Trading Equity Index Futures With a Neural Network

Article *in* The Journal of Portfolio Management · January 1992

DOI: 10.3905/jpm.1992.409432

2 authors:

Robert R Trippi
University of California, San Diego
43 PUBLICATIONS **753** CITATIONS
SEE PROFILE

Duane Desieno
Logical Designs
4 PUBLICATIONS **641** CITATIONS
SEE PROFILE

TRADING EQUITY INDEX FUTURES WITH A NEURAL NETWORK

Robert R. Trippi

Duane Desieno

forthcoming in *The Journal of Portfolio Management*

Robert R. Trippi is Professor of Finance at California State University, Long Beach (90840).
Duane Desieno is president of Logical Designs Consulting, Inc. (La Jolla, CA 92037).

Contact:

Dr. Robert R. Trippi
President
Sigma Research Associates
5666 La Jolla Boulevard, Suite l07
La Jolla, CA 92037

Telephone: (619) 459-2743

Submitted to *The Journal of Portfolio Management* October 20, 1991
Revised December 3, 1991

TRADING EQUITY INDEX FUTURES WITH A NEURAL NETWORK

Suggested subtitles:

Using machine learning to improve (enhance) investment performance

A machine learning-based system

A machine learning-enhanced trading strategy

Robert R. Trippi

Duane Desieno

Robert R. Trippi is Professor of Finance at California State University, Long Beach (90840).
Duane Desieno is president of Logical Designs Consulting, Inc. (La Jolla, CA 92037).

Contact:

Dr. Robert R. Trippi
President
Sigma Research Associates
5666 La Jolla Boulevard, Suite l07

La Jolla, CA 92037

Telephone: (619) 459-2743

Abstract

Neural networks are on the threshold of becoming assimilated into the mainstream of financial decision making. As a form of machine learning, this technology is especially attractive for developing trading systems which do not rely on human judgment. In this article we describe a neural network-based system for trading S&P 500 index futures that has, in testing, outperformed passive investment in the index. Our approach, the incorporation of several neural networks into a single composite Boolean decision rule system, demonstrates the potential to generate expected investment returns superior to any of the individual networks as well as the index.

Introduction

Neural network technology, which has been receiving increasing attention in the investment community, represents a radically different form of computation from the conventional algorithmic model. Neural networks consist of multiple simple processors arranged in a communicative network, each of which is programmed to perform one identical, elementary processing task. This technology is especially suited for simulating intelligence in pattern detection, association, and classification activities. Since such problems arise frequently in areas such as credit assessment, security investment, and financial forecasting, it is not surprising that after the Department of Defense, which in 1989 embarked on a five-year, multi-million dollar program for neural network research, financial services organizations have been the principal sponsors of research in neural network applications.

After a brief review of applications of potential interest to portfolio managers and neural network fundamentals, we will describe a specific neural network-based day trading system for Standard and Poors (S&P) 500 index future contracts that has, in ex-ante evaluation, outperformed passive investment in the index. This system, which is fairly representative of a neural network-based trading strategy implementation, also demonstrates how performance can be enhanced by integrating neural networks with conventional rule-based expert system techniques (e.g. see Lee, Trippi, Chu, and Kim (1990) and Trippi (1990)).

Background

Neural networks have proven effective in automating both routine and ad-hoc financial analysis tasks (Chithelin (1989)). Neural network-based decision aids have been developed for the following applications:

- *Credit authorization screening*
- *Mortgage risk assessment*
- *Project management and bidding strategy*
- *Economic prediction*
- *Risk rating of exchange-traded fixed-income investments*
- *Detection of regularities in security price movements*
- *Portfolio selection/diversification*

Other potential financial applications meriting further research, development, and evaluation include the following:

- *Simulation of market behavior*
- *Index construction*
- *Identification of explanatory economic factors*
- *"Mining" of financial and economic databases.*

Of greatest import to portfolio managers are neural network-based systems that can assist directly with risk assessment, asset selection, and timing decisions. Representative of such systems are those that have been built to

- generate improved risk ratings of bonds (Dutta and Shekhar (1988), Surkan and Singleton (1991), and Collins, Ghosh, and Scofield (1988))
- search for regularities in the price movements of an individual stock (White (1988))
- classify multiple stocks as to upside potential using fundamental and general economic data (Yoon and Swales (1991))
- recognize a specific price pattern, such as the Japanese "candlestick" triangle (Kamijo and Tanigawa (1990))
- determine optimal buy and sell timing for an equity index (Kimoto, Asakawa, Yoda, and Takeoka (1990))
- drive a trading strategy for a nonfinancial commodity index (Collard (1991)). [1]

How Neural Networks Learn

An individual processor within a neural network, simulated by specialized software for this purpose, is referred to as a *processing element* or PE. Each has one output but more than one input. Outputs of one PE become inputs to other PEs or outputs of the network. In most neural network paradigms, the actual processing that takes place within each PE is relatively simple: taking a weighted sum of the inputs and calculating an output value which is a function of that sum, as follows:

$$x_j = T\left(\Sigma_i w_{ij} x_i\right),$$

where x_j is the output of processing element j, w_{ij} is the weighting coefficient of the interconnect

link between processing elements i and j, and T is a *transfer* or *activation function*. The most commonly used transfer functions are variations of the S-shaped sigmoid,

$$(1\text{-}e^{-u)})^{-1}.$$

However, other functions, including Z-shaped and threshold detectors, are also used in some applications. The PE's local memory stores interconnect weights and parameters used by its transfer function. When many processors are linked, a neural network is created.

Learning in a neural network takes place by incremental changes in interconnect weight coefficients as new examples are run through the system during the training phase. For example, under the *delta learning rule* the change in interconnect weight Δw_{ij} is made proportional to the difference between the desired output o_j and the actual output x_j:

$$\Delta w_{ij} = k(o_j\text{-}x_j)x_i,$$

where x_i is the level of the input to PE j from PE i and k is a constant which determines the learning rate. The delta rule is somewhat analogous to gradient or path of steepest descent optimization, as the changes made lead to local improvement only. There are about a half-dozen different learning rules in common use.

The learning rules that can be employed depends to some extent on the pattern of interconnections among the PEs. There are about twenty different network configurations, about a dozen of which are commonly employed. In most of these, individual networks are combined in layers. Usually there is an input layer, an output layer, and one or more intermediate or hidden layers.

A Trading System for S&P 500 Futures

The S&P 500 trading system consists of several trained neural networks plus a set of rules for combining the networks' results to generate a composite recommendation for the current day's position. The trading strategy is time invariant; that is, whatever trades are executed are done so at a fixed time during the trading day. Specifically, the system enters the market 15 minutes after its opening and unwinds its positions 25 minutes before the close of trading for that day, unless a stop has previously been hit.

The general network architecture is a feed-forward network such as that shown in Figure 1, where the lines indicate *connections* along which processing element outputs are transferred to the inputs of other processing elements.

Inputs to the network are primarily technical variables for the two week period prior to the trading day. This differs substantially from the inputs used in the Collard (1991) commodity trading system, which were mostly exogenous or fundamental. Inputs include Open, High, Low and Close price information, plus several statistics derived from past price data, including recent volatility. The only real-time information used from the current trading day are the opening price and the price 15 minutes after the market opening.

The output from each network is a Long or Short recommendation. A trailing stop controls losses and protects gains. For the initial evaluation of the system, a constant 2.5 point stop was used; however, the optimal stop size appears to be a function of market volume, with smaller stops performing best when volume is light.

The system was designed to be run overnight to generate an input-output decision map for the next day for each network, such as that in Figure 2. To exploit possible synergism among networks, six

differently configured networks, labeled Net1 through Net6, were trained using the same historical data. Figure 3 shows a typical daily decision map for a composite trading decision rule set which combines the outputs of several networks. Both figures, with Short recommendations represented by darker regions, show complex nonlinearity of decision map partitions.

Training Individual Networks

In training the networks, price data from the front (most actively traded) contract was employed. The training period spanned 1168 days, from January 1986 to June 1990. This is not an exceptionally large training set compared those used in other neural network applications. The minimum training set size required for typical classification tasks has been estimated by Baum and Haussler (1989) to be N/I, where N is the number of weights in the network and I is the fraction of test examples incorrectly classified. With a target 10% error level the network would thus have no more than 116 weights, which is far too few for a problem of this complexity. Thus with the data available, we were willing to accept a network performance on training data with a higher error level as well as somewhat sensitive to the initial weights.

The test period covered 106 days from 12/20/90 through 5/31/91. With the exception of Net1, the only differences in the trained networks were their initial randomly-generated starting weights and the length of training. Net1 had a slightly different input configuration, excluding the current index price as an input.

The performance of the individual networks is summarized in Table 1. Trades of a single contract were assumed to be executed every day. Maximum drawdown and a $5000 margin were used to calculate the geometric mean return on investment over the test period and a round turn commission

of $60.00 was charged to each transaction. As mentioned above, to minimize the uncertainties of slippage, for the purposes of system evaluation we assumed transactions to be executed at a certain time of day rather than at a specific price.

Net6 represents a special case created to examine the effects of overtraining or overfitting. This network was deliberately overtrained, enabling it to achieve a relatively low 10% error rate on the training data. This ex-post optimized network would have generated nearly twice the net gain over the training period in comparison to the other networks. However, as indicated in Table 1, its ex-ante performance over the subsequent test period was worse than the other networks. Nevertheless, this network proved to be of value as a tie-breaker in the composite trading strategy to be discussed next.

Generation of Composite Trading Decision Rules

In the system developed by Kimoto et al (1990) for prediction of the TOPIX index, several different networks were trained and their outputs averaged in order to achieve a higher correlation with the training data. For our trading system, the outputs of individual networks are combined through the use of logical (Boolean) operators, to produce a set of composite rules. This process is called *rule synthesis*.

There are two possible benefits from rule synthesis. First, by using the outputs of more than one network, it is possible to more effectively recognize days in which there is genuine ambiguity, and therefore best not to trade. Second, by combining rules it may be possible to generate a composite decision rule set that is more consistently profitable than even the best of the individual networks.

On a typical day a number of networks will agree in their recommendations. For example, Net1

through Net5 were in agreement 30 of the test days. If a trade had been made only on those days, the expected value of the trade would have been $549, which is 1.9 times the expected value from simply using the best individual net. However, if trading could take place only when all networks were in agreement, it would have occurred only 26 percent of the time, so the gain over the period would have been limited to just $16,470.

The composite rule generation procedure examines the results of all possible combinations of networks on trading days where no rule yet applies. If the expected value of trades resulting from applying one or more of these newly synthesized rules is greater than that of the best individual network (in this case Net2), then the best rule is added to the current rule set. This process is continued until there are no more days are left unexamined or the rules become very complex (exceeding five logical operations).

Table 2 shows the first rule set that was developed, Composite1, which includes Net1 through Net5 and is designed to be scanned sequentially. Table 3 shows the performance of the individual rules on the days that they applied or "fired". The performance of a composite trading system using these rules in combination exceeded that of the best single network (Net2) by 21%. The combined system trades 84 out of 106 days or 79% of the time.

Using the rule synthesis procedure above, we examined whether Net6 could in any way improve the composite system performance. This resulted in the rule set Composite2, also shown in Table 2, in which rules 5, 6 and 7 are replaced with the new rules 5a and 6a. This rule set trades every day and has a per-trade expected value of $460 and cumulated gain of $48,750 over the initial test period.

It should be noted that the use of any data to determine the choice of network, setting of parameters, length of training, or use of networks in combination redefines that data as training data, and makes

it inappropriate for use in further testing. Thus, although the composite decision rule set performed better than did individual rules, it was over a period that had been used previously to generate the individual network weights. Therefore it was necessary to further test the composite rules using newer data. The results of this testing are discussed next.

Evaluating the Composite Decision Rule Set

In testing any index trading strategy it is desirable to determine whether it outperforms both the index and a purely random trading strategy applied to the index over the same time period. Our approach to the former was to regress account gain/loss against time for the Composite2 rule set and compare the results with a regression on the index. Both will have just a single free parameter, slope, since initial gain or loss at the beginning of the test period is zero. The slope can be interpreted as the expected value of a single trade when exactly one trade is made per day. Figure 4 shows the regression lines for the S&P 500 and Composite2 (labeled System Value), the latter with its 99% confidence limits on either side. The expected gain was $317.06 per day, which is significantly different at the 99% confidence level from the $139.06 per day index trend over this time period (Note 2).

Using the Composite2 rule set, both actual and simulated trades of the September S&P 500 contract were made over the period from June 24 through August 16, 1991, showing a cumulative gain of $14,247. For the 39 days that trading took place, there were gains on 24 days, or 61.5% of the time. The average gain was $961.23 and the average loss was $588.15. Note that the S&P 500 index value does not include commissions, while an $81.49 actual commission per contract was deducted in evaluating the Composite2 rule set.

To determine performance relative to a random trading strategy, we simulated 100,000 sequences of intraday positions over the same period. The percentage of sequences which would have outperformed the Composite2 rule set can be interpreted as the probability of Composite2's performance being achievable by randomly going long or short. The frequency distribution of the simulation run outcomes appears in Figure 5. The probability of the random trading strategy meeting or exceeding the Composite2 rule set performance is .01. In other words, we are 99% confident that Composite2 will outperform randomly generated long and short day positions.

Summary and Conclusions

This article has reviewed uses of neural networks in investment management and outlined a specific neural network-based intraday trading system for S&P 500 future contracts. This system, which combines several trained neural networks, appears to outperform each of its individual networks and the index. As a point of reference, the best network in Collard's system produced a gain of $10,301 over a year of trading, while our best composite synthesized rule set system achieved a gain of $14,247 in just 8 weeks. With an expected return of $317 per day, the potential annual return is $60,000 per contract.

The system was deliberately limited to trading at specified times during each trading day to facilitate performance evaluation. It is likely that performance in actual trading would be improved by varying the timing of the trades. It should be noted also that, for the purpose of evaluation, trading was forced every day. In normal use, when significant news events impact the market, trades could be suspended until the market stabilizes.

Further development of neural network-driven trading systems will likely follow two general

directions. The first is to apply existing successful system designs to other types of assets. The second direction is to attempt the improve such systems by designing the networks to operate "on-line" in real time. We expect production versions of such systems to be commonplace within the next few years.

Notes

1.These and other papers on the use of neural networks in investment management may be found in Trippi and Turban (1992).

2.Since the return variances were almost identical, it was unnecessary to explicitly consider differential risk levels. This is to be expected, as in the long run a time-invariant intraday trading strategy such as this would tend to produce a return variance identical to that of the index itself regardless of whether intraday positions were always long, always short, or randomly long and short.

References

Baum, E. B., and D. Haussler. "What Size Net Gives Valid Generalization", *Neural Computation 1*, Cambridge: MIT Press, 1989, pp. 151-160.

Chithelin, I. "New Technology Learns Wall Street's Mindset," *Wall Street Computer Review*, June 1989, pp. 19-21.

Collard, J. E. "A B-P ANN Commodity Trader", in R. P. Lippmann, J. E. Moody, and D. S. Touretzky (eds.), *Advances in Neural Information Processing Systems 3*, San Mateo: Morgan Kaufmann, 1991.

Collins, E., S. Ghosh, and C. Scofield. "An Application of a Multiple Neural Network Learning System to Emulation of Mortgage Underwriting Judgments," *Proceedings of the IEEE International Conference on Neural Networks*, July 1988, pp. II-459-466.

Dutta, S. and S. Shekhar. "Bond Rating: A Non-Conservative Application of Neural Networks." *Proceedings of the IEEE International Conference on Neural Networks*, July 1988, pp. II-443-450.

Kamijo, K. and T. Tanigawa. "Stock Price Pattern Recognition: A Recurrent Neural Network Approach," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, IEEE Neural Network Council, 1990, Vol. I, pp. 215-221.

Kimoto, T., K. Asakawa, M. Yoda, and M. Takeoka. "Stock Market Prediction System with Modular Neural Networks," *Proceedings of the International Joint Conference on Neural Networks*, San Diego, IEEE Network Council, 1990, Vol. I, pp. 1-6.

Lee, J. K., R. R. Trippi, S. Chu, and H. Kim, "K-Folio: Integrating the Markowitz Model with a Knowledge-Based System," *The Journal of Portfolio Management*, 16 (Fall 1990), pp. 89-93.
Surkan, A. and J. Singleton. "Neural Networks for Bond Rating Improved by Multiple Hidden Layers," *Proceedings of the IEEE International Conference on Neural Networks*, July 1991, pp. II-157-162.

Trippi, R. and E. Turban (eds). *Neural Networks in Financing and Investing: Applying Artificial Intelligence to Improve Real-World Performance*, Chicago: Probus Publishing Co., 1992.

Trippi, R. "Intelligent Systems for Investment Decision Making," in *Managing Institutional Assets*, F. Fabozzi, ed., New York: Harper & Row, 1990.

White, H. "Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns." *Proceedings of the IEEE International Conference on Neural Networks*, July 1988, pp. II-451-458.

Yoon, Y. and G. Swales. "Predicting Stock Price Performance: A Neural Network Approach," *Proceedings of the 24th Annual Hawaii International Conference on Systems Sciences*, Hawaii, IEEE Computer Society Press, 1991, Vol. 4, pp. 156-162.

Network	#Gains	#Losses	%Gains	Avg Gain	Avg Loss	#Stops	Max Drawdown	Total Gain	Expected Value	%ROI
Net1	49	57	46.21	320.10	786.31	67	8195.00	19865.00	187.41	150.5
Net2	56	50	52.81	261.43	803.00	64	8620.00	30490.00	287.64	223.9
Net3	51	55	48.11	303.24	786.82	62	4985.00	23190.00	218.77	232.2
Net4	54	52	50.91	241.85	836.92	61	7830.00	23540.00	222.08	183.5
Net5	54	52	50.91	221.48	804.23	63	9880.00	24140.00	227.74	162.2
Net6	48	58	45.31	194.17	851.81	69	14485.00	7915.00	74.67	40.6

Table 1. Performance of Individual Networks

Composite1

Rule 1: If all nets agree, make the indicated trade.
Rule 2: If Net1 and Net2 agree and one of Net3, Net4, or Net5 disagree, then follow Net1.
Rule 3: If Net1 and Net2 agree and two out of Net3, Net4, or Net5 disagree, then follow Net1.
Rule 4: If Net1 and Net2 agree and the others disagree, then follow Net3.
Rule 5: If Net2 and Net3 agree and the others disagree, then do not trade.
Rule 6: If Net1 and Net2 disagree and the others have the same decision, then do not trade.
Rule 7: If Rules 1 through 6 do not apply, then follow Net2

Composite2

Rule 1: If all nets agree, make the indicated trade.
Rule 2: If Net1 and Net2 agree and one of Net3, Net4, or Net5 disagree, then follow Net1.
Rule 3: If Net1 and Net2 agree and two out of Net3, Net4, or Net5 disagree, then follow Net1.
Rule 4: If Net1 and Net2 agree and the others disagree, then follow Net3.
Rule 5a: If Net2 and Net3 agree and the others disagree, then do the opposite of Net6
Rule 6a: If Rules 1 through 5 do not apply, then follow Net6

Table 2. Rules for Combining Outputs of Networks

Rule	#Days Fired	Expected Value	Gain	Cumulative Gain
1	30	549.00	16470.00	16470.00
2	20	388.75	7775.00	24245.00
3	12	606.67	7280.00	31525.00
4	7	665.00	4655.00	36180.00
5	8	-	-	-36180.00
6	12	-	-	-36180.00
7	17	41.17	700.00	36880.00

Table 3. Performance of Rules 1 - 7

Figure 1. Basic Neural Network Feed-Forward Model

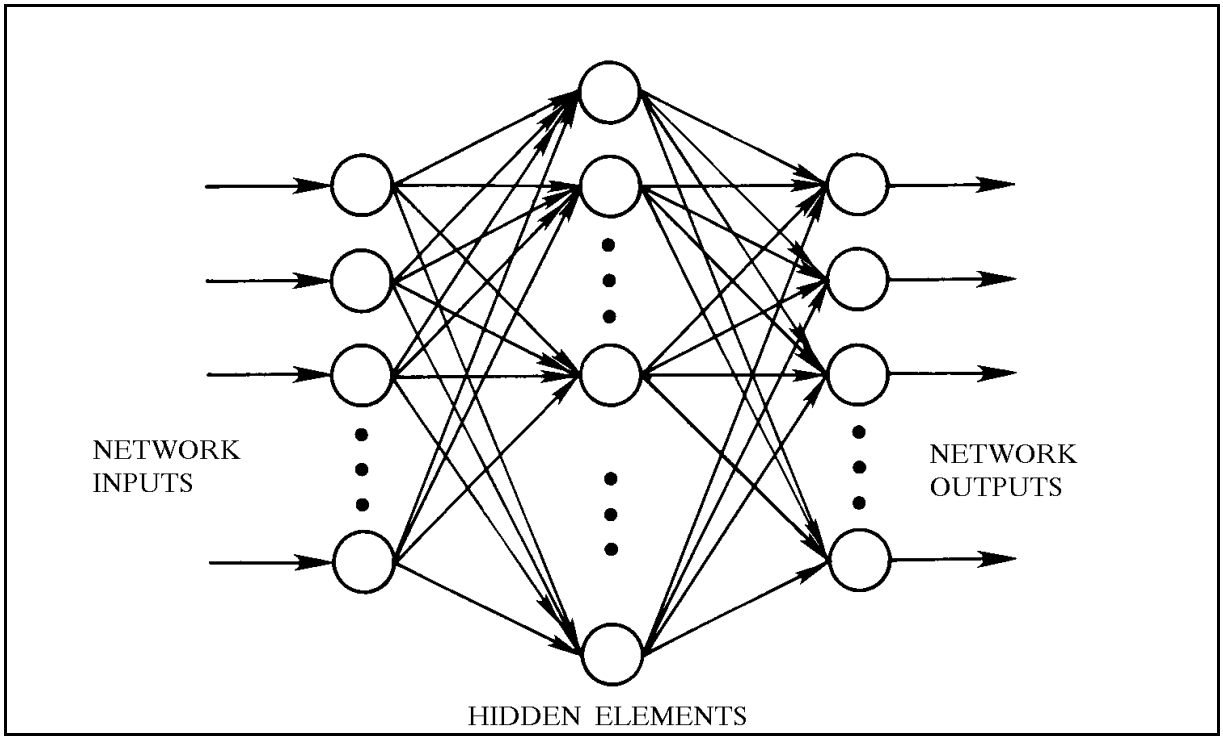

Figure 2. Typical Net2 Decision Map

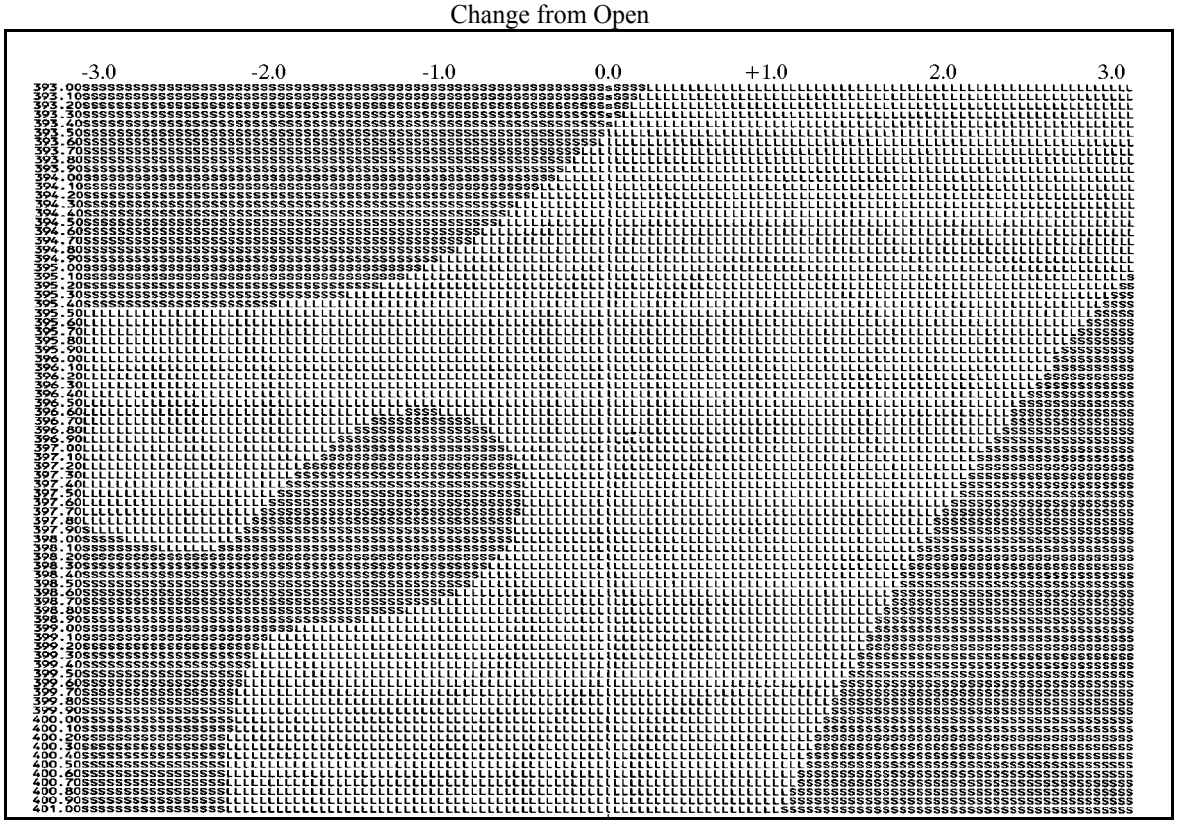

Figure 3. Typical Composite Rule Set Decision Map

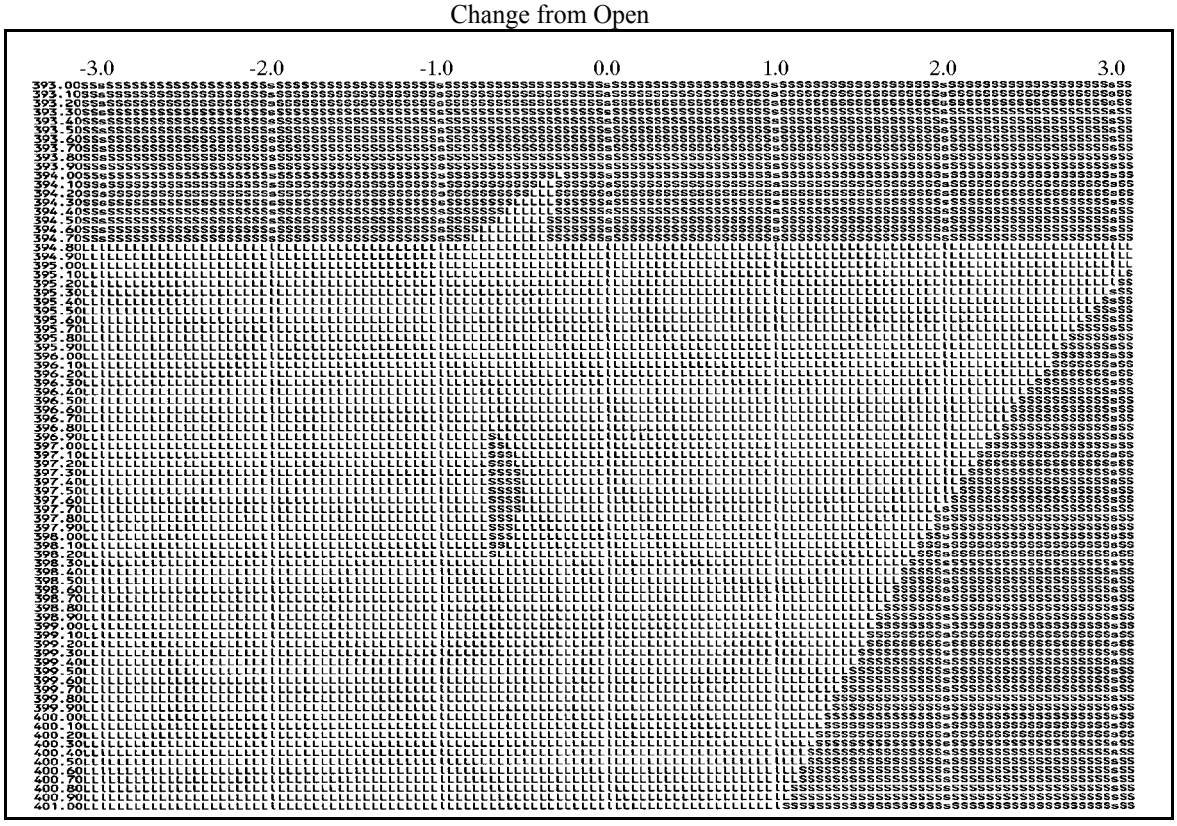

Figure 4. S&P 500 Index versus Composite2 System Performance

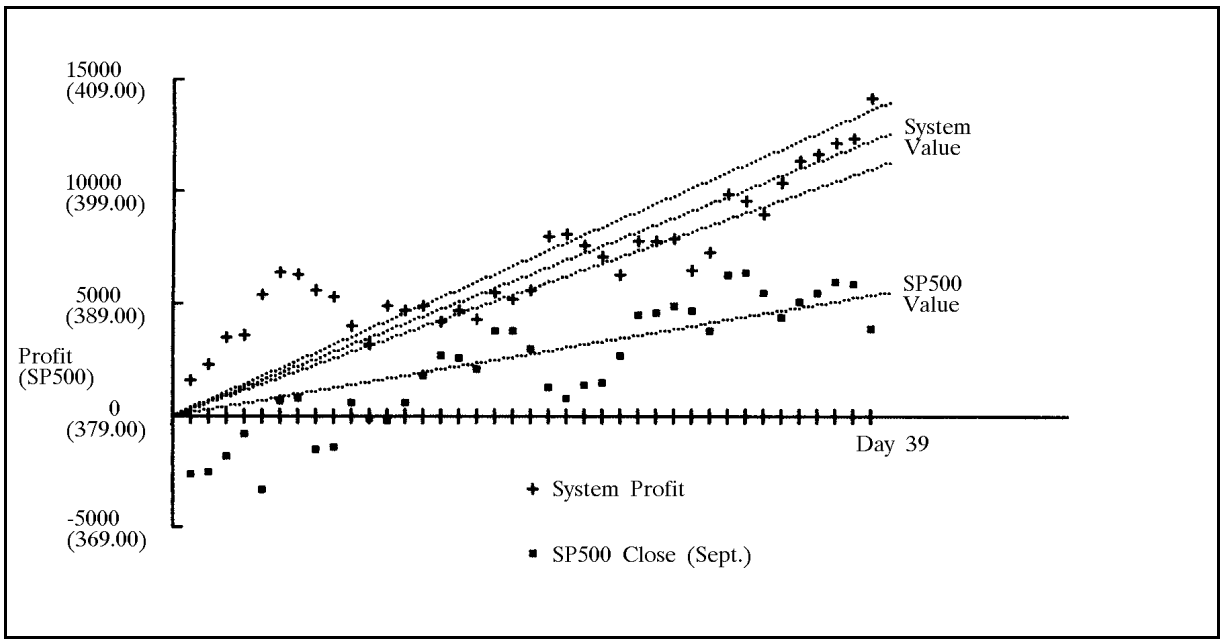

Figure 5. Gain/Loss from 100,000 Randomly Generated Intra-day Index Trading Sequences
(Composite2 = $14,200)