Embedding Technical Analysis into Neural Network Based Trading Systems

Tim Chenoweth[1,2,3] Zoran Obradović[1,*] Sauchi Stephen Lee[4]

[1] School of Electrical Engineering and Computer Science

[2] Department of Management and Systems

[3] Department of Economics

Washington State University, Pullman WA 99164-2752, USA

[4] Division of Statistics

University of Idaho, Moscow, ID 83843, USA

Abstract

We have recently proposed a promising trading system for the S&P 500 index, which consists of a feature selection component and a simple filter for data preprocessing, two specialized neural networks for return prediction, and a rule base for prediction integration. The objective of this study is to explore if including additional knowledge for more sophisticated data filtering and return integration leads to further improvements in the system. The new system is using a well-known technical indicator to split the data, and an additional indicator for reducing the number of unprofitable trades. Several system combinations are explored and tested over a five year trading period. The most promising system yielded an annual rate of return (ARR) of 15.99% with 54 trades. This compares favorably to the ARR for the buy and hold strategy (11.05%) and the best results obtained using the system with no technical analysis knowledge embedded (13.35% with 126 trades).

*Research sponsored in part by the NSF research grant NSF-IRI-9308523.

Running Title:

Technical Based Neural Networks Trading

The Corresponding Address:

Zoran Obradović,

Electrical Engineering and Computer Science,

Washington State University,

Pullman, WA 99164-2752, USA

e-mail zoran@eecs.wsu.edu,

phone 509 335-6601,

fax 509 335-3818

1. Introduction

The attempts to model financial markets phenomena in order to predict future market directions are largely unsuccessful due to the inherent complexity of the domain. The efficient market hypothesis, tested in the economics literature over a 30 years period without definitive findings, claims that the financial markets are random time series and consequently are unpredictable based on any amount of publicly available knowledge. However, until recently most quantitative approaches to testing this hypothesis were based on linear time series modeling (Black and Scholes, 1973; White, 1988). It is very hard to find statistically significant market inefficiencies using standard linear time series modeling since such linear approaches are not capable of identifying dynamic or nonlinear relationships in the historic data. However, given enough data and time, an appropriate nonparametric machine learning technique may be able to discover more complex nonlinear relationships through supervised learning from examples (Weiss, 1991). The last few years have seen such new approaches to financial modeling by both researchers in financial service companies (e.g. Mahfoud and Mani, 1995) and universities (e.g. Hutchinson, 1993).

To analyze nonlinear phenomena in the stock markets this study employs multi-layer artificial neural networks (Haykin, 1995). Neural networks (NN) are powerful computational systems that can approximate almost any nonlinear continuous function on a compact domain to any desired degree of accuracy (Cybenko, 1989). In addition, a NN can account for fundamental changes in the underlying function through incremental retraining using the back-propagation learning algorithm(Rumelhart et al.,1986). This study focuses on the United States stock market because it is one of the most closely followed markets in the world and as such very efficient. It is reasonable to assume that if the system proposed in this study can find inefficiencies in the U.S. stock market, it should also be able to find inefficiencies in other markets that are less closely watched and as such are more inefficient.

Recently, we obtained promising results by incorporating two specialized neural networks into a hybrid multi–component nonlinear system for the S&P 500 stock market predictions (Chenoweth and Obradovic, 1995a). The system was composed of a feature selection component, a filtering component for identifying the most relevant patterns for two specialized NNs (called the Up NN and the Down NN) trained to predict stock market returns, and a high level decision rule component used for determining buy/sell recommendations as a function of the two predictions obtained from the Up and Down NNs.

This paper focuses on the pattern filter and the buy/sell recommendation component of our previous system. The objective is to explore whether technical analysis based preprocessing and postprocessing improves the overall system performance. The results from this study are compared to those achieved using the previous system. Section 2 gives a global description of the proposed trading system. Details of the filtering techniques and predicted returns integration techniques are discussed in Sections 3 and 4, respectively. Finally, the experimental results are presented in Section 5 and the conclusions in Section 6.

2. The Return Rate Prediction Process

In the proposed system, the trading process (described in Figure 1) consists of three phases: data preprocessing, return rate prediction, and postprocessing. For data preprocessing, this paper explores two distinct directional filtering schemes for separating the training patterns into "up trend," "down trend," and "sideways" data sets. The first directional filter is a previously used simple approach that separates the training patterns according to the sign of the target return for a specific pattern and a prespecified threshold value. The second directional filter computes an indicator commonly used among technical analysts to determine market direction and strength of the market trend. Patterns covering a continuous, fixed size time segment of historic data (called a training

window) are designated to the Up and Down training sets based on the direction of the market trend. Section 3 describes the details for both schemes.

The return rate prediction is computed using two NNs (called the Up NN and the Down NN). Both the Up and Down NN are trained using the back-propagation algorithm (Rumelhart et al., 1986), and the disjoint training sets are generated in the preprocessing phase. Once both NNs are trained, a return is predicted by both the Up and the Down NN for the time step immediately following the training window and the predictions are integrated in the result postprocessing phase. After a single prediction step, the training window is shifted forward by one day and the patterns from the new window are used to retrain both NNs, and a prediction is made for the next day. This process is repeated until the data set is exhausted.

For example, suppose that the training window size is m and at time t the test pattern is d_t, which means the training window contains patterns d_{t-m} through d_{t-1}. First, the patterns in the training window (d_{t-m} through d_{t-1}) are separated into a discard set, an Up NN training set, and a Down NN training set using one of the pattern filtering schemes discussed in Section 3. Next, the Up and the Down NN are trained using the "up trend" and the "down trend" data sets respectively, and asked to predict the target return for test pattern d_t. Once the two predictions are collected and sent to the integration component, the training window is shifted forward one time step, so that the new test pattern is d_{t+1} and the new training window contains patterns d_{t-m+1} through d_t, and the process is repeated. This continues until the end of the ordered data set is reached.

In the postprocessing phase, the predicted returns from both the Up and the Down NN are used to compute a buy/sell recommendation as a function of those returns. This study examines four integration techniques. The first is the method used in the previous system proposed in (Chenoweth and Obradovic, 1995a). It uses a simple decision rule, which in essence determines which specialized NN is "more confident" of it's prediction, with the buy/sell recommendation based

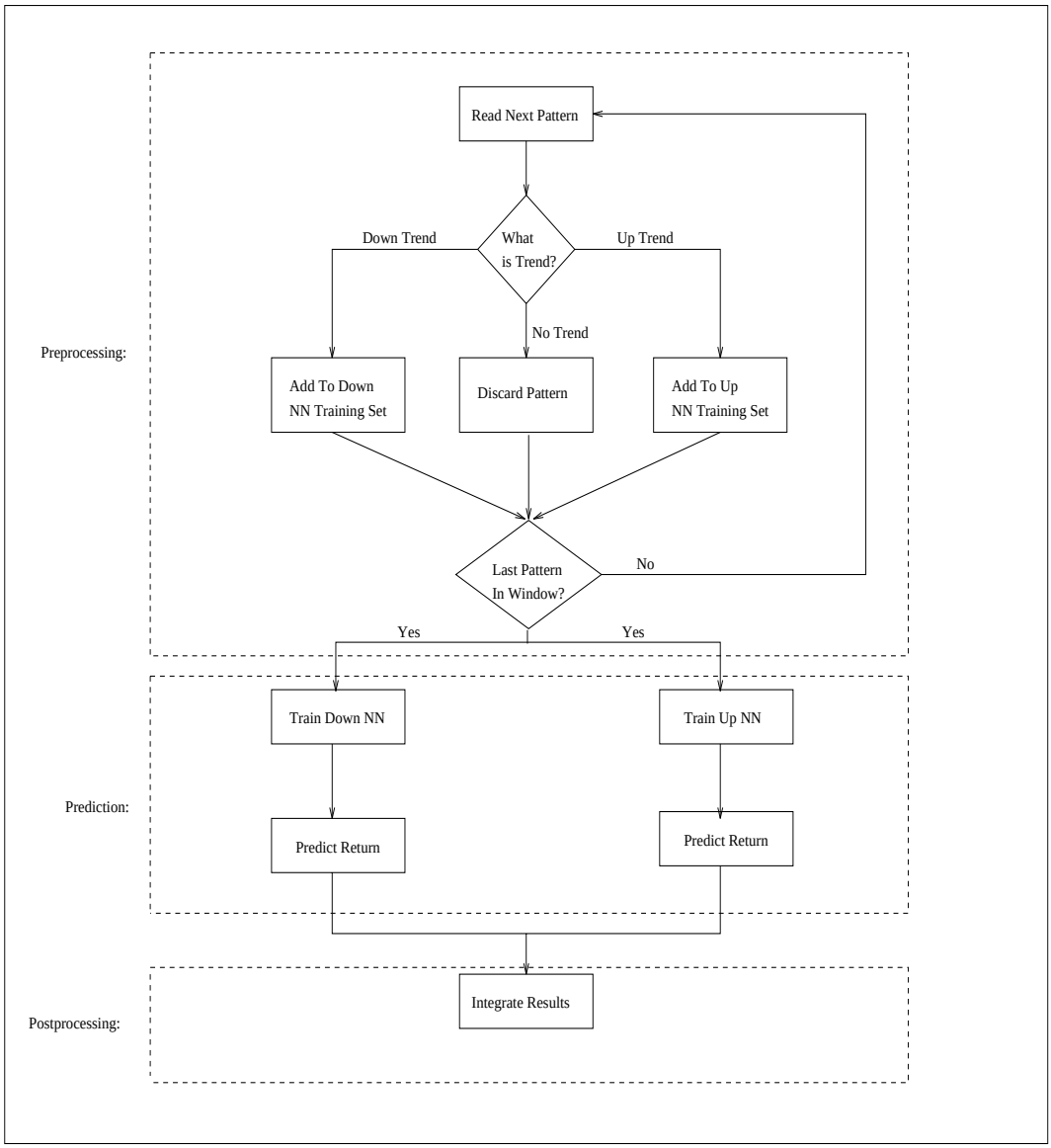

Figure 1: The Proposed Trading Process.

on that NN. The second integration technique is a voting system where each of the Up NN, the Down NN, and a technical indicator vote on the market direction (i.e. up or down). The system's buy/sell recommendation is based on the results of this vote. The third integration technique uses an additional NN that acts as a combiner for the predictions of the Up and the Down NN. This combiner NN takes the predicted returns for both the Up and the Down NN and integrates them into a single prediction, which is used to compute the system's buy/sell recommendation. Finally, the fourth integration technique is an extension of the third. This technique also uses a combiner NN, but the resulting buy/sell recommendation is filtered using a technical indicator in an effort to reduce the number of unprofitable trades. Details for all integration strategies are provided in Section 4.

3. Pattern Filtering Schemes

This section presents the details for two different approaches to the preprocessing phase of the trading system shown in Fig. 1.

3.1. Threshold Based Directional Filter

The previous system proposed in (Chenoweth and Obradovic, 1995a) used a simple data filtering approach where for each training session the target return corresponding to each pattern in the window is compared to a threshold value h_1. If the return is greater than h_1, the corresponding pattern is added to the Up NN training set, if the return is less than $-h_1$ the pattern is added to the Down NN training set. Any pattern with a target return between $-h_1$ and h_1 is discarded.

3.2. The ADX Based Directional Filter

The more sophisticated filtering approach considered in this study uses the average direction index (ADX) developed by J. W. Wilder, Jr., in the mid-1970's and further modified by several technical analysts (Elder, 1993; LeBeau and Lucas, 1992). The ADX indicator identifies trends and quantifies their strengths by measuring the fraction of today's range extends above or below the previous day's range and averages this over a period of time. It is computed using the following algorithm:

1. Compute the positive and negative directional movements (DM^+ and DM^-) as

$$\mathrm{DM}^+ = \begin{cases} \max\{T_h - Y_h, 0\} & \text{if } T_h - Y_h > Y_l - T_l, \\ 0 & \text{otherwise,} \end{cases}$$

   and

$$\mathrm{DM}^- = \begin{cases} \max\{Y_l - T_l, 0\} & \text{if } T_h - Y_h < Y_l - T_l, \\ 0 & \text{otherwise,} \end{cases}$$

   where T_h and T_l are today's market high and low values, and Y_h and Y_l are yesterday's market high and low values. It is important to note that every day has both a DM^+ and a DM^-, and that at most one of these two values is positive, while the other is zero. For example, suppose today's high and low values are 150 and 100 respectively and yesterday's high and low values are 140 and 105. Since $|150 - 140| > |100 - 105|$ the DM^+ is 10 while the DM^- is zero.

2. Measure the true range (TR) as

$$\mathrm{TR} = \max\{|T_h - T_l|, |T_h - Y_c|, |T_l - Y_c|\},$$

   where T_h, T_l, Y_h and Y_l are as previously defined and Y_c is yesterday's market closing value.

3. Compute the smoothed directional indicators DI^+ and DI^- as

$$\mathrm{DI}^+ = \frac{\mathrm{DM}^+}{\mathrm{TR}}, \quad \mathrm{DI}^- = \frac{\mathrm{DM}^-}{\mathrm{TR}},$$

4. Compute $\mathrm{DI}^+_{k_1}$ and $\mathrm{DI}^-_{k_1}$ as the average DI^+ and DI^- for the previous k_1 days.

5. Calculate the DX or directional movement index as

$$\mathrm{DX} = \frac{|\mathrm{DI}^+_{k_1} - \mathrm{DI}^-_{k_1}|}{\mathrm{DI}^+_{k_1} + \mathrm{DI}^-_{k_1}} * 100,$$

6. Finally, compute a moving average of the DX over k_2 previous days to create the function $\mathrm{Adx}(k_1, k_2)$.

Observe that the Adx is always between 0 and 100, with small values indicating that the market is moving sideways (i.e. there is no trend), and large, or raising, values indicating that the market has a trend, in which case the $\mathrm{DI}^+_{k_1}$ and $\mathrm{DI}^-_{k_1}$ for that day are compared to determine the market direction.

In this study, two ADX indicator based rules for filtering the data are tested. They are denoted by $\mathrm{ADX}_1(k_1, k_2, h_2)$ and $\mathrm{ADX}_2(k_1, k_2, h_2)$, which means Average Direction Index rule 1 and 2, with parameters k_1, k_2, and h_2.

- **Filtering rule $\mathbf{ADX_1(k_1, k_2, h_2)}$** compares today's $\mathrm{Adx}(k_1, k_2)$ value to a threshold value h_2. If today's $\mathrm{Adx}(k_1, k_2)$ value is less than h_2 or is less than yesterday's $\mathrm{Adx}(k_1, k_2)$ value, the pattern is discarded. If today's $\mathrm{Adx}(k_1, k_2)$ value is larger than h_2 *and* is larger than yesterday's $\mathrm{Adx}(k_1, k_2)$ value, today's $\mathrm{DI}^+_{k_1}$ and $\mathrm{DI}^-_{k_1}$ are compared. If $\mathrm{DI}^+_{k_1} > \mathrm{DI}^-_{k_1}$, the pattern is added to the Up NN's training set; else the pattern is added to the Down NN's training set.

- **Filtering rule $\mathbf{ADX_2(k_1, k_2, h_2)}$** again compares today's $\mathrm{Adx}(k_1, k_2)$ value to a threshold h_2. If either today's $\mathrm{Adx}(k_1, k_2)$ or yesterday's $\mathrm{Adx}(k_1, k_2)$ is below h_2, the pattern is discarded. If today's and yesterday's $\mathrm{Adx}(k_1, k_2)$ are above h_2, then today's $\mathrm{DI}^+_{k_1}$ and $\mathrm{DI}^-_{k_1}$ are compared and the pattern is assigned to the correct training set as in filtering rule ADX_1.

4. Predicted Returns Integration

This component analyzes the predicted returns obtained from the Up NN and the Down NN, and outputs a buy/sell recommendation that is used to establish either a long or short position in the market. A *long position* means purchasing an asset for later resale, while a *short position* means selling a borrowed asset now and purchasing it later. Four integration strategies are examined in this study. The first is the rule based strategy used earlier (Chenoweth and Obradovic, 1995a). The second strategy is a voting method where each of the Up NN, the Down NN, and the moving average convergence-divergence technical indicator (MACD) vote on the market direction. The third strategy uses a combiner NN to integrate the predictions of the Up NN and the Down NN into a single prediction. Finally, the fourth is an extension of the third strategy, which uses the MACD indicator to verify the prediction of the combiner NN. The details of each strategy are given in the following subsections.

4.1. The Rule Based Strategy

For the rule based strategy, the predicted returns from both the Up and the Down NN are used as input to the rule based integration component as shown in Figure 2. This strategy uses a simplistic rule which is an extension of the "buy and hold" strategy meaning that if the system does not have a strong recommendation, a long position is established. The decision rule first compares the Up NN prediction r_u to the Down NN prediction r_d and determines a buy/sell recommendation as

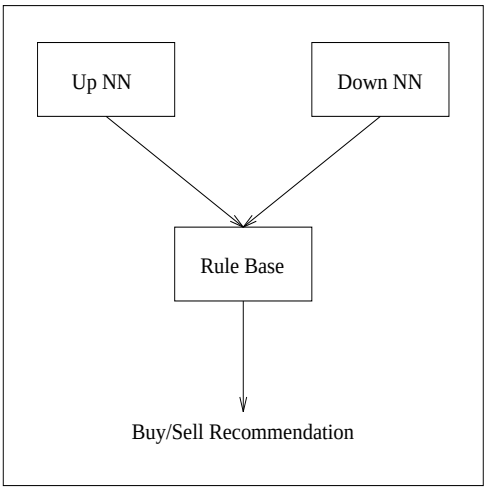

Figure 2: The Rule Based Integration Strategy.

follows:

a. If $r_u > 0$ and $r_d \geq 0$ a long position in the market is established or maintained.

b. If $r_u \leq 0$ and $r_d < 0$ a short position in the market is established or maintained.

c. Otherwise, the normalized difference *diff* is computed as

$$diff = \frac{\max\{|r_u|, |r_d|\} - \min\{|r_u|, |r_d|\}}{\max\{|r_u|, |r_d|\}},$$

and this ratio is compared to a predefined decision threshold value y to determine a buy/sell recommendation as follows:

i. If $r_u > 0$, $r_d < 0$, $diff > y$, and $|r_u| < |r_d|$, a short position in the market is established or maintained.

ii. Otherwise, a long position in the market is established or maintained.

4.2. The Voting Integration Strategy

This strategy utilizes the MACD indicator to resolve ties between predictions suggested by the Up and The Down NN as shown in Fig. 3. The MACD indicator is a well known stock market timing

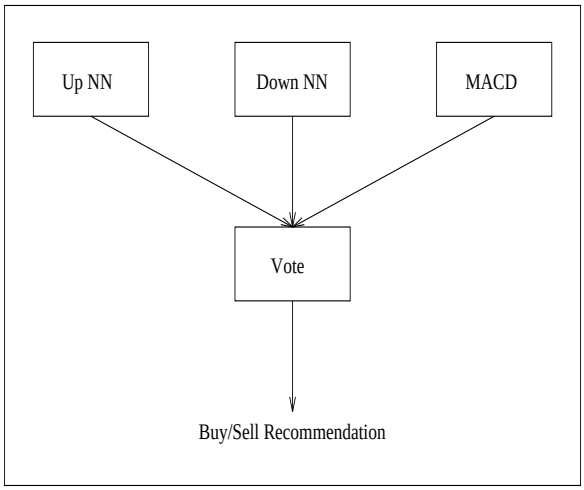

Figure 3: The Voting Integration Strategy.

device, originally developed in 1979 by G. Appel and effectively used by various traders (Elder, 1993). It is comprised of two functions:

- The Macd function, composed of two moving averages, which reacts quickly to market changes;
- The Signal function, a moving average of the Macd function,which reacts more slowly to market changes.

The MACD based trading rule recommends a long position in the market when the value of the Macd function is larger than the value of the Signal function and a short position when the Macd value is smaller than the Signal value. The daily Macd and Signal values are computed using the following algorithm:

1. Compute a moving average of the market closing prices over the ma_1 previous days.
2. Compute a moving average of the market closing prices over the ma_2 previous days, where $ma_1 < ma_2$.

3. Subtract the moving average computed over the ma_2 previous days from the moving average computed over the ma_1 previous days. This becomes the Macd(ma_1, ma_2) value for the current day.

4. Compute a moving average of the Macd values over the ma_3 previous days. This becomes the Signal(ma_1, ma_2, ma_3) value for the current day.

Using the voting integration strategy, both the Up and the Down NN are presented with the test pattern and the predicted returns are collected. The following algorithm determines the result of the vote:

a. If both the Up and the Down NN give a positive prediction, a long position in the market is established or maintained.

b. If both the Up and the Down NN give a negative prediction, a short position in the market is established or maintained.

c. If the Up and the Down NN predictions do not agree on the market direction, the MACD is used to estimate the market direction and a market position is established or maintained accordingly.

4.3. The Combiner Neural Network Integration Strategy

For this strategy the predictions from the Up and the Down NN are integrated using a third combiner NN as shown in Fig. 4. This combiner NN takes the predictions from each specialized NN and returns a single integrated predicted return. This integrated prediction is used to determine the trading action for that day (i.e. maintain current position or change the market position via a trade).

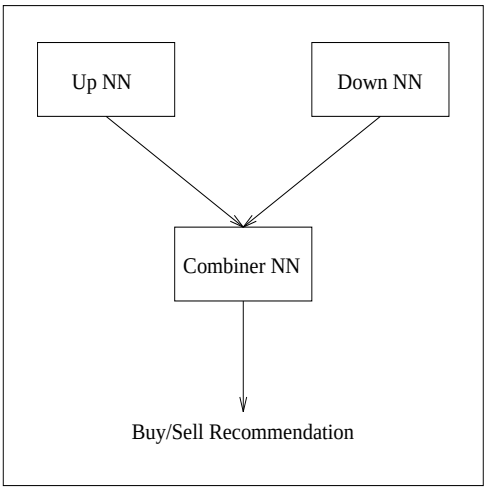

Figure 4: The Combiner Neural Network Integration Strategy.

Both the Up and the Down NN are trained on a portion of the pattern set in a training window, as described in Section 3. Once trained, a second pass is made through *all* patterns in the training window, with each pattern presented to both the Up and the Down NN and the corresponding predictions are combined into a new two dimensional pattern. The result of this second pass is a new set of two dimensional patterns, the cardinality of which is the same as the size of the training window, which becomes the training set for the combiner NN.

Once the combiner NN is trained, the test pattern is presented to both the Up and the Down NN and the resulting predictions are integrated by the combiner NN, which outputs the final return prediction. Finally, the buy/sell recommendation is determined through the following algorithm:

a. If the predicted return obtained from the combiner NN is positive, a long position in the market is established or maintained.

b. If the predicted return is negative, a short market position is established or maintained.

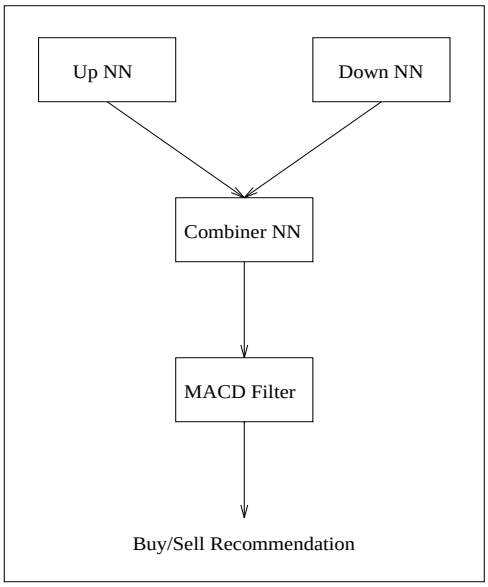

Figure 5: The Combiner Neural Network with MACD Verification Integration Strategy.

4.4. The Combiner Neural Network with MACD Verification Integration Strategy

This strategy is an extension of the previous combiner NN strategy with the objective of reducing the number of unprofitable trades. With this strategy, before a trade actually takes place the recommended market position is verified using the MACD indicator as shown in Figure 5. The following algorithm computes the system buy/sell recommendation:

a. If the MACD recommended position agrees with the NN based combiner system recommendation, the trade occurs and the market position changes.

b. If the MACD recommended position disagrees with the NN based combiner system recommendation, the current market position is maintained.

It is important to note that the MACD indicator is only used to verify the trade action suggested by the NN based combiner system. If the prediction from the combiner NN results in the current market position being maintained, the MACD indicator is not checked. Consequently, this technique reduces the number of trades recommended by the strategy proposed in Section 4.3.

S&P 500 index return
S&P 500 index return lagged one day
S&P 500 index return lagged two days
U.S Treasure Rate lagged 2 months
U.S Treasure Rate lagged 3 months
30 Year Government Bond Rate

Table 1: Features of Each Pattern.

5. Results and Analysis

5.1. Data Description

The system described in Sections 1, 2, 3, and 4 is used for S&P 500 stock market trading. The historic data used in this experiment is ordered daily financial time series patterns from the period January 1, 1982 to December 31, 1993. Patterns from January 1, 1982 to December 31, 1988 comprised the initial training window, whereas actual predictions were made for patterns from January 1, 1989 to December 31, 1993. Table 1 shows the six features used in this study. These features were selected using several statistical based selection techniques and criteria, with the partial results combined using a ranking strategy. A discussion of the details involved in the feature selection process can be found in (Chenoweth and Obradovic, 1995b).

5.2. Performance Measures

The most important criteria when measuring the performance of a stock market prediction model is whether it will make money and how much. Therefore, the model's annual rate of return (ARR) is computed as follows

$$ARR = \frac{k}{n} \sum_{i=1}^{n} r_i,$$

where n is the total number of trading time units for the experiment, k is the number of trading time units per year (i.e., 253 for daily trading), and r_i is the rate of return for time unit i.

The sum, $\sum_{i=1}^{n} r_i$, is computed by either adding or subtracting the actual daily returns for the S&P 500 index. If the system recommends a long position, the actual return is added to the sum; if a short position is recommended, the return is subtracted.

It is also important to minimize transaction costs by controlling excessive trading (i.e. a 10% return with 50 trades is more profitable than a 10% return with 100 trades). Therefore the break even transaction cost ($BETC$), which may be viewed as the return per trade, is computed as follows:

$$BETC = \frac{1}{m} \sum_{i=1}^{n} r_i,$$

where m is the total number of trading transactions, while r_i and n are defined as previously. A trade is defined as any action that changes a market position. For example, exiting the market constitutes a single trade (i.e. a buy trade to cover a short position or a sell trade to cover a long position), while switching from a short position to a long position constitutes two trades (i.e. one buy trade to cover the short position and another buy to establish the long position). Both the ARR and $BETC$ performance measures are used previously in (Chenoweth and Obradovic, 1995s; 1995b; in press; Hutchinson, 1993).

5.3. Directional Filtering Experiments

The Up and the Down NN use identical configuration parameters (shown in Table 2) determined by trial and error. For the ADX based directional pattern filter using the rule based integra-

Parameter	*Value*
Activation Function	Tangent Hyperbolic
Network Topology	6-4-1
Learning Rate	0.03
Tolerance	0.00001
Number of Iterations	5000
Training Window Size	2000

Table 2: Up and Down NN Configuration Parameters.

tion strategy, experiments were conducted for various values of the DI smoothing constant k_1, the DX smoothing constant k_2, and the filtering threshold h_2 using the $\text{ADX}_1(k_1, k_2, h_2)$ and $\text{ADX}_2(k_1, k_2, h_2)$ directional filters. For all experiments, the decision threshold y for the rule based strategy was varied from 0 to 1 in increments of 0.10. The experiments compared the ARR and the number of trades using the ADX filters versus the best values achieved using the threshold based directional filter. The best results for the threshold based filtering system were obtained using a filtering threshold h_1 equal to 0.5%. This system achieved these results using a rule based decision threshold value of 0.8 (ARR equal to 13.35% and the $BETC$ equal to 0.53%). The return for the buy and hold strategy for the period of this study was 11.05%.

Technical analysts (LeBeau and Lucas, 1992) recommend using ADX_1(18,18,15), as their results over a variety of data sets indicate that these are the optimal parameter values for manual trading strategies. Therefore, the initial filtering experiments focused on using directional filter ADX_1(18,18,15). As can be seen from Figures 6 and 7, the ADX_1(18,18,15) based system achieved a smaller ARR using more trades then the system employing the threshold based filtering approach. The best $BETC$ achieved by the ADX_1(18,18,15) based system was 0.05% and the best ARR was 4.51%.

Other technical analysts actually report using ADX smoothing parameters k_1 and k_2 in a range of 10 to 20 days. In (Elder, 93), Elder suggests using Adx(13,13) and so, our trading system experiments shown in Figures 8 and 9 utilized direction filter ADX_1(13,13,10). Experiments using other values for h_2 were conducted, however h_2 equal to 10 achieved the best overall results. The trading system using directional filter ADX_1(13,13,10) performed better than the system using ADX_1(18,18,15). However, both the best ARR (6.56%) and $BETC$ (0.28%) were significantly smaller than the best results achieved by the threshold based filtering system.

A major drawback when using the ADX for NN preprocessing is that the smoothing parameters

introduce a lagging problem. This means that there is a delay between the actual beginning of a trend and the moment the ADX identifies it. This results in important patterns being excluded from the NN training sets. To deal with this problem, the final set of data filtering experiments employed Adx(3,3), which uses smaller smoothing parameters and as such reduces the lagging problem. Technical analysts do not seem to be using such smoothing parameters, as they do not remove enough of the minor market fluctuations. However, in our trading system ADX is used for preprocessing, with predictions made by the NNs, which may be able to distinguish between major trends and minor fluctuations in the market. Figures 10 and 11 show the results obtained using directional filter $\text{ADX}_2(3,3,65)$. The best ARR achieved using $\text{ADX}_2(3,3,65)$ was 12.14%, which is significantly better than the best ARR achieved using $\text{ADX}_1(18,18,15)$ or $\text{ADX}_1(13,13,15)$, somewhat better than the ARR of the buy and hold strategy, and close to the best ARR achieved by the system employing threshold based filtering. However, this ARR was achieved with significantly more trades then the previous simple system (432 versus 126 trades). Due to the large number of trades, the $BETC$ of the best $\text{ADX}_2(3,3,65)$ based system was only 0.14%, which is smaller than the $BETC$ of the best $\text{ADX}_1(13,13,10)$ based system. Additional experiments using directional filters $\text{ADX}_2(3,3,h_2)$ with various values for h_2, and $\text{ADX}_1(3,3,15)$ resulted in a smaller ARR as compared to the directional filter $\text{ADX}_2(3,3,65)$, and as such are not reported.

5.4. Return Integration Experiments

The next set of experiments used the $\text{ADX}_2(3,3,65)$ pattern filter and varied the predicted return integration process. For experiments utilizing the MACD indicator, the moving average values used in the calculation of the Macd(ma_1, ma_2) and Signal(ma_1, ma_2, ma_3) functions were $ma_1 = 12$, $ma_2 = 26$, and $ma_3 = 9$, as recommended in (Elder, 1993). The activation function, tolerance and window size parameters used by the combiner NN were identical to those used for the Up and

Figure 7: Number of Trades Comparison Between the Threshold Based and $\mathrm{ADX}_1(18, 18, 15)$ Based Directional Filters.

Figure 9: Number of Trades Comparison Between the Threshold Based and the $\mathrm{ADX}_1(13, 13, 10)$ Based Directional Filters.

Figure 11: Number of Trades Comparison Between the Threshold Based and the $\mathrm{ADX}_2(3,3,65)$ Based Directional Filters.

the Down NN (shown in Table 2). However, the combiner network topology (2-2-1), the learning rate (0.0001) and the number of iterations (1000) were modified. Results of all experiments are summarized in Table 3. The first column in this table enumerates the systems tested in this section. The next four columns (*Rule*, *Vote*, *NN*, *MACD*) identify the return integration strategy used (rule based, voting, combiner NN, and combiner NN with MACD filter, respectively). A column marked with an "X" specifies that a particular integration strategy is used with a specific system. The remaining columns show the experimental results for the specified system.

As discussed in the previous section, System A employing the ADX filter and rule based integration achieved an ARR of 12.14%, which is better than the ARR for the buy and hold strategy. However, this is achieved with a high number of trades (432), and consequently the $BETC$ (0.14) of this system is smaller than any other system except the one employing the voting based integration strategy.

The voting based strategy employed by System B had poor results, with an ARR of -1.88 and 406 trades. This is likely due to the Up NN mostly voting to buy and the Down NN mostly voting to sell, too often leaving the final buy/sell decision to the MACD indicator. Consequently the voting method tended to follow trades based solely on the MACD, which by itself is a poor predictor (ARR equal to -0.82%).

System C, employing the combiner NN integration strategy, outperformed System F using the voting strategy and was comparable to the buy and hold strategy with an ARR of 11.03%. In addition, System C had considerably fewer trades than System A, employing the rule based integration strategy, resulting in a better $BETC$ (0.35 versus 0.14). These results indicate that using an additional NN that acts as a combiner for the results of the Up and the Down NNs is an effective strategy.

The best ARR (15.99%) and $BETC$ (1.49%) were achieved by System D, employing the com-

System	*Return Integrator*				*ARR*	*BETC*	*Trades*
	Rule	*Vote*	*NN*	*MACD*			
A	X				12.14	0.14	432
B		X			-1.88	-0.02	406
C			X		11.03	0.35	152
D				X	**15.99**	**1.49**	**54**
E	MACD Only				-0.82	-0.02	198
F	Threshold Filter/ Rule Integration				13.35	0.53	128
G	Buy and Hold				11.05	-	2

Table 3: Return Integration Results.

biner NN with MACD verification strategy. It is encouraging to note that this ARR is significantly better than the buy and hold ARR over the testing period. In addition, it is also significantly better than the ARR achieved by System F, which did not use technical indicators in either the preprocessing or postprocessing phases. Comparing the results achieved by System D with those of System C, it can be observed that the MACD filter did reduce the number of unprofitable trades. Not only did the ARR show significant improvement (15.99% versus 11.03%), but the number of trades was dramatically reduced (54 versus 152).

6. Conclusions and Future Research

The first objective of the study was to compare a NN based trading system using a threshold based pattern filtering technique to a system using a more sophisticated preprocessing technique utilizing the ADX indicator to identify trends in the S&P 500 index. The results indicated that the ADX based directional filter used for preprocessing works better with smaller ADX smoothing parameter values. However, the simple threshold based filtering technique still outperforms the ADX based filtering technique with the rule based integration strategy. We believe this is due to the ADX's inability to adjust quickly to sudden changes in the market's direction taking the form of a spike,

even for small smoothing parameter values. This problem is particularly evident in markets with a downward trend. Still, the annual rate of return obtained utilizing the ADX based filter (12.14%) is encouraging considering the very high number of trades involved (432 trades in 1,273 days). As seen from the results for System D (Table 3), the development of a postprocessing technique that removes a portion of the unprofitable trades could potentially lead to significantly higher returns.

The second objective was to explore if embedding technical analysis knowledge in the postprocessing phase would result in better buy/sell recommendations. The use of the MACD indicator to resolve conflicts between the predictions of the Up and Down NNs (i.e. the voting integration strategy) gave unsatisfactory results. This is due to the Up NN mostly voting to buy and the Down NN mostly voting to sell, too often leaving the final buy/sell decision to the MACD indicator. The MACD indicator suffers from the same inability to react quickly to sudden changes as the ADX indicator previously discussed, making the voting strategy inappropriate. However, promising results were obtained using the MACD indicator as a postprocessing filter to a system utilizing an additional NN to combine the Up and the Down NN predictions. The MACD filter successfully increased the system annual rate of return (15.99% vs. 11.03%) and reduced the number of trades (54 vs. 152), resulting in a significant increase in the return per trade ($BETC$ of 1.49 vs. 0.35).

To summarize, the study provides evidence that embedding some form of technical analysis knowledge into a neural network based trading system can improve its predictive capabilities. It is important to observe that in a neural network based trading system, technical indicators may be incorporated in a novel and potentially beneficial manner. For example, this study used the MACD indicator as a filter rather than as a timing mechanism, which is its more traditional usage. In addition, the ADX indicator is not commonly used with smoothing parameter values as small as those utilized in several experiments in this study. However, these small values were employed in the system giving the best overall results. Another important observation is that the object of this study

was to explore if technical analysis knowledge can improve the system performance. Consequently, further research is needed to determine which technical indicators are most appropriate. Additional research is also needed to explore other methods for embedding technical analysis knowledge into neural network based trading systems. Inspired by encouraging results in other domains (Fletcher and Obradovic, 1993; in press; Romero and Obradovic, 1995), we are currently exploring how to combine market information obtained from various technical indicators and utilize it as a prior knowledge for creating a market specific neural network through constructive learning.

References

Black, F., and Scholes, M. 1973. The Pricing of Options and Corporate Liabilities,. Journal of Political Economy. Vol. 81, May-June.

Chenoweth, T., and Obradovic, Z. 1995a. A Multi-component Approach to Stock Market Predictions. Proc. Third Int. Conf. on Artificial Intelligence Applications on Wall Street. pp. 74-79, New York, NY.

Chenoweth, T., and Obradovic, Z. 1995b. An Explicit Feature Selection Strategy for Predictive Models of the S&P 500 Index. Neurove$t Journal. Vol 3, No. 6, November.

Chenoweth, T., and Obradovic, Z. in press. A Multi-Component Nonlinear Prediction System for the S&P 500 Index. Neurocomputing Journal.

Cybenko, G. 1989. Approximation by Superpositions of a Sigmoidal Function. Mathematics of Control, Signals, and Systems. Vol. 2, pp. 303-314.

Elder, A. 1993. Trading For a Living. Traders Press, Inc.

Fletcher, J., and Obradovic, Z. 1993. Combining Prior Symbolic Knowledge and Constructive

Neural Networks. Connection Science: Journal of Neural Computing, Artificial Intelligence and Cognitive Research. vol. 5, nos. 3 & 4, pp. 365-375.

Fletcher, J., and Obradovic, Z. in press. A Discrete Approach to Constructive Neural Network Learning. Neural, Parallel and Scientific Computations.

Haykin S. 1995. Neural Networks, A comprehensive Foundation. New York, NY, MacMilan.

Hutchinson, J. 1993. A Radial Basis Function Approach to Financial Time Series Analysis. unpublished PhD Thesis. Massachusetts Institute of Technology.

LeBeau, C. and Lucas, D. 1992. Computer Analysis of the Futures Market. Traders Press, Inc.

Mahfoud, S., and Mani G. 1995. Genetic Algorithms for Predicting Individual Stock Performance. Proc. Third Int. Conf. on Artificial Intelligence Applications on Wall Street. pp. 174-181, New York, NY.

Romero, P., and Obradovic, Z. 1995. Comparison of Symbolic and Connectionist Approaches to Local Experts Integration. Proc. the IEEE Technical Applications Conference. Nortcon, Portland, OR.

Rumelhart, et al. 1986. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vols. 1 and 2, MIT Press, Cambridge, MA.

White, H. 1988. Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns. Proc. IEEE Int. Conf. on Neural Networks. Vol. 2, pp. 451-458.

Weiss S. M., and Kulikowski C. A. 1991. Computer Systems That Learn. Morgan Kaufmann 1991.

Captions:

Fig. 1: The Proposed Trading Process.

Fig. 2: The Rule Based Integration Strategy.

Fig. 3: The Voting Integration Strategy.

Fig. 4: The Combiner Neural Network Integration Strategy.

Fig. 5: The Combiner Neural Network with MACD Verification Integration Strategy.

Table 1: Features of Each Pattern.

Table 2: Up and Down NN Configuration Parameters.

Fig. 6: ARR Comparison Between the Threshold Based and $ADX_1(18, 18, 15)$ Based Directional Filters.

Fig. 7: Number of Trades Comparison Between the Threshold Based and $ADX_1(18, 18, 15)$ Based Directional Filters.

Fig. 8: ARR Comparison Between the Threshold Based and the $ADX_1(13, 13, 10)$ Based Directional Filters.

Fig. 9: Number of Trades Comparison Between the Threshold Based and the $ADX_1(13, 13, 10)$ Based Directional Filters.

Fig. 10: ARR Comparison Between the Threshold Based and the $ADX_2(3, 3, 65)$ Based Directional Filters.

Fig. 11: Number of Trades Comparison Between the Threshold Based and the $ADX_2(3, 3, 65)$ Based Directional Filters.

Table 3: Return Integration Results.