that can be used to achieve better generalization in trained neural networks is called early stopping (Demuth and Beale, 1998). This technique can be effectively used with the cross-validation experiment. The validation set is used to decide when to stop training. When the network begins to over-fit the data, the error on the validation cases will typically begin to rise. In this study, the training was stopped when the validation error increased for five iterations, causing the return of the weights and biases to the minimum of the validation error. The average error results of the validation cases (40 months in each fold for this study) from the n-fold cross-validation experiment can finally be used as criteria for determining the optimal network structure, namely the number of hidden layers, number of neurons, learning algorithms, learning rates, and activation functions.

3.2. Generalized regression neural network

While a number of articles address the ability of the multi-layer feed-forward neural network model with back-propagation for financial forecasting, none of these studies has practically applied the generalized regression neural network (GRNN) to forecast stock returns. Like the feed-forward neural networks, the GRNN can be used for function approximation to estimate the values of continuous dependent variables such as future position, future values, and multivariable interpolation. The GRNN is a kind of radial basis function network and also looks similar to a feed-forward neural network responding to an input pattern by processing the input variables from one layer to the next with no feedback paths (Specht, 1991). However, its operation is fundamentally different. The GRNN is based on nonlinear regression theory that can be used for any regression problem in which an assumption of linearity is not justified.

The training set contains the values of x (independent variables) that correspond to the value of y (dependent variable). This regression method will produce the optimal expected value of y, which minimizes the mean squared error. The GRNN approach uses a method that frees the necessity to assume a specific functional form. In fact, it allows the appropriate form to be expressed as a probability density function that is empirically determined from observed data using the window estimation (Parzen, 1962). Therefore, this approach is not limited to any particular forms and requires no prior knowledge of the estimated function. The GRNN formula is briefly described as follows:

$$E[y|x] = \frac{\int_{-\infty}^{\infty} y f(x,y)dy}{\int_{-\infty}^{\infty} f(x,y)dy}$$

where y is the output of the estimator, x is the estimator input vector, $E\,[y|x]$ is the expected value of y given x, and (x, y) is the known joint continuous probability density function of x and y. When the density (x, y) is not known, it will be estimated from a sample of observations of x and y. For a nonparametric estimate of (x, y), the class of consistent estimators proposed by Parzen (1962) is used. As a result, the following equation gives the optimal expected value of y:

$$y = \frac{\sum_{i=1}^{n} h_i w_i}{\sum_{i=1}^{n} h_i}$$

where w_i is the target output corresponding to the input training vector x_i and the output y, $h_i = \exp[-D_i^2/(2\sigma^2)]$ is the output of hidden neuron, $D_i^2 = (x-u_i)^T(x-u_i)$ is the squared distance between the input vector x and the training vector u, and σ is a smoothing parameter of the radial basis function. The GRNN architecture is shown in Fig. 1. The neurons of hidden layer 1 are created to hold the input vectors. The weights between the newly created hidden neurons and the neurons of hidden layer 2 are assigned the target value.

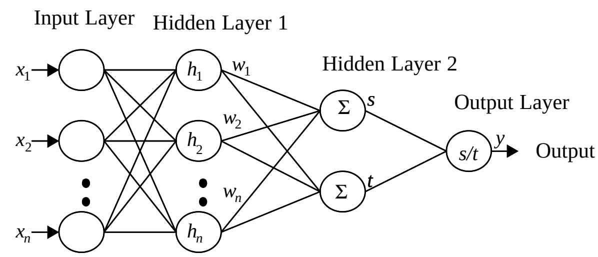

Fig. 1. Generalized regression neural network architecture.

4. Probabilistic neural network

In contrast to the GRNN used to estimate the values of continuous variables, the probabilistic neural network (PNN) finds decision boundaries between categories of patterns. Therefore, the PNN is mainly used for classification problems. The PNN is a parallel implementation of a standard Bayesian classifier and has a four-layer network that can perform pattern classification. It is based essentially on the estimation of probability density functions for various classes learned from training samples. The PNN learns from the sample data instantaneously and uses these probability density functions to compute the nonlinear decision boundaries between classes in a way that approaches the Bayes optimal (Specht, 1990). The PNN formula can be briefly explained as follows:

$$f_A(x) = \frac{1}{(2\pi)^{P/2}\sigma^P n}\sum_{i=1}^{n} z_i$$

where $f_A(x)$ is the probability density function estimator for class A, p is the dimensionality of training vector, $z_i = \exp[-D_i/(2\sigma^2)]$ is the output of hidden neuron, $D_i = (x-u_i)^T(x-u_i)$ is the distance between the input vector x and the training vector u from category A, and σ is a smoothing parameter.

Theoretically, the PNN can classify an out-of-sample data with the maximum probability of success when enough training data is given (Wasserman, 1993). Fig. 2 presents the PNN architecture. When an input is presented to the hidden layer 1, it computes distances from the input vector to the training vectors and produces a vector whose elements indicate how close the input is to the vectors of the training set. The hidden layer 2 then sums these elements for each class of inputs to produce a vector of probabilities as its net output. Finally, the activation function of the PNN output

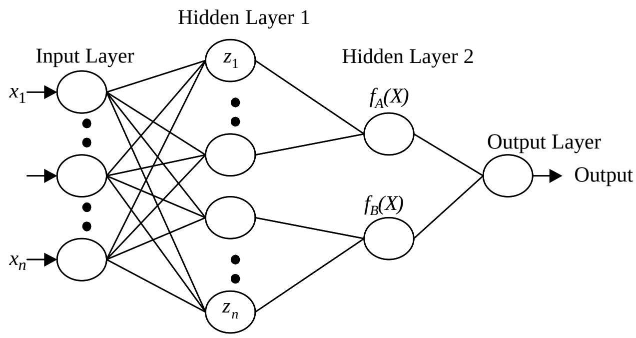

Fig. 2. Probabilistic neural network architecture.

layer picks the maximum of these probabilities and classifies it into specific output classes.

5. Data selection and model development

The selection of the input variables is a modeling decision that can greatly affect the model performance. In the neural network environment, the information gain data mining analysis was used to find good subsets of the full set of the first-period input variables. Of the 31 variables, 15 variables were selected by the information gain data mining analysis as strong relevant predictors for the data set used in this study. They include *M1*, *T3*, *T6*, *T120*, *CD1*, *CD3*, *CD6*, *SP*, *TE2*, *TE3*, *TE4*, *DE2*, *DE3*, *DE5*, and *DE7*. Thus, these variables were consistently used as the input variables for training the neural networks throughout the modeling stage. The values of the input variables were first preprocessed by normalizing them within a range of -1 and $+1$ to minimize the effect of magnitude among the inputs and thus increase the effectiveness of the learning algorithm.

It is well known that most trading practices adopted by financial analysts rely on accurate prediction of the price levels of financial instruments. Nonetheless, some recent studies have suggested that trading strategies guided by forecasts on the direction of the change in price level may be more effective and thus can generate higher profits. Aggarwal and Demaskey (1997) report that the performance of cross hedging improves significantly if the direction of changes in exchange rates can be predicted. In another study, Maberly (1986) explores the relationship between the direction of inter-day and intra-day price changes on the S&P 500 futures. Wu and Zhang (1997) investigate the predictability of the direction of change in the future spot exchange rate. Recently, Leung et al. (2000) find that the forecasting models based on the direction of stock return outperform the models based on the level of stock return in terms of predicting the direction of stock market return and maximizing profits from investment trading.

The above-mentioned studies demonstrate the usefulness of forecasting the direction of change in the price or return level by means of a gain or a loss. In fact, the results of these findings are reasonable because accurate price estimation, as determined by its deviation from the actual observation, may not be a good predictor of the direction of change in the price levels of financial instruments. To facilitate a more effective forecast developed in this study, the two forecasting approaches, namely classification and level estimation, were investigated to evaluate the resulting performances of the model development. Specifically, the feed-forward neural networks were developed to both estimate the value (level) and classify the direction (sign) of excess stock returns on the S&P 500 index portfolio. Based on their network architectures, however, the GRNN was used to estimate the level of excess stock return while the PNN was employed to classify the sign of excess stock return in this study. Finally, the conventional linear regression model was developed to serve as a benchmark for performance comparison purposes. Note that the second period test data were never used during the model development so that these forecasting models were always tested on truly untouched out-of-sample data.

5.1. Neural network models for level estimation

For the feed-forward neural network using the back-propagation algorithm, a sigmoid hyperbolic tangent function was selected as the activation function to generate an even distribution over the input values. A single hidden layer was chosen for the neural network model since it has been successfully used for financial classification and prediction (Swales and Yoon, 1992). Accordingly, the feed-forward neural network was built with three layers including the input layer, hidden layer, and output layer. Each of the relevant 15 input variables was assigned a separate input neuron to the input layer of the feed-forward neural network. One output neuron was used in the output layer to represent the predicted excess stock return of a given set of the 15 input variables. The network configuration of this level estimation experiment is given in Fig. 3. In this study, the connection weights were initially randomized and then determined during the backpropagation training process.

After many experiments with various numbers of hidden layer neurons, learning algorithms, and learning rates,

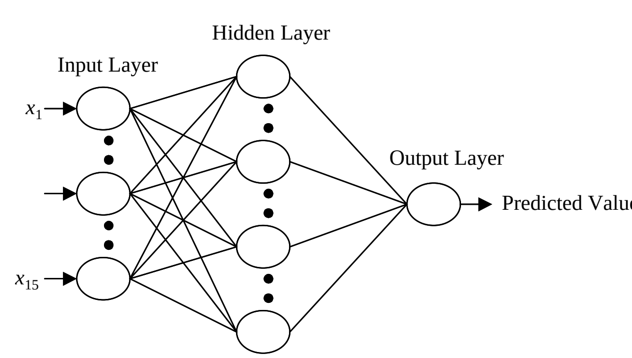

Fig. 3. A three layer feed-forward neural network for level estimation.

the feed-forward neural network employing 15 neurons in the input-layer, 21 neurons in the hidden layer, 0.2 learning rate, and a resilient backpropagation training algorithm was found to be the best network architecture based on the lowest average root-mean squared error over the five-fold cross-validation experiment. In other words, this network architecture generated the lowest average root-mean squared error over the cases of five omitted folds (validation sets) in this study. Note that the root-mean squared error (RMSE) used in the feed-forward neural network for level estimation is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - t_i)^2}$$

where y_i is the predicted excess stock return, t_i is the actual excess stock return, and n is the number of validation cases (40 in this study). The average RMSE results were calculated only after the neural network outputs have been scaled back to their normal values. By conducting the five-fold cross-validation experiment, the forecasting results will not be based on a single network output because five neural network models were developed from the five different data sets. For this reason, the predicted excess stock returns of the five network outputs were averaged to generate the weighted excess return in this experiment. As a result, the weighted excess return (*WER*) can be derived as:

$$WER = \frac{y_1 + y_2 + y_3 + y_4 + y_5}{5}$$

where y_1, y_2, y_3, y_4, and y_5 are the predicted excess stock returns of the five neural network models developed from the five-fold cross-validation experiment.

To further improve the forecasting performance, we also examined a portfolio network model consisting of the network architecture that produced the lowest RMSE in each omitted fold cross-validation experiment. In other words, the neural network model generating the lowest RMSE from each omitted fold experiment was chosen as one of the five neural networks deliberately combined as the portfolio network model. The resulting portfolio network architectures using the lowest RMSE in each omitted fold experiment are provided in Table 1. It is observed that the suitable neurons used in the hidden layer of the five combined portfolio networks that were trained based on different omitted folds are different. This observation suggests the importance of network modeling for a separate omitted fold experiment because the potentially better trained neural network may be obtained from the specific validation cases. Again, the *WER* of the portfolio network model was calculated from the five combined portfolio network outputs.

Table 1
Portfolio neural network model for level estimation

Omitted Folds	Input layer neurons	Hidden layer neurons	Learning rate
1	15	23	0.3
2	15	27	0.2
3	15	24	0.3
4	15	11	0.2
5	15	21	0.2

Unlike the feed-forward neural networks, the GRNN can be designed very quickly, and no early stopping technique is required during its training. Therefore, there would be no need to randomly partition the data into equal-sized folds for cross-validation. As a result, the first period (200 months) of the data set was used in network training for predicting the excess stock returns of the last 86 months. In this study, a smoothing parameter of the radial basis function equal to 1.00 was selected to approximate the network function more efficiently. The GRNN training process employed the same input variables, pre-processing techniques, and post-processing techniques as those of the feed-forward neural network models.

5.2. Neural network models for classification

Other than the output layer structure, the feed-forward neural network for classification employed the same network structures as those used for level estimation. Since there are two classes of the signs of excess stock return, two output neurons were employed for the output layer to represent the different classes of the predicted excess stock return. In this study, the vectors $[+1\ -1]$ and $[-1\ +1]$ represented the predicted positive and negative signs of excess stock return, respectively. Fig. 4 provides the network configuration of the feed-forward neural network for classification. The output neuron with the highest value was taken to represent the predicted sign of excess stock return based on a given set of the 15 input variables.

In this experiment, the feed-forward neural network employing 15 neurons in the input layer, 27 neurons in the hidden layer, 0.3 learning rate, and a resilient back-propagation training algorithm was found to be the best network architecture with the lowest average root-mean squared error over the five-fold cross-validation experiment.

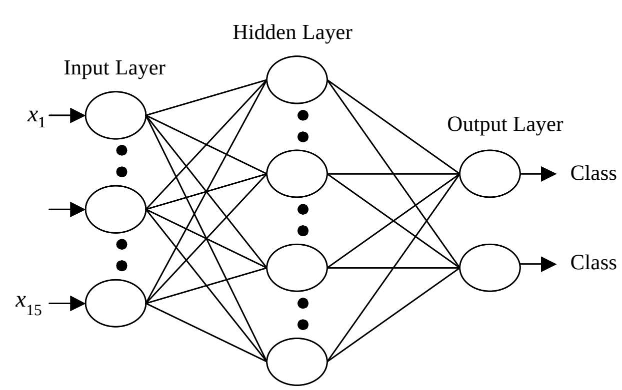

Fig. 4. A three layer feed-forward neural network for classification.

The root-mean squared error (RMSE) used in the feed-forward neural network for classification is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{2n}\sum_{i=1}^{n}\{(y_1 - t_1)^2 + (y_2 - t_2)^2\}}$$

where y_1 and y_2 are the predicted classes of excess stock return of the two output neurons, t_1 and t_2 are the actual classes of excess stock return, and n is the number of validation cases. Just like the previously developed feed-forward neural network models for level estimation, the forecasting results will be based on five network outputs. Therefore, the majority of the signs of five network outputs is used to determine the decisive predicted sign of excess stock return. For example, when the five network models generate three positive predicted signs and two negative predicted signs of excess stock return based on a given set of the 15 input variables, the decisive predicted sign of excess stock return is resolved to be positive.

In addition, a portfolio network model for classification that consists of the network architecture producing the lowest RMSE in each omitted fold cross-validation experiment was explored. The resulting portfolio network architectures using the lowest RMSE in each omitted fold experiment are given in Table 2. As can be seen, the suitable hidden layer neurons of the five combined portfolio networks are different implying the similar observation of those of the portfolio network model for level estimation. Similarly, the decisive predicted sign of excess stock return of the portfolio network model was derived from the majority of the five combined portfolio network outputs.

Like the GRNN, the design of the PNN is fast and straightforward. In fact, neither training nor early stopping technique is required during its design. Therefore, the first period (200 months) of the data set was used in network modeling for predicting the sign of the excess stock returns of the last 86 months. Also, a smoothing parameter equal to 1.00 was selected to entirely consider several nearby design vectors. Again, the PNN design employed the same input variables and pre-processing techniques as those of the feed-forward neural network models.

5.3. Linear regression for level estimation

In the linear regression forecasting, the backward stepwise regression for dimensionality reduction was employed to assume a linear additive relationship. This method started with the full set of variables in the model. The worst of the original variables was determined and removed from the full set. At each subsequent iteration or step, the worst of the remaining variables was removed from the last updated set. The significant *t*-statistics were used as criteria for retention of the significant input variables in the linear regression model. The remaining variables were thus used in predicting excess stock returns. In this study, the backward stepwise technique kept 10 variables, *PP*, *M1*, *T3*, *T12*, *T60*, *CD1*, *CD6*, *BAA*, *SP*, and *DE7*, as the significant input variables in the regression model ($\alpha = 0.05$). The regression model has the following function:

$$\begin{aligned} ERt+1 = {} & -0.444 + (0.959 \times PP_{t-1}) + (0.100 \times M1_{t-1}) \\ & + (2.525 \times T3_t) + (5.981 \times T12_t) \\ & + (-4.584 \times T60_t) + (-1.050 \times CD1_t) \\ & + (-5.472 \times CD6_t) + (-1.437 \times BAA_t) \\ & + (-0.027 \times SP_t) + (8.295 \times DE7_t) \end{aligned}$$

where all the regression coefficients are significant and the F-statistic is 2.027 (*p-value* 0.033), indicating that these forecasting variables contain information about future excess stock returns (F-critical $=1.91$). The regression model shows that the changes of *PP*, *M1*, *T3*, *T12*, and *DE7* have a positive effect on predictions of excess stock return, whereas the effect on excess stock returns of *T60*, *CD1*, *CD6*, *BAA*, and *SP* is negative.

Table 2
Portfolio neural network model for classification

Omitted Folds	Input layer neurons	Hidden layer neurons	Learning rate
1	15	21	0.3
2	15	19	0.2
3	15	28	0.3
4	15	27	0.3
5	15	23	0.3

6. Results

The predictive performances of the developed models were evaluated using the untouched out-of-sample data (second period). This is due to the fact that the superior in-sample performance does not always guarantee the validity of the forecasting accuracy. One possible approach for evaluating the forecasting performance is to investigate whether traditional error measures such as those based on the RMSE or correlation (CORR) between the actual out-of-sample returns and their predicted values are small or highly correlate, respectively. However, there is some evidence in the literature suggesting that traditional measures of forecasting performance may not be strongly related to profits from trading (Pesaran and Timmermann, 1995). An alternative approach is to look at the proportion of time that the signs of excess stock returns (SIGN) are correctly predicted. In fact, Leitch and Tanner (1991) state that the forecasting performance based on the sign measure matches more closely to the profitability performance than do traditional criteria.

Therefore, in Table 3, we report all of the three performance measures of the original level estimation

Table 3
Second period performance measures

		CORR	RMSE	SIGN
Level Estimation Models	Original Level NN	0.0231	1.1614	0.6628*
	Portfolio Level NN	0.0528	1.1206	0.6860*
	GRNN	0.0714	1.1206	0.6860*
	Regression	0.0300	1.4467	0.4767
Classification Models	Original Class NN	0.2300	1.2200	0.6279*
	Portfolio Class NN	0.3150	1.0997	0.6977*
	PNN	0.3020	1.2575	0.6047*

feed-forward neural network using the lowest average RMSE (Original level NN), the portfolio level estimation feed-forward neural network using the lowest RMSE in each omitted fold (Portfolio level NN), the GRNN, the linear regression model (Regression), the original classification feed-forward neural network using the lowest average RMSE (Original Class NN), the portfolio classification feed-forward neural network using the lowest RMSE in each omitted fold (Portfolio Class NN), and the PNN from November 1992 to December 1999. RMSE in Table 3 represents the root-mean squared error between the actual and predicted signs of excess stock return. CORR refers to the Pearson correlation coefficient between the actual and predicted signs of excess stock return (Pesaran and Timmermann, 1992). SIGN denotes the proportion of times the predicted signs of excess stock returns are correctly classified. Note that the $+1$ and -1, representing the positive and negative decisive predicted signs of the PNN and the classification feed-forward neural networks, were used to compute the resulting classification performances in the study. To compare the classification performances with those of the Regression, the GRNN, and the feed-forward neural networks for level estimation, the original RMSE and CORR performance measures of these level estimation models were recalculated in connection with the signs of $+1$ and -1 of the classification models. That is, when the positive predicted value of excess stock return is generated by the level estimation models, it will be converted to the $+1$ or vice versa. The reason for this recalculation is that the PNN model is designed to give the exact signs of $+1$ and -1. Therefore, the prediction of the other forecasting models is required to adjust for unbiased performance comparisons.

According to Table 3, the empirical results show that neither the classification nor the level estimation neural network models can accurately predict the signs of excess stock return because of the relatively low correlation relationship, although each of these models, excepting the Original Level NN model, is unquestionably better than the model using linear regression. This is due to the fact that the CORR of these models indicates higher positive relationship between the actual and predicted signs of excess stock return. It is also observed that the CORR of the classification models is constantly better than that of the level estimation models. In particular, the Portfolio Class NN has the highest CORR (0.3150) that can be obtained from the experiment. This reveals that the neural networks, especially the classification models, perform more accurately in correctly predicting the portion of future excess stock returns.

Regarding the second performance measure, the results again confirm that the linear regression model is the least accurate performer because it generates the highest RMSE (1.4467) compared to that of the neural network models. In contrast, the Portfolio Class NN model produces the lowest RMSE (1.0997). Nonetheless, the remaining two classification models, the Original Class NN and PNN models, signal slightly higher RMSE results than those of the level estimation neural network models. For the third performance measure, the results show that the percentage of the correct signs (SIGN) generated by the neural network models is far more accurate and consistently predictive than that of the linear regression forecast. This is because the correct signs produced by all of the neural network models are always either greater than or equal to 0.6047. For statistical evaluation, the null hypothesis of no predictive effectiveness was calculated by conducting a one-sided test of H_o: $p=0.50$ against H_a: $p > 0.50$. The SIGN marked with an asterisk (*) in Table 3 indicates the significant differences from the benchmark of 0.5 at a 95% level of confidence. More importantly, the Portfolio Class NN model once again signals the highest SIGN (0.6977) obtainable from the study, whereas the linear regression forecast has obtained only 0.4767 of the correct signs. This result verifies that the correct signs generated by each neural network model are better than random. In summary, the overall out-of-sample forecasts using the GRNN and Portfolio Class NN models are more accurate than those using the Original Level NN and Portfolio Level NN, Original Class NN, PNN, and Regression models with respect to their approaches. Particularly, the Portfolio Class NN model is proven to be the best performer in all of the performance measures used in this study. These findings strongly support the non-linearity relationship between the past financial and economic variables and the future stock returns in the financial markets.

6.1. Trading simulation

A trading simulation was developed in an effort to further examine if the neural network models could practically be used to generate higher profits than those earned either by employing the traditional regression model or by simply following a buy-and-hold (passive) investment strategy. The operational details of the trading simulation are explained as follows: The trading simulation assumes that in the beginning of each monthly period the investor makes an asset allocation decision of whether to shift assets into T-bills or an S&P 500 stock portfolio. It should be noted that the S&P 500 is the value-weighted index in which its index level can be used to perfectly track capital gains on the underlying portfolio if investors were to buy each share in the index in proportion to its outstanding market value. This strategy, which seems impossible for small investors, is applicable for portfolio and fund managers who have the ability to shape a one-month ahead asset allocation in the equity markets. In spite of this limitation, small investors can purchase shares in index mutual funds that hold shares in proportion to their representation in the S&P 500. Finally, it is assumed that the money invested in either T-bills or a stock portfolio becomes illiquid and remains detained in that security until the end of the month.

In the beginning of each month the investor has to decide whether to purchase the S&P 500 portfolio or T-bills, depending on whether the predictions generated by the forecasting models call for a positive or negative excess return result in the next month, respectively. Note that the reason why the excess stock return generated by the forecasting models is used as a trading decision, as compared to a risk free return in a one-month T-bill, is that it provides a measure of how well the models perform relative to the minimum returns gained from depositing the money in a risk-free account (Leung et al., 2000). The above-mentioned strategies imply full investment in either a stock or T-bill for the whole month. Leveraging or short selling when investing is not allowed in this study, since several factors such as up-tick exchange rules, dividends paid during short selling, and margin calls must be considered to reflect a more realistic trading practice. Dividends and transaction costs are also ignored for this study. It should also be noted that the trading strategies between the level estimation and classification models are slightly different because of the unique nature of their approaches. The following describes the trading strategies developed for both level estimation and classification models.

Trading strategy for level estimation models:

If $ER_{t+1} > 0$, then
  Fully invest in stocks or maintain, and receive the actual stock return for the period $t+1$ (R_{t+1});
  Else (if $ER_{t+1} \leq 0$), then
    Fully invest in Treasury bills or maintain, and receive the actual Treasury bill return for the period $t+1$ ($T1H_t$)

where *ER* is the excess stock return given by the forecasting models.

Trading strategy for classification models:

If $C_{t+1} = +1$, then
  Fully invest in stocks or maintain, and receive the actual stock return for the period $t+1$ (R_{t+1});
Else (if C_{t+1}=-1), then
  Fully invest in Treasury bills or maintain, and receive the actual Treasury bill return for the period t+1 ($T1H_t$)

where *C* is the sign of excess stock return given by the forecasting models.

Inspired by the notation that predictability with small forecasting errors may not necessarily imply profitability, another experiment is adapted based on the sign of forecasted excess stock returns using the Portfolio Level NN model. In contrast to the Portfolio Level NN model which uses the weighted excess return (*WER*) calculated from the five network outputs of the neural network models with the lowest RMSE from each omitted fold experiment, this new portfolio model (Sign Port Level NN) directs the trading based on the majority of the signs of the five network outputs obtained from the Portfolio Level NN model. After performing the trading simulations, the resulting mean or monthly return on investment, standard deviation, and Sharpe ratio generated from each forecasting model over the second period (86 months) are calculated and presented in Table 4.

The monthly returns gained from always investing in a stock portfolio (Buy-and-Hold) or a short term T-bill

Table 4
Simulated trading results

		Monthly Return	Std. of Return	Sharpe Ratio	Equal-variance
NN Level Estimation Models	Original Level NN	1.55	3.56	0.33	1.20
	Portfolio Level NN	1.58	3.61	0.34	1.20
	GRNN	1.47	3.64	0.30	1.12
	Sign Port Level NN	1.62	3.55	0.35	1.25
NN Classification Models	Original Class NN	1.51	2.99	0.38	1.32
	Portfolio Class NN	1.72	3.16	0.43	1.43
	PNN	1.26	2.67	0.33	1.20
Benchmark	Regression	0.89	2.49	0.21	0.89
	Buy-and-Hold	1.54	3.68	0.32	1.16
	T-bill	0.37	-	-	-

(see *T1H* in the Appendix) are provided as the benchmark for performance comparisons in this study. According to Table 4, the buy-and-hold strategy has a monthly return of 1.54% over the last 86-month period, and the monthly return on the risk-free T-bill account is approximately 0. 37%. The trading results show that the GRNN, Original Class NN, PNN, and Regression models generate less monthly returns than that of the buy-and-hold account. In fact, the resulting monthly return guided by the regression forecast is 0.81% less than that of the buy-and-hold account. This result suggests that the classical linear regression model and some neural network models cannot effectively be used to account for an accurate forecast of stock returns. On the other hand, slightly higher monthly returns over the buy-and-hold account can be obtained from trading driven by the Original Level NN and Portfolio Level NN models. Noteworthy, it is found that the monthly return generated by the Portfolio Class NN and Sign Port Level NN forecasts are better than those of the other forecasting models with respect to their approaches.

One observation that can be made about the results is that the S&P 500 Index during the trading period presents an episode of significant rise in stock prices (bull market). In fact, its return is approximately equal to an annualized 18. 46% over the trading period. This may be the reason why several forecasting models developed could not significantly achieve better performance than that of the buy-and-hold account during this period. Fig. 5 illustrates samples of cumulative investment return over the second period forecast (from November 1992 to December 1999) guided by the Portfolio Class NN, Regression, Buy-and-Hold, and T-bill models. As can be perceived from Fig. 5, the cumulative investment return of the Regression model is slightly higher than that of the buy-and-hold account in the early period of the forecasting months (from November 92 to September 95). Disappointedly, the cumulative investment return of the Regression model keeps increasing at the declining rate as compared to that of the buy-and-hold account in the later period of the forecasting months (after September 95). This may be due to the fact that the Regression model has no persistent ability to generate long-term profits. Even so, the cumulative investment return guided by these forecasting models is better than that of the risk-free T-bill account.

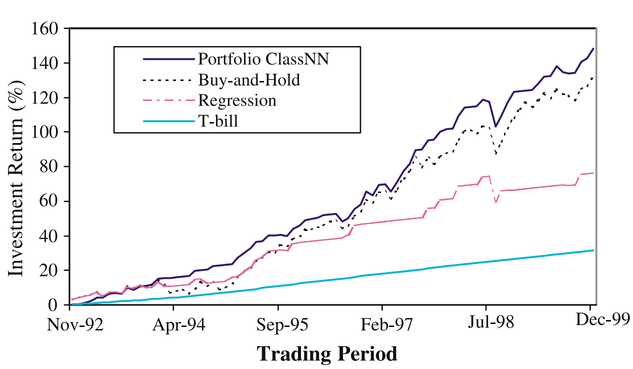

Fig. 5. A cumulative investment return (86 months) of the Portfolio Class NN, Regression, Buy-and-Hold, and T-bill models.

A Sharpe ratio analysis was also included in the study. It is simply the mean excess return of the trading divided by its standard deviation. The higher the Sharpe ratio, the higher the return and the lower the volatility. It is observed that all classification neural network models yield higher Sharpe ratio performance than that of the buy-and-hold account. In fact, all of the feed-forward neural network models developed in the study generate higher Sharpe ratio performance than that of the buy-and-hold account. More importantly, the Portfolio Class NN model, which yields a monthly return of 1.72%, is the best performer among the forecasting models evaluated in the study in terms of the Sharpe ratio performance. A paired two-sample test for mean return differences was also performed. The resulting *t*-statistics of the Original Level NN, Portfolio Level NN, GRNN, Sign Port Level NN, Original Class NN, Portfolio Class NN, and PNN models in comparison to the Regression model are 2.315, 2.327, 1.860, 2.509, 2.826, 3.248, and 1. 781, respectively (t-critical = 1.988 and $\alpha = 0.05$). The results indicate that all of the neural network models, excluding the GRNN and PNN models, significantly outperform the conventional linear regression forecast at a 95% level of confidence.

For comparability, the equivalent-variance portfolios (Equal-variance) using a combination of the return based on trading from each forecasting tool and the risk-free asset were created to evaluate the investment returns under the same volatility. Since the Regression model has the lowest standard deviation, the standard deviations of the other forecasting models were reduced to match that of the Regression model by combining them with the existing T-bill account. The results show that all of the neural network models, except the GRNN model, outperform the buy-and-hold account under this risk-adjusted return calculation. Remarkably, the monthly return based on trading guided by the Portfolio Class NN model is 0.27% greater than that of the buy-and-hold strategy under the same investment uncertainty. In other words, the difference of 0.27% is approximately equal to an annualized 3.24% of investment return under the identical risk exposure. Moreover, the averaged returns over each type of neural network forecasting models (classification versus level estimation under the same volatility) were also computed. The results show that the averaged return based on trading guided by classification models is 1.32%, compared to 1.19% for the level estimation models. This result supports the findings of Leung et al. (2000) that the classification models can generate higher profits than the level estimation models on the S&P 500 index trading.

7. Discussion and conclusions

An attempt has been made in this study to investigate the predictive power of financial and economic variables by adopting the variable relevance analysis technique in machine learning for data mining. This approach seems particularly attractive in selecting the variables when the usefulness of the data is unknown, especially when non-linearity exists in the financial market as found in this study. Since it has been long known that the determinant between the variables and their interrelationships over stock returns could change over time, different relevant input variables may be obtained by conducting this data mining technique under different time periods. In particular, we examined the effectiveness of the neural network models used for level estimation and classification. The results show that the trading strategies guided by the neural network classification models generate higher profits under the same risk exposure than those suggested by the other strategies, including the buy-and-hold strategy, as well as the level estimation forecasts of neural network and linear regression models.

More importantly, it is found that the highest profitability improvement guided by the Portfolio Class NN model is consistent with its three superior performance measures, namely the Pearson correlation, the root-mean squared error, and the correct sign of excess stock return. However, it can be observed from several forecasting models that better results of performance measurement do not always imply higher profitability. For instance, the GRNN model has better results of all three performance measures than the Original Level NN model does, yet it does not outperform the Original Level NN model in terms of the profits obtained from trading. This suggests that the forecast that has a higher percentage of correct sign may not necessarily yield higher profit. In fact, it may be due to the fact that the Original Level NN model gives better prediction of signs when the actual monthly stock return is highly volatile, thus receiving higher trading profits. This observation suggests the importance of making an accurate asset allocation (between stock and T-bill) when the positive or negative actual stock return of the next month is significant. Therefore, potentially higher investment return may be obtained from training the networks to correctly predict the signs of trading only when significant profit opportunities exist. The above empirical results show that the trading results based on several neural network forecasts can arrive at higher profitability improvement than the buy-and-hold strategies. However, this does not mean that the efficient market hypothesis can be totally ignored. The reason being that the buy-and-hold account can also be very profitable. In fact, the profitability obtained from the neural network forecasts will likely be less if transaction costs are taken into consideration.

Feed-forward neural network training is usually not very stable since the training process may depend on the choice of a random start. Training is also computationally expensive in terms of the training times used to determine the appropriate network structure. The degree of success, therefore, may fluctuate from one training pass to another. Although the portfolio neural networks yield impressive profits on average, it should raise concern that higher profits are derived at the expense of exposing the investors to higher risk. Nonetheless, the empirical findings in this study show that our proposed development of the portfolio network models using the n-fold cross-validation and early stopping techniques does not sacrifice any of the first period data used for training and validating the networks. This is especially useful when the data size is limited. In particular, we find that the method for improving the generalization ability of feed-forward neural networks, a combination of n-fold cross-validation and early stopping techniques, clearly help improve the out-of-sample forecasts. In addition to the early stopping advantage, improvement may be due to the fact that five-time network modeling allows the networks to extract more useful information from the data. Thus, the prediction based on the weighted excess return or the majority of excess return sign could effectively be used to reduce the prediction error. As a result, the portfolio network models for both classification and level estimation consistently outperform the linear regression, the generalized regression neural network, the probabilistic neural network, and the buy-and-hold account. More interestingly, the Sign Port Level NN model is able to generate a higher return than the models employing the level of excess stock returns. This strongly suggests that the portfolio neural networks that direct the trading based on the majority of the five network outputs can be developed and used as a more efficient forecasting tool.

In conclusion, both researchers and practitioners have studied stock market prediction for many years. Many studies conclude that stock returns can be predicted by some financial and economic variables. To this end, our finding suggests that financial forecasting is always and will remain difficult since such data are greatly influenced by economical, political, international, and even natural events. Obviously, this study covers only fundamental available information, while the technical analysis approach remains intact. It is far from perfect as the technical analysis has been proved to provide invaluable information during stock price and stock return forecasting and to some extent has been known to offer a relative mixture of human, political, and economical events. In fact, there are many studies done by both academics and practitioners in this area. If both technical and fundamental approaches are thoroughly examined and included during the variable relevance analysis modeling, it would no doubt be a major improvement in predicting stock returns. This study assumes trading strategies of investing in either the stock index portfolio or risk-free account in the absence of trading costs. During the simulated trading exercise, the authors also noticed that the profitability results could change if a different trading strategy was adopted by investors. In fact, it is possible that investors would benefit from further investigation on profits received from different trading decisions. Finally, future research should consider the trading simulation under the scenarios of stock dividends, transaction costs, and individual-tax brackets to replicate the realistic investment practices.

Appendix

SP Nominal Standard & Poor's 500 index at the close of the last trading day of each month. Source: Commodity Systems, Inc. (CSI)

DIV Nominal dividends per share for the S&P 500 portfolio paid during the month. Source: Annual dividend record/Standard and Poor's Corporation

T1 Annualized average of bid and ask yields on one-month T-bill rate on the last trading day of the month. It refers to the shortest maturity T-bills not less than one month in maturity. Source: CRSP tapes, the Fama risk free rate files

T1H Monthly holding period return on one-month T-bill rate on the last trading day of the month, calculated as *T1/12*

R Nominal stock returns on the S&P 500 portfolio, calculated as $R_t=(SP_t-SP_{t-1})/SP_{t-1}$

ER Excess stock returns on the S&P 500 portfolio, calculated as $ER_t=R_t-T1H_{t-1}$

DY Dividend yield on the S&P 500 portfolio, calculated as $DY_t=DIV_t/SP_t$

T3 3-month T-bill rate, secondary market, averages of business days, discount basis. Source: H.15 Release – Federal Reserve Board of Governors

T6 6-month T-bill rate, secondary market, averages of business days, discount basis. Source: H.15 Release – Federal Reserve Board of Governors

T12 1-year T-bill rate, secondary market, averages of business days, discount basis. Source: H.15 Release – Federal Reserve Board of Governors

T60 5-year T-bill constant maturity rate, secondary market, averages of business days. Source: H.15 Release – Federal Reserve Board of Governors

T120 10-year T-bill constant maturity rate, secondary market, averages of business days. Source: H.15 Release – Federal Reserve Board of Governors

CD1 1-month certificate of deposit rate, averages of business days. Source: H.15 Release - Federal Reserve Board of Governors

CD3 3-month certificate of deposit rate, averages of business days. Source: H.15 Release – Federal Reserve Board of Governors

CD6 6-month certificate of deposit rate, averages of business days. Source: H.15 Release – Federal Reserve Board of Governors

AAA Moody's seasoned Aaa corporate bond yield, averages of business days. Source: The Federal Reserve Bank of St. Louis

BAA Moody's seasoned Baa corporate bond yield, averages of business days. Source: The Federal Reserve Bank of St. Louis

PP Producer Price Index: Finished Goods. Source: U.S. Department of Labor, Bureau of Labor Statistics

IP Industrial Production Index: Market Groups and Industry Groups. Source: G.17 Statistical Release – Federal Reserve Statistical Release

CP Consumer Price Index: CPI for All Urban Consumers. Source: U.S. Department of Labor, Bureau of Labor Statistics

M1 M1 Money Stock. Source: H.6 Release – Federal Reserve Board of Governors

TE1 Term spread between *T120* and *T1*, calculated as $TE1=T120-T1$

TE2 Term spread between *T120* and *T3*, calculated as $TE2=T120-T3$

TE3 Term spread between *T120* and *T6*, calculated as $TE3=T120-T6$

TE4 Term spread between *T120* and *T12*, calculated as $TE4=T120-T12$

TE5 Term spread between *T3* and *T1*, calculated as $TE5=T3-T1$

TE6 Term spread between *T6* and *T1*, calculated as $TE6=T6-T1$

DE1 Default spread between *BAA* and *AAA*, calculated as $DE1=BAA-AAA$

DE2 Default spread between *BAA* and *T120*, calculated as $DE2=BAA-T120$

DE3 Default spread between *BAA* and *T12*, calculated as $DE3=BAA-T12$

DE4 Default spread between *BAA* and *T6*, calculated as $DE4=BAA-T6$

DE5 Default spread between *BAA* and *T3*, calculated as $DE5=BAA-T3$

DE6 Default spread between *BAA* and *T1*, calculated as $DE6=BAA-T1$

DE7 Default spread between *CD6* and *T6*, calculated as $DE7=CD6-T6$

References

Abhyankar, A., Copeland, L. S., & Wong, W. (1997). Uncovering nonlinear structure in real-time stock-market indexes: the S&P 500, the DAX, the Nikkei 225, and the FTSE-100. *Journal of Business & Economic Statistics*, *15*, 1–14.

Aggarwal, R., & Demaskey, A. (1997). Using derivatives in major currencies for cross-hedging currency risks in Asian emerging markets. *Journal of Future Markets*, *17*, 781–796.

Balvers, R. J., Cosimano, T. F., & McDonald, B. (1990). Predicting stock returns in an efficient market. *Journal of Finance*, *55*, 1109–1128.

Breen, W., Glosten, L. R., & Jagannathan, R. (1990). Predictable variations in stock index returns. *Journal of Finance*, *44*, 1177–1189.

Burrell, P. R., & Folarin, B. O. (1997). The impact of neural networks in finance. *Neural Computing & Applications*, *6*, 193–200.

Campbell, J. (1987). Stock returns and the term structure. *Journal of Financial Economics*, *18*, 373–399.

Chenoweth, T., & Obradovic, Z. (1996). A multi-component nonlinear prediction system for the S&P 500 Index. *Neurocomputing*, *10*, 275–290.

Demuth, H., & Beale, M. (1998). *Neural Network Toolbox: for use with MATLAB* (5th end). Natick: The Math Works, Inc..

Desai, V. S., & Bharati, R. (1998). The efficiency of neural networks in predicting returns on stock and bond indices. *Decision Sciences*, *29*, 405–425.

Elton, E. J., & Gruber, M. J. (1991). *Modern Portfolio Theory and Investment Analysis* (4th edn.). New York: John Wiley & Sons.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, *25*, 383–417.

Fama, E. F., & French, K. R. (1988). Dividend yields and expected stock returns. *Journal of Financial Economics*, *22*, 3–25.

Fama, E. F., & French, K. R. (1989). Business conditions and expected returns on stocks and bonds. *Journal of Financial Economics*, *25*, 23–49.

Fama, E. F., & Schwert, W. G. (1977). Asset returns and inflation. *Journal of Financial Economics*, *5*, 115–146.

Ferson, W. (1989). Changes in expected security returns, risk, and the level of interest rates. *Journal of Finance*, *44*, 1191–1217.

Gencay, R. (1998). Optimization of technical trading strategies and the profitability in securities markets. *Economics Letters*, *59*, 249–254.

Han, J., & Micheline, K. (2000). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers.

Hill, T., O'Connor, M., & Remus, W. (1996). Neural network models for time series forecasts. *Management Science*, *42*, 1082–1092.

Jensen, M. (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics*, *6*, 95–101.

Keim, D., & Stambaugh, R. (1986). Predicting returns in the stock and bond markets. *Journal of Financial Economics*, *17*, 357–390.

Leitch, G., & Tanner, J. E. (1991). Economic forecast evaluation: profits versus the conventional error measures. *American Economic Review*, *81*, 580–590.

Leung, M. T., Daouk, H., & Chen, A. S. (2000). Forecasting stock indices: a comparison of classification and level estimation models. *International Journal of Forecasting*, *16*, 173–190.

Lo, A. W., & MacKinlay, A. C. (1988). Stock market prices do not follow random walks: evidence from a simple specification test. *Review of Financial Studies*, *1*, 41–66.

Maberly, E. D. (1986). The informational content of the interday price change with respect to stock index futures. *Journal of Futures Markets*, *6*, 385–395.

Malliaris, M., & Salchenberger, L. (1993). A neural network model for estimating option prices. *Journal of Applied Intelligence*, *3*, 193–206.

Mills, T. C. (1990). Non-linear time series models in economics. *Journal of Economic Surveys*, *5*, 215–241.

Motiwalla, L., & Wahab, M. (2000). Predictable variation and profitable trading of US equities: a trading simulation using neural networks. *Computer & Operations Research*, *27*, 1111–1129.

Nelson, M., Hill, T., Remus, W., & O'Connor, M. (1999). Time series forecasting using neural networks: should the data be deseasonalized first? *Journal of Forecasting*, *18*, 359–367.

Pantazopoulos, K. N., Tsoukalas, L. H., Bourbakis, N. G., Brun, M. J., & Houstis, E. N. (1998). Financial prediction and trading strategies using neurofuzzy approaches. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, *28*, 520–530.

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

Pesaran, M. H., & Timmermann, A. (1992). A simple nonparametric test of predictive performance. *Journal of Business & Economic Statistics*, *10*, 461–465.

Pesaran, M. H., & Timmermann, A. (1995). Predictability of stock returns: robustness and economic significance. *Journal of Finance*, *50*, 1201–1227.

Peterson, G. E., Aylward, S. R., Bond, W. E., & St Clair, D. C. (1995). Using Taguchi's method of experimental design to control errors in layered perceptrons. *IEEE Transactions on Neural Networks*, *6*, 949–961.

Poddig, T., & Rehkugler, H. (1996). A world of integrated financial markets using artificial neural networks. *Neurocomputting*, *10*, 251–273.

Priestley, M. B. (1988). *Non-linear and Non-stationary Time Series Analysis*. London: Academic Press.

Qi, M., & Maddala, G. S. (1999). Economic factors and the stock market: a new perspective. *Journal of Forecasting*, *18*, 151–166.

Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. San Francisco: Morgan Kaufmann Publishers.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (Vol. 1) (pp. 318–362). Cambridge: The MIT Press, 318–362.

Schwert, W. (1990). Stock returns and real activity: a century of evidence. *Journal of Finance*, *45*, 1237–1257.

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, *3*, 109–118.

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

Swales, G. S., & Yoon, Y. (1992). Applying artificial neural networks to investment analysis. *Financial Analysts Journal*, *48*, 78–80.

Vellido, A., Lisboa, P. J. G., & Vaughan, J. (1999). Neural networks in business: a survey of application (1992-1998). *Expert Systems with Applications*, *17*, 51–70.

Wasserman, P. D. (1993). *Advanced Methods in Neural Computing*. New York, NY: Van Nostrand Reinhold.

Wood, D., & Dasgupta, B. (1996). Classifying trend movements in the MSCI U.S.A. capital market index - a comparison of regression, ARIMA, and neural network methods. *Computers & Operations Research*, *23*, 611–622.

Wu, Y., & Zhang, H. (1997). Forward premiums as unbiased predictors of future currency depreciation: a non-parametric analysis. *Journal of International Money and Finance*, *16*, 609–623.