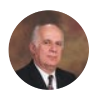
Neural Networks as a Decision Maker for Stock Trading: A Technical Analysis Approach

Article *in* International Journal of Smart Engineering System Design · October 2003

DOI: 10.1080/10255810390245627

David Enke
Missouri University of Science and Technology



C.H. Dagli
Missouri University of Science and Technology

International Journal of Smart Engineering System Design, 5:313–325, 2003

ISSN: 1025-5818 print/1607-8500 online
DOI: 10.1080/10255810390245627

Neural Networks as a Decision Maker for Stock Trading: A Technical Analysis Approach

Suraphan Thawornwong
David Enke
Cihan Dagli

Intelligent Systems Center and Smart Engineering Systems Laboratory, University of Missouri - Rolla, Rolla, Missouri, USA

There has been a growing interest in applying neural networks and technical analysis indicators for predicting future stock behavior. However, previous studies have not practically evaluated the predictive power of technical indicators by employing neural networks as a decision maker to uncover the underlying nonlinear pattern of these indicators. The objective of this paper is to investigate if using these indicators as the input variables to a neural network will provide more accurate stock trend predictions, and whether they will yield higher trading profits than the traditional technical indicators. Three neural networks are examined in the study to predict the short-term trend signals of three stocks across different market industries. The overall results indicate that the proportion of correct predictions and the profitability of stock trading guided by these neural networks are higher than those guided by their benchmarks.

Keywords: neural networks, technical analysis, technical indicators, stock prediction, stock trading, trend signal

Stock trend or stock price prediction is an important financial subject that has attracted researchers' attention for many years. This is due to the fact that a successful prediction model could result in substantial monetary rewards. However, predicting stock price or stock return is not a simple task, since many market factors are involved and their structural relationships are too complex to be clearly extracted. Technical analysis has a long tradition in forecasting movements in a financial time series (Plummer 1989); however, it also has a long history of being criticized by academics and practitioners (Malkiel 1995).

This criticism can be explained by the fact that technical analysis is built on weak foundations. For example, the expectation that some historical patterns of a stock price will be repeated in the future may not necessarily be fulfilled, since market conditions change over time (Wong and Ng 1994) and there is no explanation as to why these patterns should be expected to repeat (Jegadeesh 2000). As noted in the efficient market hypothesis (Fama 1970), it is impossible to forecast future price, since the price already reflects everything that is currently known about the stock.

Nonetheless, in recent years technical analysis has been widely accepted as one of many viable analytical options among both financial economists and brokerage firms (Achelis 1995). In fact, large investments are rarely made without touching this technical environment. This is due to the fact that many researchers are open to the idea that markets may not be fully efficient and prices may be affected by human sentiments found in psychology rather than economics (Barberis et al. 1998).

Technical analysis also appears to be a compromising tool, since it offers a relative mixture of human, political, and economical events. Theoretically, it attempts to predict the trend of stock prices by using data surrounding past prices and volumes. The main problem with this technique is that it relies heavily on the discovery of strong empirical regularities in observations of the price and volume movements (Liu and Lee 1997). In other words, the supporter of this technique is only concerned with the identification of major turning points for assessing the securities' movement. In reality, these regularities are not always evident, often masked by noise, and vary from security to security, making it difficult for investors who use this technique to consistently and accurately determine future prices.

In addition to the increased use of technical analysis, investors today are becoming more dependent on advanced computer algorithms and technologies to benefit from a wider range of investment choices (Elton and Gruber 1991). Artificial neural networks are one of the technologies that have caused the most excitement in this financial environment. They provide an

Received 7 October 2002; accepted 8 January 2003.

Address correspondence to David Enke, University of Missouri-Rolla, 204 Engineering Management Building, Rolla, MO, 65409-0370, USA. E-mail: enke@umr.edu

interesting technique that can theoretically approximate any nonlinear continuous function on a compact domain to any designed degree of accuracy (Cybenko 1989). The novelty of neural networks lies in the ability to model nonlinear processes without a priori assumption about the nature of the generating process (Hagen et al. 1996). This is 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), and should prove valuable in an area such as technical analysis. Neural networks also offer the flexibility of numerous architecture types, learning algorithms, and validation procedures.

This paper employs the above-mentioned advantages of neural networks to enhance the effectiveness of the technical analysis indicators in predicting stock trend signals. Several popular technical indicators are selected as the inputs variables used to train the neural networks. The objective is to explore if the neural networks can be used as decision makers to uncover the regularities of the underlying price and volume movements. Three neural network models, including the feed-forward, probabilistic, and learning vector quantization neural networks are examined for their ability to provide an effective forecast of the future stock trend signals.

Three major stocks across different market industries are selected to test the robustness of the three resulting neural network models. Trading strategies are also developed to account for the best opportunity to take a position while using the neural networks for stock trend predictions. Finally, the predictability (correct predicted trend) and trading profitability (risk and return) directed by these models are compared against those guided by a buy-and-hold strategy and other individual technical analysis indicators.

THE TECHNICAL ANALYSIS APPROACH

Two common techniques are used to forecast future stock behavior: technical analysis and fundamental analysis. Technical analysis studies the historical data surrounding price and volume movements of the stock by using charts as a primary tool to forecast future price movements (Murphy 1999). Investors base their analysis on the premise that some historical patterns of stock prices are assumed to repeat in the future, and, thus, these patterns can be used for predictive purposes. The motivation behind technical analysis is its ability to identify changes in trends at an early stage, and to maintain an investment strategy until the weight of the evidence indicates that the trend has reversed.

On the other hand, fundamental analysis concentrates on the economic forces of supply and demand that cause prices to move higher, lower, or stay the same (Schwager 1995). The relevant factors (e.g., company, industry, and economic specific) affecting stock prices are examined in order to determine a security's intrinsic value. When attempting to determine the direction of likely price changes, the two techniques approach the objective of predicting stock movement from different directions. Fundamental analysis studies the cause of market movement, while technical analysis studies the effect.

During the last few years, a considerable number of studies have been done attempting to address the ability of neural networks to predict future stock movements by using fundamental factors; for example, see the articles by Desai and Bharati (1998), Dropsy (1996), Leung et al. (2000), Motiwalla and Wahab (2000), Poddig and Rehkugler (1996), Qi (1999), and Qi and Maddala (1999). This increasing interest is due to the fact that nonlinearity has been greatly emphasized by various researchers and financial analysts, and neural networks are capable of performing nonlinear modeling without a priori knowledge about the relationships among these factors.

Beyond using purely fundamental factors, several studies have also included a few technical analysis indicators in their experiments; see Austin et al. (1997), Chenoweth and Obradovic (1996), Kuo (1998), Longo and Long (1997), and Quah and Srinivasan (1999). In addition, historical time series of stock returns and index values have been examined by a number of studies as an alternative for the technical analysis approach; for example, see Brown et al. (1998), Chandra and Reeb (1999), Cogger et al. (1997), Darrat and Zhong (2000), Fernandez et al. (2000), Saad et al. (1998), and Zemke (1999). Unfortunately, only a few studies have addressed the predictability of neural networks to predict stock price movements using the well-known theories of technical analysis.

Gencay and Stengos (1998) combined the use of two simple trading rules, namely the moving averages and trading range breaks, with a feed-forward neural network to predict daily returns of the Dow Jones Industrial Average (DJIA) index. The buy and sell signals generated from the trading rules were used as inputs in the forecasting models. Kim and Han (2000) transformed continuous values of several indicators, such as the Relative Strength Index, Stochastic Oscillator, William's %R, Commodity Channel, and Momentum into discrete values in accordance with certain thresholds. The directions of change of the Korea stock price index were the predictions given by a trained feed-forward neural network.

One experiment done by Lam and Lam (2000) applied the differences of the twenty-day moving averages and stock prices as the inputs to train a feed-forward neural network for generating the next closing price of the Hang Seng Index Futures Contract in Hong Kong. Tsaih et al. (1998) relied on a rule-base of the Stochastic Oscillator and Relative Strength Index indicators, as well as a recurrent neural network that used the Relative Strength Index indicator as one of the inputs for predicting the directions of daily price change in the S&P 500 futures contracts. A voting mechanism was then adopted to generate the trading recommendation.

The work done by Chan and Teong (1995) provides perhaps the most comprehensive comparison of the results obtained from the feed-forward neural networks and the technical indicators found in the literature. Deutchemark (DEM/USD) daily data was used for the study. Only inputs used to calculate regular technical indicators (i.e., highs, lows, and closes of DEM/USD) were employed to predict future highs, lows, and closes. These three predicted values were subsequently used to manually compute the predicted technical indicator. Three popular technical indicators chosen for the study were the Moving Average, Stochastic Oscillator, and Momentum indicators. The trading results showed that using the neural network to take advantage of getting in and out of a trade, before the regular technical indicator signals the stock trend, seems to be the key to profitable trades in the study.

Although most of the above-mentioned studies provide valuable findings, they each rely on either a few popular technical indicators, or a single neural network type. More importantly, the technical indicators have not been tested with neural networks in such a way that would have best utilized their potential. Furthermore, the neural network ability was handicapped in many studies due to the use of just a few technical indicator inputs.

INDICATOR AND VARIABLE SELECTION

Many of the well-founded theories of technical analysis that assume linear relationships are likely to only capture the strong regularities of price and volume movements. However, human elements and political/economical occasions have not been known to conform to such simple regularities. In particular, many technical indicators have been found to give conflicting signals of stock movement (Vamsi et al. 2001). Thus, inexperienced investors using technical analysis often get confused and hesitate to make investment decisions.

This can be explained by the fact that the rationales behind a signal recommendation were practically derived from various technical analysts. These rationales are likely to diverge depending on the analysts: experience, criteria, preference, and empirical observations, which also tend to diverge from security to security. Therefore, it should be possible to employ neural networks to manage these unclear regularities and diversities, while also using the neural networks as a framework that automatically helps investors make a decisive investment decision when recommendations from conventional technical indicators contradict each other.

It has long been known that neural network training can be made more efficient if certain preprocessing steps have been performed on the network inputs (Han and Micheline 2000). This approach is particularly applicable to certain aspects of technical indicators. Though neural networks can independently learn any function, neural network performance could be further improved if stock related values, such as highs, lows, and volumes, are preprocessed into more meaningful information, which are indeed the technical indicators. This would mimic how most technical analysis users do not simply rely on the original values of opens, highs, lows, closes, and volumes to determine the signals of future stock movement. Undoubtedly, using technical indicators would allow the neural networks to concentrate on the important details necessary for an accurate prediction of future stock movement underlying the nonlinear relationship.

The present study focuses on short-term stock prediction. Therefore, daily data, including opens, highs, lows, closes, and volumes were collected to perform the technical indicator calculations for a total of 846 trading days (August 22, 1996 to December 29, 1999). The resulting daily indicator values were then divided into two periods. The first period included the indicator values from August 22, 1996 to June 30, 1999, for a total of 720 trading days, while the second period contained the indicator values from July 1, 1999 to December 29, 1999, for a total of 126 trading days. The former period was used for training and validating the neural networks. The latter period was reserved for out-of-sample evaluation and comparison of performances among the neural networks and regular technical indicators. All data was adjusted for applicable stock splits and dividend distributions. Three major stocks across different market industries were also selected and studied to support the robustness of the three neural network models. These stocks include Lockheed Martin Corp. (LMT), representing aerospace and defense, Caterpillar, Inc. (CAT), representing construction and agricultural machinery, and Delta Air Lines, Inc. (DAL), representing the airline group.

Five technical indicators, including the relative strength index, money flow index, moving average, stochastic oscillator, and moving average convergence/divergence were selected in this study. Since different technical analysts often use different criteria to capture the buy and sell signals, based on their accumulated experience, only the most commonly used criteria were considered in this paper as a default to remove conflicting recommendations. The following provides a brief description of the technical trading criteria (Achelis 1995; Murphy 1999; Schwager 1996) subsequently used to perform the technical indicator calculations and stock trading exercises in this study.

Relative Strength Index (RSI)

The RSI compares the relative strength of price gains on days that close above the previous day's close to price losses on days that close below the previous day's close of a single security (not the relationship to another security, index, or sector). The formula for the RSI is:

$$RS = \frac{Average\ of\ 14\ days'\ up\ closes}{Average\ of\ 14\ days'\ down\ closes}$$
$$RSI = 100 - \frac{100}{1+RS}. \tag{1}$$

The RSI is plotted on a vertical scale of 0 to 100. Movements above 70 are considered overbought, while an oversold condition occurs when a move is under 30. The first move to the overbought or oversold region is a warning. Failure swings above 70 or below 30 on the RSI are strong indications of market reversals. In other words, bullish divergence between a lower bottom in prices and a higher bottom in the RSI sets up a potential buy signal, and bearish divergence between a higher top in prices and a lower top in the RSI sets up a potential sell signal.

Money Flow Index (MFI)

The MFI measures the strength of money flowing in and out of a security. It is related to the RSI, but where the RSI only considers prices, the MFI accounts for volume. The calculations are as follows:

$$\begin{aligned} \mathit{Typical\ Price} &= \frac{\mathit{High} + \mathit{Low} + \mathit{Close}}{3} \\ \mathit{Money\ Flow} &= \mathit{Typical\ Price} \times \mathit{Volume} \\ \mathit{Money\ Ratio} &= \frac{\mathit{Positive\ Money\ Flow}}{\mathit{Negative\ Money\ Flow}} \\ \mathit{MFI} &= 100 - \frac{100}{1 + \mathit{Money\ Ratio}}. \end{aligned} \tag{2}$$

If the current day's closing price is greater than the previous day's closing price, it is considered positive money flow. If the current day's closing price is less, it is considered negative money flow. In this study, positive money flow and negative money flow are the sums of the positive money and negative money over fourteen days, respectively. Warning signs for overbought or oversold are considered when the MFI is above 80 or below 20. Failure swings above 80 or below 20 on the MFI are strong indications of market reversals.

Moving Average (MA)

The MA is an indicator that shows the average value of a stock price over a period of time. As the stock price changes, its average price moves up or down. The formula for the MA is:

$$MA = \frac{\sum_{1}^{14} \mathit{Closing\ Price}}{14}. \tag{3}$$

The most popular method of interpreting the MA is to compare the relationship between the MA with the stock price itself. Therefore, a buy signal is generated when the stock price rises above its moving average, and a sell signal is generated when the stock price falls below its moving average.

Stochastic Oscillator (SO)

The SO is used to indicate an overbought or oversold condition and is displayed as two lines. The main line is called "%K." The other line, called "%D," is a three-day moving average of %K. The formula for %K is:

$$\%K = \frac{C - LX}{HX - LX} \times 100, \tag{4}$$

where C is the latest close, LX is the lowest low for the last fourteen days, and HX is the highest high for the same fourteen days.

These two lines oscillate between a vertical scale ranging from 0 to 100. The %K line is a faster line, while the %D line is a slower line. A sell signal occurs when the faster %K line crosses below the slower %D line from above the 80 level. The %K line crossing above the %D line from below the 20 level is a buy signal.

Moving Average Convergence/Divergence (MACD)

The MACD is a difference between an $\alpha = 0.075$ and an $\alpha = 0.15$ exponential moving average (EMA). An $\alpha = 0.20$ exponential moving average of the MACD (the signal line) is then plotted on top of the MACD to show buy and sell opportunities. The formulas are:

$$\begin{aligned} EMA &= [\alpha \times \mathit{Today's\ Close}] \\ &\quad + [(1-\alpha) \times \mathit{Yesterday's\ EMA}] \\ MACD &= [0.075\ \mathit{EMA\ of\ Closing\ Prices}] \\ &\quad - [0.15\ \mathit{EMA\ of\ Closing\ Prices}] \\ \mathit{Signal\ Line} &= 0.20\ \mathit{EMA\ of\ MACD}. \end{aligned} \tag{5}$$

The basic MACD trading rule is to sell when the MACD falls below its signal line, and to buy when the MACD rises above its signal line.

For each of the five technical indicators, trading criteria are made based on the discovery of particular patterns identified by recent movements of the technical indicators. According to the five technical trading criteria, there are a total of eight variables, including the Closing Price (CP), RSI, MFI, MA, %K, %D, MACD, and Signal Line (SL) required to capture the underlying buy and sell signals of the observed stock. More importantly, recent historical data of indicator and stock price movements are also necessary to make the stock trend prediction. As previously mentioned, neural network modeling can be more effective if these variables are preprocessed to replicate the patterns of stock and technical indicator movements.

In this study, the difference ($CP_{t-1} - CP_{t-2}$) of stock closing price was employed so that it could be compared in terms of change with the daily stock price movements. This is due to the fact that the change of variables in forecasting financial time series may be more meaningful to the network models than those using the original values (Pantazopoulos et al. 1998). Similarly, the MA, SO, and MACD trading guidelines require that line crossing (below or above) must occur before the buy or sell signal can be initiated.

Accordingly, the differences of these lines were calculated so that they could be measured in terms of how close the lines will finally cross each other. The crossing can explicitly be captured by sign changes (positive and negative) of the resulting values of the line differences. The differences for MA, SO, and MACD indicators were defined as $MA_{t-1}-CP_{t-1}$, $\%K_{t-1}-\%D_{t-1}$, and $MACD_{t-1}-SL_{t-1}$, respectively. Note that no preprocessing steps were done for the RSI_{t-1} and MFI_{t-1} indicators since the indications of market reversals can be identified by comparing the change of stock price movements ($CP_{t-1}-CP_{t-2}$) with these two indicators.

Finally, the six selected inputs were included in the base sets with three-day time lags to account for recent movements of the technical indicators. As a result, eighteen input variables, including $CP_{t-1}-CP_{t-2}$, $CP_{t-2}-CP_{t-3}$, $CP_{t-3}-CP_{t-4}$, RSI_{t-1}, RSI_{t-2}, RSI_{t-3}, MFI_{t-1}, MFI_{t-2}, MFI_{t-3}, $MA_{t-1}-CP_{t-1}$, $MA_{t-2}-CP_{t-2}$, $MA_{t-3}-CP_{t-3}$, $\%K_{t-1}-\%D_{t-1}$, $\%K_{t-2}-\%D_{t-2}$, $\%K_{t-3}-\%D_{t-3}$, $MACD_{t-1}-SL_{t-1}$, $MACD_{t-2}-SL_{t-2}$, and $MACD_{t-3}-SL_{t-3}$, were employed to predict the direction of the next day's stock price (y_t). These time lags were used throughout the experiment to maintain realistic situations when the technical indicators and stock prices are calculated and gathered. Constructing the data in this manner ensures that only observable (not future data) were employed as inputs to the neural network models.

TRADING PRACTICES AND STRATEGIES

Investors today are allowed to profit from both an increase and decline in stock prices. As is commonly known, investors purchasing a share of stock to open a long position profit from price increases. Alternatively, there is another trading practice called a short sale which allows investors to profit from the decline in stock prices. Initially, an investor borrows a share of stock from a broker and sells it. Later, the short seller must purchase a share of the same stock in the market in order to replace the share that was borrowed. This is called covering the short position. The short seller anticipates that the stock price will fall, expecting that the share can be purchased at a lower price than it was initially sold for. Therefore, profits on the short position increase when stock prices fall, while price increases result in profits when the long position is held.

In this paper, a 1% round-trip transaction cost (sell a share to close a long position, and sell a borrowed share to create a short position; or purchase a share to cover a short position and purchase a share to open a long position) was charged to the investor when an asset allocation of security positions was made. This was done to account for more realistic trading results, since profitability could be washed out by excessive trading, especially when the security positions were assessed daily. In an effort to examine if the neural network models could practically have been used to generate higher profits, a trading strategy was devised as trading criteria in connection with the predicted signals of neural network forecasts. This trading strategy will not only assist investors in making their trading decisions, but will also allow them to perform systematic trading as compared against the technical indicators.

The trading assumes that an investor either maintains a current position, or makes an asset allocation of whether to shift from long to short or from short to long when the stock market is opened for each trading day. Further, it is assumed that the money invested in either a current day long or short position becomes non-liquid and remains detained in that position until the next trading day is opened. After the market is closed for each trading day, the investor has to decide whether to maintain the current position, or make an asset allocation, depending on the signal generated by the neural network forecasts calling for a long or short position in the next trading day as compared to the current day. Note that in addition to the transaction cost scenarios, the stock valuation was also adjusted for all applicable stock splits and dividends to replicate realistic investment. However, uptick exchange rules, taxes, and margin trading were ignored for this study. The following describes the trading criteria with respect to the predicted signals (directions) of the neural network forecasts:

If $\hat{y}_t = +1$, then
  Maintain the current long position, or cover a short position and open a long position (pay the 1% transaction cost and receive the profit/loss of the covered short position)
Else (if $\hat{y}_t = -1$), then
  Maintain the current short position, or close a long position and create a short position (pay the 1% transaction cost and receive the profit/loss of the closed long position),

where $\hat{y}_t$ is the predicted direction of the next day stock prices given by the neural network forecasting models. The $+1$ and -1 represent the predicted upward and downward directions of the next day's stock price movement, respectively. Note that a one-way transaction cost representing 0.5% of the transaction was used for the first and last trades, since only one trade is required to enter and exit the market.

Trading simulation was conducted to compare the profitability received from the trading practices guided by the neural network models to that from the practices suggested by the well-known technical indicators. To perform the simulated trading exercise, an investor is assumed to buy or short one share of stock on the first recommended buy or short signal, respectively. Subsequently, the investor will be in the market for the whole trading period, either staying in a long or short position, depending on the recommended trading signal. Finally, the investor will either sell or cover the share of the stock when the market is closed on the final trading day, regardless of the last signal received.

It should also be noted that the trading strategies and practices of the technical indicators employed in this study are similar to those of the neural network forecasts for comparable performance evaluations. As such, a sell signal identified by the technical indicators will be used to create a short position at the beginning of the next trading day, while the long position will be created for the next trading day when a buy signal of the technical indicators is generated. Similarly, the current investment position will be maintained if the technical indicators signal neither a sell nor buy recommendation. Additionally, if the technical indicators indicate a trading recommendation to take the similar investment action as the one that is currently held, no trading exercise will be taken for the next trading day.

To measure the risk and return of stock trading driven by the neural network and technical indicator forecasting models, the average daily return (Return) and risk (σ) of trading based on the forecasts of each model, and the resulting Sharpe ratio (SR) for the whole trading period were computed. Steps of calculation conducted in this study are described as follows:

$$\mathrm{Return} = \frac{1}{\mathrm{N}}\sum_{\mathrm{t}=1}^{\mathrm{N}} \frac{[(\hat{\mathrm{y}}_{\mathrm{t}} \times \mathrm{P}_{\mathrm{t}}) - \mathrm{TC}_{\mathrm{t}}]}{\mathrm{TOT}}$$

$$\hat{\mathrm{y}}_{\mathrm{t}} = \begin{cases} +1 & \text{if a long signal is suggested} \\ -1 & \text{if a short signal is suggested} \\ 0 & \text{otherwise (before the first occurrence of a signal)} \end{cases} \tag{6}$$

$$\mathrm{P}_{\mathrm{t}} = \text{Opening Price}_{\mathrm{t}+1} - \text{Opening Price}_{\mathrm{t}},$$

where Return is the average daily return based on trading guided by each neural network or technical indicator forecasting model, N is the number of trading days (126 in this study), TC is the transaction cost at time t if an asset allocation is made, TOT is the total investment (initial invested capital plus transaction costs) required to perform stock trading over the whole trading period, $\hat{y}_t$ is the recommended position based on the aforementioned trading strategies guided by the forecasting model, and P_t is the realized value of profits or losses with respect to one trading day. Equation 7 is used for calculating the level of risk, while Eq. 8 is used to calculate the Sharpe ratio.

$$\sigma^2 = \frac{1}{N}\sum_{t=1}^{N}\left[\frac{[(\hat{y}_t \times P_t) - TC_t]}{TOT} - \mathrm{Return}\right]^2 \qquad \sigma = \sqrt{\sigma^2} \tag{7}$$

$$SR = \frac{\mathrm{Return}}{\sigma}, \tag{8}$$

where σ is the risk of trading based on the recommended positions of the forecasting model, and SR is the Sharpe ratio, which gives the return earned by taking one unit risk of trading on the recommendations of the forecasting model. The higher the SR, the higher the return and the lower the risk. For simplicity, the risk-free rate is ignored in this study.

NEURAL NETWORK MODELING

The theory of neural network computation provides interesting techniques that mimic the human brain and nervous system. Neural networks are characterized by the pattern of connections between 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 types of neural networks often used for classification problems were implemented in this study to compare their predictive ability against the classical technical analysis indicators. The following three subsections give a brief introduction to these three neural network models.

Feed-Forward Neural Network (FNN)

The FNN has been widely used for financial forecasting due to its ability to correctly classify and predict the dependent variable (Vellido et al. 1999). For each training sample, the input variables are fed simultaneously into a layer of neuron 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 inputted to another hidden layer, and so on. The outputs of the last hidden layer are inputted to output layer units which issue 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 FNN. It is a method for assigning responsibility for mismatches to each of the processing elements in the network, which is achieved by propagating the gradient of the activation function back through the network to each hidden layer, all the way down to the first hidden layer. The weights and biases are then modified so as to minimize the mean squared error between the network's prediction and the actual target. Since the FNNs 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 an explanation of the backpropagation algorithm used to train the FNNs.

The determination of network parameters is a user-defined process. In most cases, it follows a heuristic approach, where several networks with different learning rates, momentums, numbers of hidden neurons, and activation functions are trained, with selection of the best performing network (Tan and Wittig 1993). In this study, a sigmoid hyperbolic tangent function was selected as the activation function to generate an even distribution over the input values. The resilient backpropagation-training algorithm was employed to train the feed-forward neural networks, because this optimization method is generally much faster than the standard steepest descent algorithm, and it also requires only a modest increase in memory requirements. 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 FNN was built with three layers, including the input layer, hidden layer, and output layer. Each of the eighteen input variables was assigned a separate input neuron to the input layer. Since there are two classes, two output neurons were employed for the output layer to represent different classes of the predicted direction. The vectors $[+1 \ -1]$ and $[-1 \ +1]$ represent the predicted upward and downward directions of the next day stock trend, respectively. The configuration of the FNN used in this study is given in Figure 1. The output neuron with the highest value was taken to represent the predicted direction based on a given set of input variables.

An early stopping technique was also employed to achieve better generalization by eliminating the over-fitting problems that usually occur during network training in trained neural networks (Demuth and Beale 1998). When the network begins to over-fit the data, the error on the validation cases will typically begin to rise. The training was then stopped, causing a return of the weights and biases to the minimum of the validation error. In this study, the 576 trading days (80%) of the first period were randomly selected as a training set for determining the network specifications. The validation set, for a total of 144 trading days (the remaining 20% of the first period), was consequently used to decide when training should be stopped. Note that 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 increase the effectiveness of the learning algorithm. The connection weights and biases were initially randomized and then determined during the backpropagation training process. The number of hidden neurons and appropriate learning rate that minimize the classification errors of the validation set were also determined during neural network training.

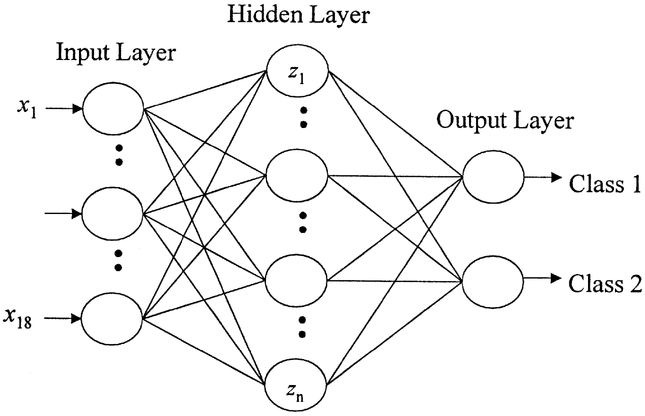


FIG. 1 *A three-layer feed-forward neural network for classification.*

Probabilistic Neural Network (PNN)

The PNN is useful for classification problems, since it finds decision boundaries between categories of input patterns. It is based on the estimation of probability density functions for various classes learned from training samples. Interestingly, 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). Generally speaking, the PNN is a parallel implementation of a standard Bayesian classifier and has two hidden layers that can perform pattern classification. The PNN formula can be briefly explained as follows:

$$f_A(x) = \frac{1}{(2\pi)^{P/2}\sigma^P n}\sum_{i=1}^{n} z_i, \tag{9}$$

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.

Figure 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. 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

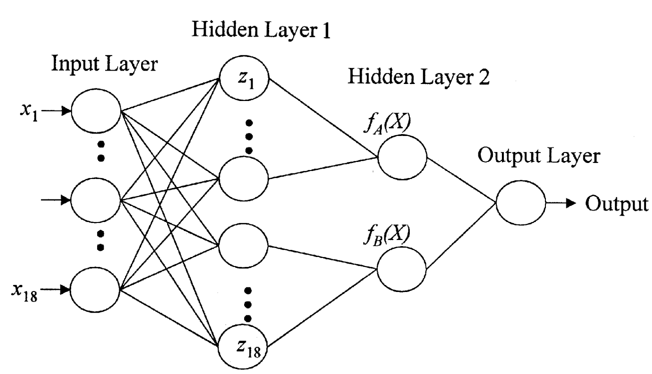


FIG. 2 *Probabilistic neural network architecture.*

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

The design of the PNN is fast and straightforward. In fact, neither validation nor early stopping is required during its design. Therefore, there is no need to randomly partition the data into training and validation sets for determining the network specifications. To take this unique advantage, the first period (August 22, 1996 to June 30, 1999) of the data set was used in network modeling. The $+1$ and $+2$ signals were used as the network outputs to represent the predicted downward and upward directions, respectively. Also, a smoothing parameter equal to 1.00 was selected to consider several nearby design vectors. Again, the PNN design employed the same preprocessing techniques as those implemented for the FNN.

Learning Vector Quantization Neural Network (LVQ)

The LVQ is a two-layer network that can classify input vectors into target classes. The first hidden layer is normally called a competitive layer, and the second hidden layer is known as a linear layer (Kohonen 1995). The architecture of the LVQ developed in this study is presented in Figure 3. In the competitive network, the neuron with the nonzero output indicates which class contains the input vector. For the LVQ's competitive layer, the winning neuron indicates a subclass, rather than a class. Therefore, there may be several different neurons (subclasses) that can make up each class.

Subsequently, the linear layer is used to combine subclasses into a single target class. The process of combining subclasses to form the class allows the LVQ to create complex class boundaries. The only requirement is that the competitive layer must have enough neurons and each class must be assigned an appropriate amount of competitive neurons.

There currently exist several versions of the LVQ learning algorithm, each containing slightly different properties (Kohonen 1995). The algorithm selected for this study was the LVQ2, which is efficient for tuning

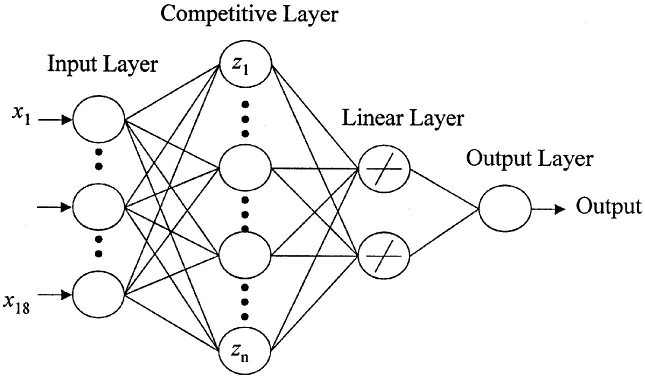


FIG. 3 *Learning vector quantization neural network architecture.*

the decision boundaries between the competing classes. The network outputs were similar to those used during the PNN design. The class percentages of 0.5 upward and 0.5 downward were selected to represent an equivalent distribution of future stock movements. The LVQ employed the same training set, validation set, and preprocessing technique as those used for the FNN modeling. Similarly, the training set was used to determine the appropriate learning rate and number of hidden neurons that minimize the classification errors of the validation set.

EMPIRICAL RESULTS

The predictive performance of the developed neural networks and technical analysis indicators was evaluated using the untouched out-of-sample data (second period). One possible approach for evaluating the forecasting performance is to investigate whether traditional error measures, such as those based on the root-mean squared error or correlation between the actual out-of-sample targets and their predicted values, are small or highly correlate, respectively. However, there is some evidence in the finance literature suggesting that traditional measures of forecasting performance may not be strongly related to profits obtained from trading (Pesaran and Timmermann 1995).

An alternative approach is to look at the proportion of time that the signals of future stock price changes (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, the SIGN presented in Table 1 was selected as the performance measure to report the predictability results of the neural network models and technical analysis indicators.

The predictability results obtained from always investing in each stock (buy-and-hold) are also provided as the benchmark for performance comparisons in this study. To explore the predictability results further, the outputs of the three neural network models (FNN, PNN, and LVQ) were combined to form a portfolio network model (PortNN). As such, the decisive predicted direction of stock price movement of the PortNN was derived from the majority of the three combined portfolio network outputs. For instance, when the three network models generate two positive predicted directions and one negative predicted direction of stock price movement based on a given set of the eighteen input variables, the decisive prediction is resolved to be an upward direction. The average of SIGN across all three stocks for each developed model provided in the last column of Table 1 is also calculated. Note that the MFI failed to identify the strong indications of market reversals hidden in the CAT's and DAL's securities, resulting in two inoperative SIGN calculations.

According to Table 1, the results show that both the technical indicators and neural networks successfully

TABLE 1 Predictability Results

		SIGN			
		LMT	CAT	DAL	Average
Technical Indicators	RSI	0.4880	0.4828	0.4800	0.4836
	MFI	0.4884	–	–	–
	MA	0.5439	0.4435	0.4530	0.4801
	SO	0.4960	0.5050	0.5242	0.5084
	MACD	0.4727	0.5210	0.6168	0.5368
Neural Networks	FNN	0.5476	0.6429	0.5794	0.5900
	PNN	0.5873	0.5476	0.5317	0.5555
	LVQ	0.5556	0.5476	0.5794	0.5609
	PortNN	0.5714	0.5873	0.5794	0.5794
Buy-and-Hold		0.3968	0.4444	0.4683	0.4365

generate higher average SIGNs than the buy-and-hold account. Nonetheless, it seems that neither the neural networks nor the technical indicators can accurately signal the direction of stock price movement because of the relatively low average SIGN; although each of the neural network models is unquestionably better than the model using the individual technical indicators or the buy-and-hold strategy, since the average SIGN of the neural network models is far more accurate and consistently predictive. Particularly, it was observed that the FNN has generated the highest average SIGN (0.5900) obtainable from the experiment. The highest average SIGN of the technical analysis approach is generated by the MACD.

When each SIGN of the neural networks and technical indicators was examined more closely, it was found that the FNN signals the highest SIGN (0.6429 of the CAT security) that can be obtained from the neural network experiment, while the highest SIGN (0.6168 of the DAL security) of the technical analysis can be achieved by using the MACD. In particular, it was also observed that only five out of the thirteen technical analysis experiments (for each of the five technical indicators and for the prediction in each of the three securities less two unsuccessful predictions of the MFI) generate the correct signs that are higher than the randomness of 0.5.

Even so, the MACD actually outperforms all of the neural network models when it was tested only on the DAL security. More importantly, the correct signs produced by all of the twelve neural network forecasts, for each of the four network models, and for the prediction in each of the three securities, are always greater than or equal to 0.5317. This reveals that the neural networks perform more accurately in predicting the direction of future stock price movement. In fact, this result also suggests that the correct signs generated by each neural network model are better than random. Noteworthy, the PortNN, which generates the second highest average SIGN, has achieved a relatively constant predictability performance among all three securities examined in the study.

Trading Profitability

After performing the trading simulation, the total number of transactions (trades) and the average daily returns on investment obtained from stock trading practices guided by the neural networks and the technical indicators over the second period (July 2, 1999 to December 30, 1999) were calculated and are presented in Table 2.

According to the profitability results, it was observed that the RSI and the buy-and-hold account generate negative average daily returns for all of the three securities examined in the study. In contrast, an average return of the LMT security received from trading directed by the MFI is positive (0.1023%), even though its predictability (0.4884) is actually less than the randomness of 0.5. As previously mentioned, the trading exercises of the CAT and DAL securities directed by the MFI never took place, since the strong regularities were not detected by the MFI during the trading periods. In addition, even though the MA yields the worst negative average returns obtained for the CAT and DAL trading exercises, it actually provides the highest positive average return (0.3042%) that can be

TABLE 2 Profitability Results

		LMT		CAT		DAL		Portfolio
		# of Trades	*Return*	# of Trades	*Return*	# of Trades	*Return*	*Return*
Technical Indicators	RSI	8	– 0.1004%	4	– 0.0015%	2	– 0.0817%	– 0.0612%
	MFI	2	0.1023%	–	–	–	–	–
	MA	14	0.3042%	52	– 0.3663%	48	– 0.3483%	– 0.1368%
	SO	8	– 0.1523%	6	– 0.0659%	8	0.1039%	– 0.0381%
	MACD	12	– 0.3324%	32	0.1644%	36	0.1748%	0.0023%
Neural Networks	FNN	32	0.1341%	90	0.0479%	100	– 0.1108%	0.0237%
	PNN	18	0.2654%	30	0.1111%	30	0.0773%	0.1513%
	LVQ	50	0.0371%	56	0.0608%	92	0.0959%	0.0646%
	PortNN	30	0.1085%	62	0.0502%	86	0.0407%	0.0665%
Buy-and-Hold		2	– 0.3771%	2	– 0.2032%	2	– 0.1146%	– 0.2316%

obtained in the experiment when the LMT trading practices are performed.

Similarly, the SO produces a 0.1039% average return on the DAL trading exercises, while the other two trading practices do not generate positive average return results. Again, it should be noted that though the SO produces the 0.5050 of SIGN on the CAT-price movement predictions, the resulting average return is negative. Particularly, it was found that the MACD generates the highest positive average returns when the trading practices were tested on the CAT and DAL securities (0.1644% and 0.1748%, respectively). Once again, the predictability and profitability performances of the MACD on the LMT security are poor compared to those of the other developed models.

With regard to the neural network predictions, it was found that the two highest average returns of the LMT and CAT securities (0.2654% and 0.1111%, respectively) are directed by the PNN forecasts, while the highest average returns of the DAL security (0.0959%) is driven by the LVQ prediction. In addition, it was observed that the FNN has obtained positive average return results of 0.1341% and 0.0479% on the LMT and CAT trading exercises, whereas the average return of DAL stock trading (−0.1108%) is unexpectedly negative.

The remaining three neural networks (PNN, LVQ, and PortNN) generate positive average returns for all of the trading practices evaluated in the three securities of this study. Even though the highest average returns of the neural networks are not superior to those of the technical indicators, this empirical finding indicates that the overall trading profits directed by the neural network predictions are consistently better than those guided by the technical analysis recommendations.

The DAL trading directed by the FNN was examined more closely to figure out why the average return is not positive, even though the SIGN (0.5794) generated by the FNN forecast is higher than random. In fact, this SIGN is identical to those of the LVQ and PortNN. It is found that the number of trades (100) made for the DAL trading exercise directed by the FNN prediction is the highest possible found in this study. Further, it was discovered that the FNN has actually generated a positive 0.1884% of average investment return on the DAL trading when the transaction cost is not taken into consideration.

This finding indicates that excessive trading is mainly responsible for the negative average return result of the DAL trading, suggesting that a higher SIGN does not always provide higher investment return. The decrease in the investment return of stock trading is due to an increase in a total number of trading exercises, especially when the transaction cost is taken into consideration. Therefore, a correlation between the SIGN and the Return was also calculated to examine whether the SIGN is strongly related to trading profits, especially when the profit opportunities are washed out by the transaction costs. The relationship ρ(SIGN, Return) = 0.7322 was calculated and indicates a positive relationship, meaning the investment return obtained from trading is positive, and almost certainly correlated to the SIGN. Furthermore, it was observed that the number of trading-based forecasts of the neural networks were considerably higher than those of the technical indicators.

The last column in Table 2 indicates the average rate of return that can be obtained from holding an equally weighted portfolio of the three securities. In other words, the investment return on the portfolio is the equally weighted average of the investment returns of each security comprising the portfolio. The reason for forming this stock portfolio in the present study is to limit an investor's risk exposure of always investing in a particular security. In this study, the portfolio return can be derived as:

$$Return_P = w \times Return_1 + w \times Return_2 + w \times Return_3, \tag{10}$$

where Return_P is the average rate of return on holding the stock portfolio, $w = 1/3$ is the weight of investment, and Return_1, Return_2, and Return_3 are the average returns of investing in the LMT, CAT, and DAL securities, respectively.

Similar to the average SIGN, the average portfolio returns directed by the neural networks and technical indicators are better than that generated by the buy-and-hold strategy. In fact, the buy-and-hold account obtains the lowest average daily portfolio return (−0.2316%) over the 126 trading days. The trading results show that the MFI and MACD for technical analysis, as well as all developed models of the neural networks, generate positive average portfolio returns.

Particularly, the results show that the PNN, LVQ, and PortNN of the neural networks significantly generate higher average portfolio returns than those of the technical analysis. Specifically, the PNN, which yields an average portfolio return of 0.1513%, is the best performer among the developed models evaluated in this study. In fact, this 0.1513% is approximately equal to a total portfolio return of 19.06% over the 126 trading days. This implies that the neural networks can be used as an effective tool not only to enhance the predictive ability of the technical analysis indicators, but also to increase the profit opportunities of the trading practices.

Finally, Table 3 shows the measure of resulting risks (σ and SR) that occurred from trading guided by the neural networks and technical indicators. Again, the last two columns in Table 3 indicate the risks of holding the equally weighted portfolio of the three securities. The portfolio σ was calculated using the following formula.

$$\sigma_P = \sqrt{w^2(\sigma_1^2 + \sigma_2^2 + \sigma_3^2) + 2w^2(cov_{12} + cov_{13} + cov_{23})}, \tag{11}$$

where σ_P is the risk of holding the stock portfolio, σ_1, σ_2, and σ_3 are the risk of investing in the LMT, CAT, and

TABLE 3 Trading Risk Results

		LMT		CAT		DAL		Portfolio	
		σ	*SR*	σ	*SR*	σ	*SR*	σ	*SR*
Technical Indicators	RSI	2.08%	− 0.0484	1.67%	− 0.0009	2.07%	− 0.0395	1.03%	− 0.0592
	MFI	2.30%	0.0445	–	–	–	–	–	–
	MA	2.04%	0.1489	1.58%	− 0.2325	1.72%	− 0.2021	1.01%	− 0.1348
	SO	2.07%	− 0.0734	1.88%	− 0.0350	2.00%	0.0520	1.06%	− 0.0360
	MACD	2.10%	− 0.1580	1.72%	0.0955	1.71%	0.1021	1.07%	0.0021
Neural Networks	FNN	1.93%	0.0695	1.39%	0.0345	1.49%	− 0.0745	0.98%	0.0242
	PNN	1.99%	0.1333	1.73%	0.0643	1.88%	0.0411	1.08%	0.1396
	LVQ	1.82%	0.0203	1.60%	0.0381	1.55%	0.0618	0.95%	0.0683
	PortNN	1.94%	0.0560	1.55%	0.0323	1.56%	0.0261	0.86%	0.0772
Buy-and-Hold		2.11%	− 0.1790	1.91%	− 0.1065	2.10%	− 0.0545	1.25%	− 0.1846

DAL securities, respectively, and cov_{12}, cov_{13}, and cov_{23} are the covariance of the daily trading returns between the two securities.

It can be observed that the σ results based on trading for each stock guided by the neural networks were obviously less than those received from the buy-and-hold account, and generally less than those guided by the technical indicators. In addition, it was found that the σ of holding the stock portfolio is reduced to about half that of holding the individual stock. In particular, it was observed that the PortNN provides the lowest portfolio σ (0.86%) evaluated in the study. Nonetheless, the PNN generated a slightly higher portfolio σ (1.08%) than those of the technical indicators. With respect to the risk-adjusted rate of return as measured by the *SR*, the results confirm again that trading exercises guided by the neural networks consistently outperform those directed by both the technical indicators and the buy-and-hold account. Particularly, it was found that when investing in the stock portfolio, the PNN generates the highest *SR* (0.1396) obtainable from the experiment.

CONCLUSION

The predictive ability of technical indicators can be improved by adopting neural networks to uncover the underlying nonlinear patterns of these technical indicators for short-term stock trend prediction. Three stocks across different market industries were tested to support the robustness of the neural network models. The results lead to the conclusion that different technical indicators, which rely heavily on strong regularities, often do not work consistently for all securities, which in turn generates varied stock signals resulting in diverse trading profits.

This may be due to the fact that the rationales behind these signal recommendations come from different criteria. Neural networks seem to be a perfect tool to uncover and manage these irregularities, since they independently learn the underlying relationships of various technical indicators on a selected security. In particular, neural networks can be used as a decision maker that automatically helps investors perform stock trading when decisions received from different indicators contradict each other.

More importantly, it is found in the study that the profitability improvement is positively correlated to the predictability measure (SIGN). Nonetheless, it should be noted that higher and lower SIGNs do not always guarantee higher and lower profitability, respectively, as discovered in the study. On one hand, the excessive trading exercises may be partly responsible for the reduction of trading returns when transaction cost is taken into consideration. On the other hand, it may be due to the fact that higher trading profits can be obtained from an accurate prediction when the actual stock price changes are highly volatile.

This observation suggests the importance of making an accurate asset allocation (between long and short positions) when the actual positive or negative price changes of the next trading days are significant. Therefore, potentially higher investment return may be obtained from training the networks to correctly predict the directions of stock movement only when significant profit opportunities exist.

In conclusion, our findings suggest that the neural networks can improve the effectiveness of traditional technical analysis, and, thus, help investors make better investment decisions. However, this study covers only technical indicators, while fundamental analysis remains intact. It is far from perfect, as the fundamental available information has been proven to provide some predictive factors in stock price and stock return forecasting. In fact, there are numerous studies done by both academics and practitioners in this area. If both technical and fundamental approaches are thoroughly examined and included during neural network modeling, it would no doubt be a major improvement in predicting future stock movements.

As discussed earlier, the total number of trading exercises directed by the neural network forecasts is obviously higher than those guided by technical analysis. In fact, a prediction of 1% upward or downward change (or greater) in stock prices could be further employed to represent a reasonable opportunity to make a profitable trade. In addition, a percent buffer can be used to replicate the major turning points of stock trends identified by conventional technical analysis. The main idea is that if the predicted price changes are not large enough to compensate a round-trip cost of transaction, the current position will be maintained instead of switching from a long position to a short position, or vice versa. It would be beneficial for investors to perform trading only when the profits from doing so exceed the costs. Furthermore, additional criteria of the trading strategies should be further designed, since the success of trading practices is critically dependent on the effective control of both limiting losses and protecting profits. Finally, future research should consider more stock evaluations and trading simulations under different scenarios of transaction costs and individual tax brackets to replicate realistic investment practices.

REFERENCES

Achelis, S.B. 1995. *Technical Analysis from A to Z: Covers Every Trading Tool from the Absolute Breadth Index to the Zig Zag.* Chicago, IL: Probus Publisher.

Austin, M., C. Looney, and J. Zhuo. 1997. Security market timing using neural network models. *New Review of Applied Expert Systems* 3:3–14.

Barberis, N., A. Shleifer, and R. Vishny. 1998. A model for investor sentiment. *Journal of Financial Economics* 49:307–343.

Brown, S.J., W.N. Goetzmann, and A. Kumar. 1998. The Dow theory: William Peter Hamilton's track record reconsidered. *Journal of Finance* 53:1311–1333.

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

Chan, K.C.C., and F.K. Teong. 1995. Enhancing technical analysis in the FOREX market using neural networks. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1023–1027, Piscataway, NJ.

Chandra, N., and D.M. Reeb. 1999. Neural networks in a market efficiency context. *American Business Review* 17:39–44.

Chenoweth, T., and Z. Obradovic. 1996. Embedding technical analysis into neural network based trading systems. *Applied Artificial Intelligence* 10:523–541.

Cogger, K.O., P.D. Koch, and D.M. Lander. 1997. A neural network approach to forecasting volatile international equity markets. *Advances in Financial Economics* 3:117–157.

Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 2:303–314.

Darrat, A.F., and M. Zhong. 2000. On testing the random-walk hypothesis: a model-comparison approach. *The Financial Review* 35:105–124.

Demuth, H., and M. Beale. 1998. *Neural Network Toolbox: for Use with MATLAB*, 5th edition. Natick, MA: The Math Works, Inc.

Desai, V.S., and R. Bharati. 1998. A comparison of linear regression and neural network methods for predicting excess returns on large stocks. *Annals of Operations Research* 78:127–163.

Dropsy, V. 1996. Do macroeconomic factors help in predicting international equity risk premia?: Testing the out-of-sample accuracy of linear and nonlinear forecasts. *Journal of Applied Business Research* 12:120–133.

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

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

Fernandez-Rodriguez, F., C. Gonzalez-Martel, and S. Sosvilla-Rivero. 2000. On the profitability of technical trading rules based on artificial neural networks: Evidence from the Madrid stock market. *Economic Letters* 69:89–94.

Gencay, R., and T. Stengos. 1998. Moving averages rules, volume and the predictability of security returns with feedforward networks. *Journal of Forecasting* 17:401–414.

Hagen, M.T., H.B. Demuth, and M. Beale. 1996. *Neural Network Design*. Boston, MA: PWS Publishing Company.

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

Jegadeesh, N. 2000. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation—Discussion. *Journal of Finance* 55:1765–1770.

Kim, K.J., and I. Han. 2000. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications* 19:125–132.

Kohonen, T. 1995. *Self-Organizing Maps*. New York/Berlin: Springer.

Kuo, R.J. 1998. A decision support system for the stock market through integration of fuzzy neural networks and fuzzy delphi. *Applied Artificial Intelligence* 12:501–520.

Lam, K., and K.C. Lam. 2000. Forecasting for the generation of trading signals in financial markets. *Journal of Forecasting* 19:39–52.

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

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

Liu, N.K., and K.K. Lee. 1997. An intelligent business advisor system for stock investment. *Expert Systems* 14:129–139.

Longo, J.M., and M.S. Long. 1997. Using neural networks to differentiate between winner and loser stocks. *Journal of Financial Statement Analysis* 2:5–15.

Malkiel, B.G. 1995. *A Random Walk Down Wall Street*, 6th edition. New York, NY: Norton & Co.

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

Murphy, J.J. 1999. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. New York, NY: New York Institute of Finance.

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

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

Plummer, T. 1989. *Forecasting Financial Markets: The Truth Behind Technical Analysis*, London, UK: Kogan Page.

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

Qi, M. 1999. Nonlinear predictability of stock returns using financial and economic variables. *Journal of Business & Economic Statistics* 17:419–429.

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

Quah, T., and B. Srinivasan. 1999. Improving returns on stock investment through neural network selection. *Expert Systems with Applications* 17:295–301.

Rumelhart, D.E., and J.L. McClelland. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Cambridge, MA: The MIT Press.

Saad, E.W., D.V. Prokhorov, and D.C. Wunsch. 1998. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on Neural Networks* 9:1456–1470.

Schwager, J.D. 1995. *Fundamental Analysis*. New York, NY: John Wiley & Sons.

Schwager, J.D. 1996. *Technical Analysis*. New York, NY: John Wiley & Sons.

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

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

Tan, C.N.W., and G.E. Wittig. 1993. A study of the parameters of a backpropagation stock price prediction model. In *Proceedings of First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, pages 288–291, Dunedin, New Zealand.

Tsaih, R., Y. Hsu, and C.C. Lai. 1998. Forecasting S&P 500 stock index futures with a hybrid AI system. *Decision Support Systems* 23:161–174.

Vamsi, B.K., D. Enke, and C. Dagli. 2001. Intelligent technical stock analysis using fuzzy logic and trading heuristics. *Intelligent Engineering Systems through Artificial Neural Networks* 11:739–744.

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

Wong, R.K., and P.N. Ng. 1994. A Hybrid approach for automated trading systems. In *Proceedings of the IEEE Australian and New Zealand Conference on Intelligent Information Systems*, pages 278–282, Piscataway, NJ.

Zemke, S. 1999. Nonlinear index prediction. *Physica A.* 269:177–183.