ELSEVIER

Decision Support Systems 32 (2002) 361–377

Decision Support Systems

www.elsevier.com/locate/dsw

Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: a case study in *romantic* decision support

William Leigh [a,*], Russell Purvis [b], James M. Ragusa [a]

[a] *Department of Management Information Systems, College of Business, University of Central Florida, Orlando, FL 32816 1400, USA*
[b] *Clemson University, Clemson, SC, USA*

Received in revised form 1 February 2001; accepted 1 August 2001

Abstract

The 21st century is seeing technological advances that make it possible to build more robust and sophisticated decision support systems than ever before. But the effectiveness of these systems may be limited if we do not consider more eclectic (or *romantic*) options. This paper exemplifies the potential that lies in the novel application and combination of methods, in this case to evaluating stock market purchasing opportunities using the "technical analysis" school of stock market prediction. Members of the technical analysis school predict market prices and movements based on the dynamics of market price and volume, rather than on economic fundamentals such as earnings and market share. The results of this paper support the effectiveness of the technical analysis approach through use of the "bull flag" price and volume pattern heuristic. The *romantic* approach to decision support exemplified in this paper is made possible by the recent development of: (1) high-performance desktop computing, (2) the methods and techniques of machine learning and soft computing, including neural networks and genetic algorithms, and (3) approaches recently developed that combine diverse classification and forecasting systems. The contribution of this paper lies in the novel application and combination of the decision-making methods and in the nature and superior quality of the results achieved. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Technical analysis; Neural networks; Forecasting; Genetic algorithms; Pattern recognition; Heuristics; Financial decision support; Market efficiency

1. Introduction

This new century opens on an unprecedented availability and selection of development tools for building decision support systems [4]. These tools have reduced the complexity and long development time inherent in building systems that offer valuable insights into the complex problems offered in today's business world. But will these technological enhancements manifest into systems that exploit the vast opportunities that are now available?

This century also promises to be a time of discontinuous and increasingly rapid change, with new risks taking the place of ones we understand. Time pressures and the rush of events will require that

* Corresponding author. Fax: +1-407-823-5741.
E-mail address: leigh@pegasus.cc.ucf.edu (W. Leigh).

PII: S0167-9236(01)00121-X

decision support tools be used in an efficient, unified and adaptive manner. This will require satisficing with good results, often without understanding "scientifically" the underlying decision contexts that we analyze. Decision support systems, however, cannot meet these opportunities without changing the way the systems are approached and built. Indeed, a bolder, more eclectic style will be necessary, which we term *romantic*.

Classical connotes beauty of form, good taste, restraint, and clarity. *Romantic* is extravagant, wild, free, imaginative, and fantastic, and is in revolt from that which is classical. We contend that in the early 21st century, the classical style in decision support practice will be supplanted by a more romantic style. A romantic style for decision support combines seemingly disparate sets of theories, data, and techniques, uses radically new data analysis and machine learning techniques, which employ nonlinear and connectionist models, realizes systems through the assembly of powerful hardware and software components, combines tools and techniques to develop hybrid decision-making mechanisms, may involve assaults on the accepted wisdom of decision-making, and achieves superior results.

The work in this paper is empirical. We cannot explain the results attained here, or the results claimed by practitioners for technical analysis, with any strong theoretical basis. The romantic style is pragmatic and values decision-making results over understanding and values data over theory. The romantic style is data-driven, rather than theory-driven (but, of course, there is nothing more pragmatic than a bit of good theory at the right time). Diebold discusses this tension, which is more palpable in some firmly ensconced branches of academia, including finance, than in industry (page 589, Ref. [7]):

> There is a long and unfortunate tradition in economics, however, of placing too much emphasis on theory—as opposed to evidence (data)—in guiding the specification and evaluation of forecasting models. Within such a framework, one is led to focus on the task of estimating the parameters of a relationship suggested by a priori theory (at the cost of neglecting the model selection problem), after which the optimal prediction problem is easily solved (conditional upon the assumed model).

Winkler agrees (page 606, Ref. [53]):

> I prefer, however, to take the view that, in many situations, there is no such thing as a 'true' model for forecasting purposes. The world around us is continually changing, with new uncertainties replacing old ones. As a result, the longer-term search for a 'true' model is doomed to fail in many cases because unanticipated changes prevent us from enjoying the luxury of getting to the longer term in a stable environment. This suggests that models should be adaptive, but even adaptive models only represent our best state of knowledge at a given time; they do not represent the 'truth' in any sense.

Decision support systems, as a discipline, owes much to artificial intelligence, which has been predominantly pragmatic and data-driven from the beginning, and as such has always evidenced the attributes of what we are calling *romantic*, to a degree. But the need for adaptability and the achievement of results is more compelling in the 21st century than before, and the opportunity is greater than before. The new elements that are present now are the methods and techniques of soft computing and machine learning for decision-making and forecasting, in particular neural networks and genetic algorithms, and the new science of the combination of those tools. These new elements make possible decision support systems that are more *romantic* than those which have gone before to an extent that by comparison, the decision support systems of today appear *classical*.

The way in which different decision-making and forecasting methods are combined is critical. The objective is for a coordinated use of different methods to arrive at a better decision than the employment of any of the methods alone. There is reason to believe that this not overly difficult to achieve (for the case of classification algorithms, see Ref. [2], and for forecasting, see Ref. [34]); and that greater diversity in the decision-making methods which are combined leads to better combined decisions (for the case of classifier systems, see Ref. [52]). The development and use of effective methods of decision tool integration and combining are critical to realizing successful decision support systems of this romantic style. We explore, exemplify, and survey such combining methods in this paper.

In the following sections we describe four experiments that focus on stock market price forecasting—each experiment building on the results of the previous ones. The first experiment, Experiment A, employs the conventional pattern recognition technique of template matching to identify one of the basic price–volume pattern market direction signals of technical analysis. We compare the change in level of the New York Stock Exchange Composite Index after the trading day, which is identified as marking the end of the pattern, when it is identified in time series data, with the average change in level of the same index over all trading days in the period of comparison. Experiment B uses the template fitting values of the identified instances of the technical pattern as a basis for forecasting the level of the NYSE Composite Index with a neural network model. Experiment C employs a genetic algorithm to improve the forecasting quality of the neural network forecast in terms of its squared multiple correlation R^2. Experiment D applies the methods of Experiments A and B in a cross-validation experimental design. The system in Experiment D is deployable for decision support.

Two of the techniques we use, neural networks and genetic algorithms, have come into use in the decision support arsenal only in the last 10 years. The experiments are accomplished using a desktop computer and commercially available spreadsheet, neural network, genetic algorithm, and statistical software. No general-purpose-language programming is needed. The theoretical basis for the series of experiments conducted is stock market *technical analysis*, which is presently considered disreputable in the academic finance community. However, excellent results are achieved—romantic decision support is pragmatic and does not respect accepted wisdom. As a result of the approach taken, and the results achieved, we believe that this work exemplifies the romantic style of decision support for financial forecasting.

2. Technical analysis

The generally accepted efficient markets hypothesis (EMH), explained and surveyed by Fama [14], states that market prices follow a random walk and cannot be predicted based on their past behavior. According to the EMH there are three degrees of market efficiency. The *strong form* states that *all information that is knowable* is immediately factored into the market's price for a security. If this is true, then all of those stock analysts are definitely wasting their time, even if they have access to private information. In the *semi-strong form* of the EMH, *all public information* is considered to have been reflected in price immediately as it became known, but possessors of private information can use that information for profit. The *weak form* holds only that any *information gained from examining the security's past trading history* is immediately reflected in price. Of course, the past trading history is public information, which implies that exceptions and counter-examples to the weak form also apply to the strong and semi-strong forms.

Discoveries of "anomalies", relationships that can be used to earn abnormal returns, which appear to violate the EMH in its strong and semi-strong form are numerous in the finance literature. Well-known anomalies involve abnormal returns in relation to: unexpected earnings announcements, firm size, month of January, day of the week, analysts' recommendations, impact of the federal budget deficit announcement, and others. Raghubir and Das [42] catalog anomalies and provide extensive literature references. Also, see Refs. [22,23].

Even though they are both futile endeavors according to the EMH, two approaches to the analysis of stock market price prediction dominate practice: *fundamental* analysis and *technical* analysis. These approaches differ in their underlying assumptions. Fundamental analysis accepts the weak form of the EMH and ignores the semi-strong form. Fundamental analysts assume that prices in financial markets are based on economic principles, and prices may be predicted based on fundamental and publicly available economic data, such as earnings and market share, interest rates, cost trends, competitive forces, and so forth. Graham and Dodd [19] wrote the classic guide to fundamental analysis of investments years ago. On the other hand, those who practice technical analysis accept most tenets of the semi-strong form of the EMH (that is, that available knowledge of the economic fundamentals and market conditions affecting a particular investment are available to all and have been factored into the current stock market price) and ignore the weak form of the EMH. Technical analysts are concerned with the dynamics of the market price and volume behavior itself, rather than with the fundamental economic nature of specific securities that are traded. Charles Dow developed the original Dow Theory for technical analysis in 1884, and a modern explication is found in Edwards and Magee [12].

The EMH, in particular its weak form, is generally interpreted to imply that the technical approach to price prediction is invalid. Papers on technical analysis appear frequently in the practitioner literature (for example Refs. [36,38]) but, until recently, rarely in the academic literature except in negative or defensive form (for example Ref. [48]). In the last 2 years major, positive reports as to the effectiveness of "momentum trading rules", which are based only on price and volume historical information (but do not involve price and volume patterns which we investigate in this article), have appeared more than once in the most respected finance journals (for example Refs. [22,23]). These momentum anomalies appear to negate the EMH in its weak form, though there is a debate going on over whether this is true [3], and the academic acceptance of the effectiveness of momentum trading effectiveness possibly marks a change in the way technical analysis research work will be accepted.

Our work uses the weak form of the EMH as a null hypothesis. We test our forecast against the overall average 20-day horizon price increase experienced in the period we are using. For our work with a broad-based composite index, the overall average in the period is equivalent to the return from a buy-and-hold or random-selection trading strategy, which are implied as optimal by the weak form of the EMH. We consider the discovery of a forecasting method that can be applied as a trading strategy yielding statistically significant returns that are better than the overall average for the period to be a failure to confirm our null hypothesis, which is the weak form of the EMH.

Gencay [16] uses only price history and applies neural networks to implement moving-average forecasting rules, which are momentum rules, to predict the Dow Jones Industrial Average. Results are not reported in such a way as to be able to tell whether a trading rule based on the methods in the paper would do better than buy-and-hold or random purchases, and there is no way to compare the results of Gencay with our work. Gencay's major result is that the nonlinear, neural network implementations of the moving-average rules consistently outperform the linear regression implementations by a modest amount.

We work with one price and volume pattern heuristic of technical analysis. We have found no rigorous testing of this pattern or other pattern heuristics of technical analysis anywhere.

3. Experiment A: recognizing the *bull flag* with pattern recognition

The sort of technical analysis that we use is based on the identification of certain graphical patterns of price and volume time series data to identify buy (and sell) signals. Our work concentrates on one technical analysis pattern, the *bull flag*. The definition of *flag* from Downes and Goodman [9]:

> FLAG—technical chart pattern resembling a flag shaped like a parallelogram with masts on either side, showing a consolidation within a trend. It results from price fluctuations within a narrow range, both preceded and followed by sharp rises or declines.

A bull flag pattern is a horizontal or sloping flag of *consolidation* followed by a sharp rise in the positive direction, the *breakout*.

The template we use for the bull flag pattern is shown in Fig. 1. This is a 10-by-10 grid with weights ranging from -2.5 to $+1.0$ in the cells. The weighting is used to define areas in the template for the descending consolidation and for the upward-tilting breakout portions of this bull flag heuristic pattern.

The 10×10 grid is applied to the time series of price or volume data one trading day at a time, with the leftmost time series data point being the values for the trading day which precedes the current day by 59 trading days, and the rightmost time series data point being the trading day which is currently being analyzed. Values for the earliest 10% of the trading days (6 days of the 60 in the rolling window) are mapped to the first column of the grid, values for the next-to-earliest 10% of the trading days are mapped to the second column of the grid, and so on, until the most recent 10% of the trading days are mapped to the rightmost column.

.5	0	-1	-1	-1	-1	-1	-1	-1	0
1	.5	0	-.5	-1	-1	-1	-1	-.5	0
1	1	.5	0	-.5	-.5	-.5	-.5	0	.5
.5	1	1	.5	0	-.5	-.5	-.5	0	1
0	.5	1	1	.5	0	0	0	.5	1
0	0	.5	1	1	.5	0	0	1	1
-.5	0	0	.5	1	1	.5	.5	1	1
-.5	-1	0	0	.5	1	1	1	1	0
-1	-1	-1	-.5	0	.5	1	1	0	-2
-1	-1	-1	-1	-1	0	.5	.5	-2	-2.5

| Consolidation | Breakout |

Fig. 1. *Bull flag* template used in this study. The first seven columns represent a *consolidation* and the last three columns represent a *breakout*.

We fit the 10×10 grid to a rolling window of 60 trading days at a time. The vertical fitting process is adaptive: the highest value in the window is made to correspond with the top of the grid, and the lowest value in the window is made to correspond with the bottom of the grid. The percentage of values that fall in each cell of a column is multiplied by the weight in the corresponding cell of the bull flag template (a cross-correlation computation).

For example: There will be 6 trading days represented in each column of a single 60 trading day window. If all 6 of these trading days have price values which are in the lowest decile of the 60 price values for the day, then 100% (6 values out of a total of 6 in the column) will be the value in the lowest cell of the 10 cells in the column. If this column is the leftmost of the columns in the window, then this 100% will be multiplied by the value in the corresponding cell in the bull flag template (which is the one in the lowest left-hand corner), which has the value of -1.0 (See Fig. 1), to result in a cell fit value of $-1.0 \times 100\% = -1.0$. This is done for the 10 cells in the column and summed, resulting in a fit value for the column of -1.0, since there will be 0.0% in the other nine cells of the column.

In this way, 10 column fit values for price and 10 column fit values for volume are computed for each trading day. Summing all 20 values for a trading day results in a total fit for the trading day. (This process is an example of *template matching* as described in Duda and Hart [10]).

Results of this template fitting process, implemented with Microsoft's Excel spreadsheet tool, for price and volume data for the New York Stock Exchange Composite Index for the period January 1, 1981 to December 31, 1996, appear in Fig. 2. On this scatter-plot a point represents each trading day in the period. The horizontal axis is the total fitting value for the trading day point (the sum of the 10 column fit totals for price and the 10 column totals for volume for the time series of 60 days including and preceding the trading day), and the vertical axis shows the percentage of increase (or decrease) in the NYSE Composite Index price observed 20 trading days after that trading day.

The third-order polynomial trend-line plotted on Fig. 2 indicates that the higher total fitting values correlate somewhat with higher price increases. Fig. 3 isolates the total fit and 20-day price change values for the trading days, which have total fit values at

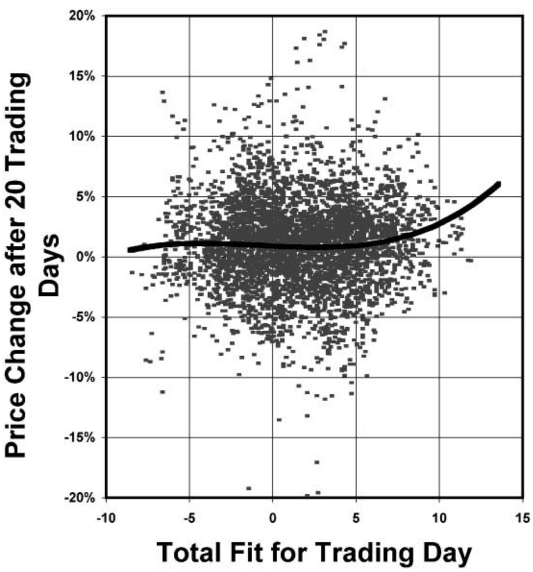

Fig. 2. Scatter-plot of total fit for each trading day versus price change after 20 days with third-order polynomial trend-line.

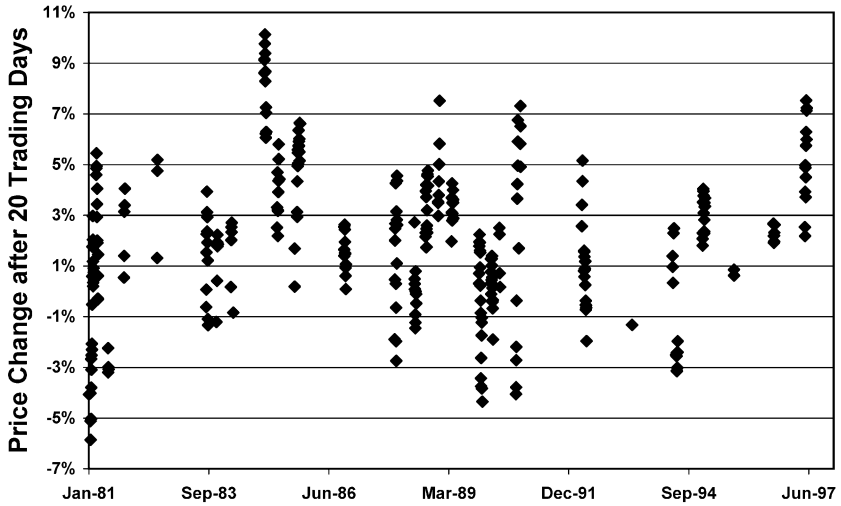

Fig. 3. Ninety-percentile and better trading days by date showing the price change 20 trading days afterward.

the 90-percentile level or better (which we will designate as our identified bull flags.) It may be seen from Fig. 3 that trading days meeting this 90-percentile or better criteria are spread throughout the period under study.

Note that *the percentile for total fit is calculated on a running basis*, that is, the percentile for the 100th trading day, for example, is calculated relative to the set of the first 100 trading days, and the percentile for the 345th trading day is calculated relative to the set of the first 345 trading days, and so forth. Calculation of percentile in this way uses only prior knowledge, which makes this analysis applicable to the development of forecasting and trading rules.

Table 1 contains statistics for a comparison of the complete set of data with the subset of the data having at least 90-percentile total fit values, which we use as our identified bull flags for Experiment B and Experiment C. The statistics are computed in two sections: for the trading days before 1994 and for the trading days from 1994 to the end of 1996. (We establish these two sections as a training sample and a holdout test sample for subsequent experiments. The break between the training and test samples was selected to have enough data points in each set for meaningful significance testing). Table 1 indicates that a selection of trading days based on the 90-percentile or better value results in significantly higher price changes than selection of trading days at random, and this result is true for the pre-1994 set and for the 1994-later set. *T*-test significance for the comparison of 90-percentile fit or better with all of the days in the sample is 0.0000 for the pre-1994 data and 0.0052 for the 1994-later data, and this indicates a failure to support the null hypothesis implied by the weak form of the EMH that prices cannot be predicted through use of price and volume data alone.

Table 1
Comparison of mean price increase for 90th percentile and better trading days to all trading days in training sample and in holdout test sample

	Pre-1994	1994-Later
For all trading days in period:		
Mean price increase after 20 trading days	0.84%	1.35%
Standard deviation	0.0423	0.0296
Number of trading days in period	3289	1011
For identified bull flag trading days (90-percentile or better total fit):		
Mean price increase after 20 trading days	2.00%	2.36%
Standard deviation	0.0297	0.0268
Number of trading days meeting criteria	264	52
Comparison:		
Difference in means	1.15%	1.02%
Significance[a]	0.0000	0.0052

[a] For one-tailed, two-sample unequal variance *t*-test on means in column.

4. Experiment B: forecasting price with neural networks

Neural networks are nonparametric, nonlinear models that can be trained to map past values of a time series, for purposes of classification or function estimation. We use a *feedforward* neural network with *backpropagation* learning, which is the most conventional sort of neural network. The feedforward neural network computes input-to-output mappings based on calculations occurring in a system of interconnected nodes, which are arranged in layers; the output of each node is calculated as a nonlinear function of the weighted sum of inputs from the nodes in a layer which precedes it in computation order. The process of back-propagation learns the weights for the connections between the nodes through training from data, resulting in a minimized least-mean-square error measure between the actual, desired values and the estimated values from the output of the neural network. Seminal work in neural networks includes Refs. [25,44]. Neural network principles and practices in the financial context are discussed in many places (for example Refs. [1,11,24,37,40,43,50,54]); and forecasting is a common application [26,55]; and neural networks are used in decision support systems [21, 46,49]. Several researchers report modest, positive results with the prediction of market prices using neural networks [20,28,45], but not by using price and volume histories alone, and no one uses technical analysis pattern heuristics.

Our network configuration consists of 22 input nodes, 1 hidden layer with 6 nodes, and 1 output node. The input nodes correspond to the 20 column fitting values (10 for price and 10 for volume) and 2 window height values (the difference between the lowest price and the highest price in the 60-day price window, and the corresponding for volume.) The single output node supplies the prediction of the future price at a 20-trading-day forecasting horizon. We normalize input values to the neural network parameter learning using a zero-mean unit-variant (zscore) procedure. The neural network software used is the Pattern Recognition Workbench, developed and marketed by Unica Technologies (Lincoln, MA) and documented in Ref. [27]. Other parameter settings used were the product's default values. Fig. 4 is a block diagram showing the relationship between the rolling window template fitting mechanism and the neural network system.

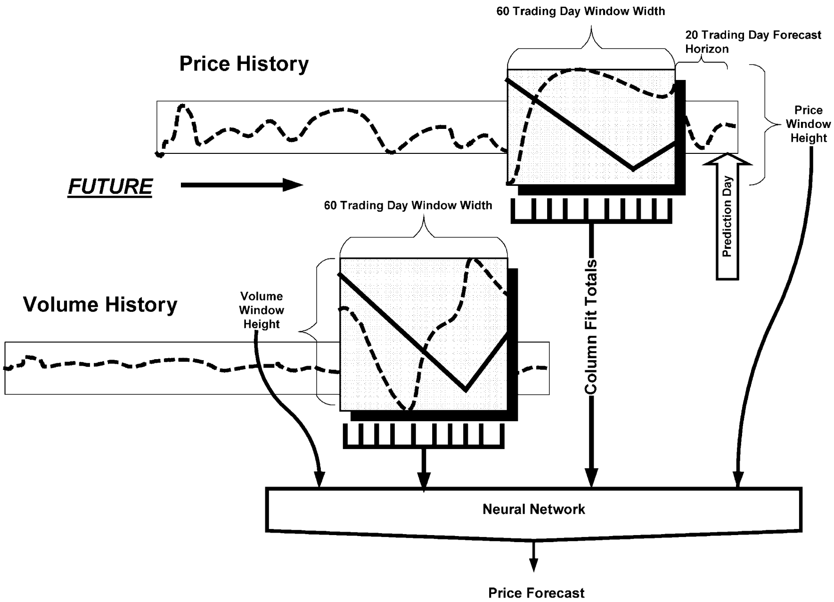

Fig. 4. Relationship between rolling window template fitting mechanism and the neural network system. The window height adapts to the range of values occurring in the window.

Our initial approach is to use *all* of the trading days in the pre-1994 sample for training and *all* of the trading days in the 1994-and-after holdout sample for testing. After training the neural network, the average price increase for simulated purchases on those trading days in the testing sample, which had neural network estimated values for the 20-day price change greater than zero (simulating a trading rule of "buy and hold for 20 days if the total fit is at the 90-percentile level or better"), were not significantly different from the overall price change average for the testing sample—as expected. The mean price change was within 0.03% and the standard deviation was within 0.004 of the respective values for the overall averages reported for the 1994-and-after holdout; testing sample reported on Table 1.

However, a second approach achieves marked success. Training with *only* the trading days occurring at the end of identified (90-percentile and better) bull flags from the pre-1994 training sample, and testing on *only* the 90-percentile and better trading days in the 1994-and-after holdout sample gives better results; shown in Table 2. The simulated trading rule of "buy and hold for 20 days if the fit value for the trading day is at the 90-percentile level or better" results in 46 purchases in the holdout sample period (passing up buying on 6 of the 90-percentile and better identified days in the holdout test sample). The improvement in the price change for 20 days for those trading days with a neural network estimated price change greater than zero over the mean value for all of the identified (90-percentile and better) bull flag trading days in the holdout test sample of 0.66% is significant at the .0866 level. (Note that the first 90-percentile and better trading days in the holdout test sample occurred in the end of April, 1984, and the last 90-percentile and better trading days in the training sample occurred at the end of May, 1993, an 11-month gap, so there is no pollution of the training sample with any element of the test sample due to the 20-day horizon which is used for training).

The significance level of the *t*-test comparison with the mean for all trading days in the 1994-later sample is 0.0000, and this is the probability that the improvement in price prediction (compared to random chance) achieved by the combined template matching and neural network results from random chance. We successfully predict future price from past price and volume history. This fails to confirm the null hypothesis implied by the weak form of the EMH, that is, that price cannot be predicted from price and volume history.

Table 2
Comparison of mean price increase for identified bull flags (90-percentile and better fitting trading days) which have an Experiment-B-neural-network-estimated price change for 20 days greater than zero to all trading days in holdout test sample

	1994-Later
For identified bull flag trading days (90-percentile or better total fit):	
Mean price increase after 20 trading days	2.36%
Standard deviation	0.0268
Number of trading days meeting criteria	52
For identified bull flag trading days which have neural network price increase estimate > 0.0%:	
Mean price increase after 20 trading days	3.02%
Standard deviation	0.0207
Number of trading days meeting criteria	46
Comparison:	
Difference in means	0.66%
Significance[a]	0.0866

[a] For one-tailed, two-sample unequal variance t-test on means in column.

5. Experiment C: improving R^2 with a genetic algorithm

Genetic algorithms are heuristic search procedures (explained in many places, for example Ref. [18]), which may be used in an optimum-seeking manner to configure neural networks [1,35,47]. We use a genetic algorithm to determine the subset of our 22 input variables to use to improve the R^2 correlation between the neural network estimated price increase and the actual, experienced price increase. We use the Pattern Recognition Workbench genetics algorithm software (Unica Technologies) for reducing the number of input variables to a neural network. The genetic algorithm and the default parameter settings, which we use, are documented in Ref. [27].

As the genetic algorithm is a generate-and-test procedure, we divide the 90-percentile and better pre-1994 training sample into subsets to use for training and testing of the genetic algorithm. We use the first 200 trading days for training and the next 64 for testing (from the pre-1994 set of 90-percentile fit and better fit trading days.) The best R^2 value found on the 64 trading day test set was 0.324. Fig. 5 shows that the reduction of the number of input variables did result in improved correlation in several cases during the genetic algorithm's search and testing with the 64-day test set.

Table 3 lists the output coefficients from the best neural network configuration found by the genetic algorithm's search. These output coefficients are the weights used for the respective input variable values in the neural network prediction. The weights in Table 3 may be interpreted as implying that the consolidation part (first seven columns) of the pattern for price is more important for forecasting 20-day horizon price than the consolidation part of the pattern for volume, and that the breakout part (last three columns) of the pattern for volume is more important than the breakout part of the pattern for price.

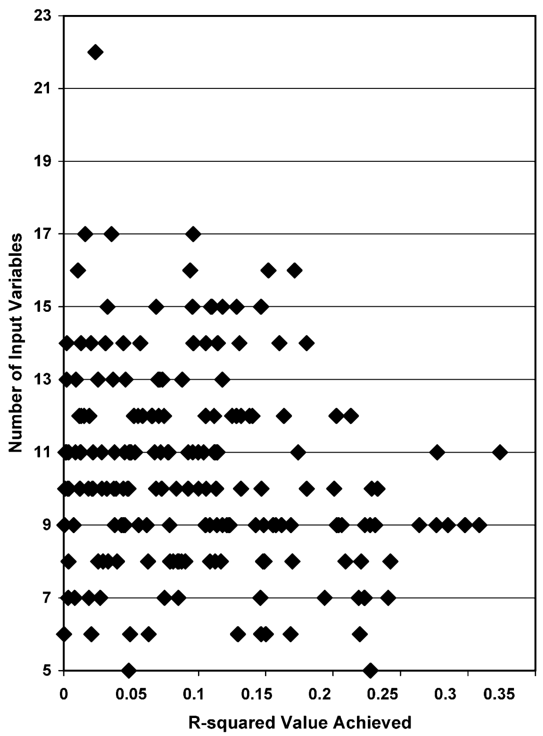

Fig. 5. R^2 value achieved versus number of input variables for each configuration of the neural network generated and tested by the genetic algorithm.

Table 3
Output coefficients for the best Experiment C neural network model

Price window height	0
Volume window height	0
Price fit column 1	0.1339
Price fit column 2	0.2636
Price fit column 3	0.0947
Price fit column 4	0.0407
Price fit column 5	0.0645
Price fit column 6	0.0266
Price fit column 7	0
Price fit column 8	0.0480
Price fit column 9	0
Price fit column 10	0
Volume fit column 1	0
Volume fit column 2	0.0693
Volume fit column 3	0
Volume fit column 4	0
Volume fit column 5	0
Volume fit column 6	0
Volume fit column 7	0
Volume fit column 8	0.1454
Volume fit column 9	0.0401
Volume fit column 10	0.0734

The fitting columns in the variable names in Table 3 are numbered from oldest, 1, to most recent, 10.

The best neural network model found by the genetic algorithm yielded a 0.558 R^2 value for the 52 identified bull flag trading days (90-percentile fit and better) in the holdout test sample. This is better than the 0.356 R^2 value resulting from the Experiment B neural network that uses all 22 input variables. However, the improvement in forecasting ability is achieved at a cost. This neural network configuration with a reduced number of input variables, with the improved R^2, only delivered a 2.50% mean price improvement, which is less of an increase over the average for all trading days than that achieved by the original neural network configuration of Experiment B using all 22 input variables (3.02%) and is not much better than the average price increase experienced for identified (90-percentile fit and better) bull flag trading days overall (2.36%). The trade-off between risk and performance is always with us, it seems.

Fig. 6 shows for the final, reduced neural network configuration, the values for the estimated profit increase on the identified bull flag trading days in the holdout test sample compared with the actual, experienced profit increase on those trading days. The quality of the correlation is striking when we

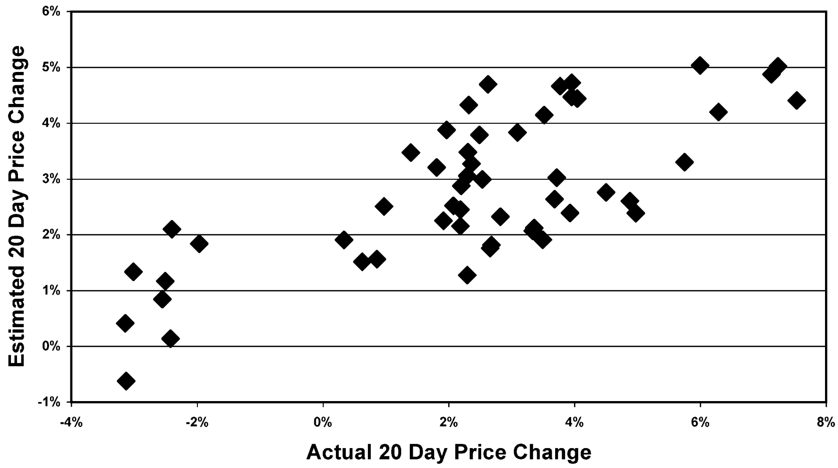

Fig. 6. Values from the 1994-and-later holdout sample for the actual and Experiment-C-neural-network-estimated profit increases for each identified bull flag trading day.

realize that this Experiment C neural network model is trained-and-tested on pre-1994 data (as old as the beginning of 1981) and is used to identify buying opportunities in an interval from the beginning of 1994 to the end of 1996.

6. Experiment D: cross-validation

The universal approximation ability of neural networks often makes it possible to find a model that fits well if enough variations in modeling parameters are tried. Thus, critics may accuse neural network modeling efforts of "data snooping", or modeling from hindsight. The use of significantly long time series, convincing out-of-sample tests, and cross-validation designs is a defense against criticisms of data snooping bias.

We perform this experiment somewhat later than the previous three experiments and have more test data for the New York Stock Exchange Composite Index, this time for the period from 08/06/80 to 06/10/99. The input nodes to the neural network are for the 10 columns of the bull flag pattern template fitting for price and for the window height for price, normalized by the maximum price in the 60-trading-day window (that is, the difference between the maximum and minimum prices occurring in the 60 trading days, divided by the maximum price.) Note that in experiment D we use *price only*, not price and volume as were used in Experiments A, B, and C. There is no pre-processing to select identified bull flag trading days using the pattern recognizer and the 90-percentile rule, as was done before the application of the neural network for Experiment B. All of the training and test data is input to the neural network in Experiment D.

We employ a sliding-window time series cross-validation methodology for evaluation (similar to that used in Ref. [26]). The data is in time series order. We train on 1000 trading days of data, skip 20 days, and test on the subsequent 80 days of data; then slide the training window 80 days forward, train on 1000 days again, skip the next 20, and test on 80 days again; and so forth. The 20 trading days are skipped before the test set begins so that testing is done with a neural network that is trained with at least 20-day-old data, which is the case when forecasting 20 days in advance. In this way we test on 47 folds of 80 trading days each. The setting of each time period's parameters based on the previous period's performance, as is accomplished in this sliding-window validation methodology, is known as *reoptimization* and results in a system that adapts to market drift and testing that has out-of-sample results [33].

For Experiment D we configure the neural net for classification, to differentiate the trading days into three classes: (1) those which are expected to experi-

Table 4
Results by fold for Experiment D1

Fold	Fold values					Cumulative values				
	Average profit			Days	*T*-test	Average profit			Days	*T*-test
	Overall (%)	Purchase (%)	Difference (%)			Overall (%)	Purchase (%)	Difference (%)		
1	0.03	0.71	0.68	12	0.1441	0.03	0.71	0.68	12	0.144114
2	2.35	1.64	− 0.71	70	0.0936	1.19	1.50	0.31	82	0.221751
3	0.97	1.13	0.16	72	0.3586	1.11	1.33	0.21	154	0.240151
4	2.36	4.17	1.80	38	0.0002	1.43	1.89	0.46	192	0.046996
5	3.60	4.25	0.65	54	0.1143	1.86	2.41	0.55	246	0.016830
6	0.22	0.29	0.07	74	0.4437	1.59	1.92	0.33	320	0.079914
7	0.76	0.76	0.00	80	0.5000	1.47	1.69	0.22	400	0.172560
8	4.36	4.36	0.00	80	0.5000	1.83	2.13	0.30	480	0.084785
9	1.07	1.76	0.70	63	0.1215	1.75	2.09	0.34	543	0.047696
10	− 5.19	− 6.21	− 1.02	36	0.3158	1.05	1.57	0.52	579	0.028433
11	2.41	4.71	2.30	17	0.0005	1.18	1.66	0.49	596	0.032347
12	0.63	1.36	0.73	64	0.1177	1.13	1.63	0.50	660	0.020493
13	0.61	1.33	0.73	65	0.0478	1.09	1.61	0.52	725	0.011859
14	1.28	1.17	− 0.11	72	0.4211	1.10	1.57	0.46	797	0.015241
15	2.30	2.30	0.00	80	0.5000	1.18	1.63	0.45	877	0.011896
16	1.87	1.87	0.00	80	0.5000	1.23	1.65	0.43	957	0.011096
17	− 1.34	− 0.90	0.45	71	0.2107	1.08	1.48	0.40	1028	0.012820
18	1.57	2.26	0.69	56	0.1144	1.10	1.52	0.42	1084	0.008318
19	− 2.80	− 3.77	− 0.97	18	0.0664	0.90	1.43	0.53	1102	0.000955
20	0.51	1.14	0.64	48	0.1979	0.88	1.42	0.54	1150	0.000613
21	4.16	4.16	0.00	80	0.5000	1.03	1.60	0.56	1230	0.000348
22	0.84	1.35	0.51	60	0.0464	1.02	1.59	0.56	1290	0.000212
23	0.61	0.61	0.00	80	0.5000	1.01	1.53	0.52	1370	0.000343
24	0.77	1.20	0.43	65	0.2414	1.00	1.51	0.52	1435	0.000271
25	0.73	1.62	0.89	32	0.0181	0.99	1.52	0.53	1467	0.000134
26	0.66	0.67	0.02	66	0.4825	0.97	1.48	0.51	1533	0.000155
27	1.07	1.16	0.09	77	0.3716	0.98	1.46	0.49	1610	0.000151
28	0.14	0.14	0.00	80	0.5000	0.95	1.40	0.45	1690	0.000229
29	0.77	1.35	0.58	49	0.0086	0.94	1.40	0.46	1739	0.000133
30	0.30	0.75	0.45	9	0.2254	0.92	1.40	0.48	1748	0.000058
31	− 1.05	− 1.42	− 0.37	20	0.3083	0.86	1.37	0.51	1768	0.000016
32	0.83	1.68	0.85	32	0.0020	0.86	1.37	0.52	1800	0.000008
33	0.68	1.32	0.64	35	0.0489	0.85	1.37	0.52	1835	0.000005
34	2.49	2.63	0.15	64	0.1723	0.90	1.41	0.51	1899	0.000003
35	1.92	1.92	0.00	80	0.5000	0.93	1.43	0.51	1979	0.000002
36	2.27	2.27	0.00	80	0.5000	0.96	1.47	0.50	2059	0.000002
37	1.37	1.16	− 0.21	61	0.2795	0.98	1.46	0.48	2120	0.000002
38	0.49	0.85	0.37	69	0.2667	0.96	1.44	0.47	2189	0.000002
39	2.53	2.53	0.00	80	0.5000	1.00	1.48	0.47	2269	0.000001
40	1.75	1.75	0.00	80	0.5000	1.02	1.49	0.46	2349	0.000002
41	2.87	2.87	0.00	80	0.5000	1.07	1.53	0.46	2429	0.000001
42	1.03	1.14	0.10	79	0.4257	1.07	1.52	0.45	2508	0.000002
43	3.60	3.17	− 0.44	62	0.1734	1.12	1.56	0.43	2570	0.000003
44	− 1.99	− 0.40	1.59	59	0.0221	1.05	1.51	0.46	2629	0.000001
45	2.19	1.83	− 0.37	72	0.3611	1.08	1.52	0.44	2701	0.000002
46	2.19	2.19	0.00	80	0.5000	1.10	1.54	0.44	2781	0.000002
47	− 0.03	0.27	0.30	69	0.3160	1.08	1.51	0.43	2850	0.000002

"Purchase" refers to the profit resulting from execution of recommendations from the neural net and holding for 20 trading days. "Days" refers to the number of trading days on which purchasing was recommended. The "*T*-test" column values are probabilities of random choice leading to the results found, in a one-tailed *t*-test between the "Purchase" average profit and the "Overall" average profit.

Table 5
Results by fold for Experiment D2

Fold	Fold values					Cumulative values				
	Average profit			Days	*T*-test	Average profit			Days	*T*-test
	Overall (%)	Purchase (%)	Difference (%)			Overall (%)	Purchase (%)	Difference (%)		
1	0.03	–	–	0	–	0.03	–	–	0	–
2	2.35	1.22	− 1.13	11	0.0478	1.19	1.22	0.03	11	0.4819179891
3	0.97	1.76	0.79	1	–	1.11	1.26	0.15	12	0.3899939971
4	2.36	1.74	− 0.62	3	0.2932	1.43	1.36	− 0.07	15	0.4391756372
5	3.60	6.85	3.25	10	0.0000	1.86	3.55	1.69	25	0.0078948191
6	0.22	− 0.77	− 0.99	31	0.0719	1.59	1.16	− 0.43	56	0.2131117889
7	0.76	–	–	0	–	1.47	1.16	− 0.31	56	0.2818479124
8	4.36	4.02	− 0.34	53	0.2396	1.83	2.55	0.72	109	0.0227083408
9	1.07	–	–	0	–	1.75	2.55	0.80	109	0.0121966080
10	− 5.19	–	–	0	–	1.05	2.55	1.50	109	0.0000502505
11	2.41	–	–	0	–	1.18	2.55	1.37	109	0.0001462467
12	0.63	3.20	2.57	16	0.0043	1.13	2.63	1.50	125	0.0000104428
13	0.61	2.18	1.58	31	0.0020	1.09	2.54	1.45	156	0.0000010418
14	1.28	2.21	0.93	41	0.0689	1.10	2.47	1.37	197	0.0000003376
15	2.30	4.04	1.74	41	0.0000	1.18	2.74	1.56	238	0.0000000001
16	1.87	2.10	0.23	27	0.2786	1.23	2.68	1.45	265	0.0000000001
17	− 1.34	− 0.58	0.76	12	0.1534	1.08	2.54	1.46	277	0.0000000000
18	1.57	4.23	2.66	8	0.0079	1.10	2.58	1.48	285	0.0000000000
19	− 2.80	–	–	0	–	0.90	2.58	1.69	285	0.0000000000
20	0.51	–	–	0	–	0.88	2.58	1.71	285	0.0000000000
21	4.16	–	–	0	–	1.03	2.58	1.55	285	0.0000000000
22	0.84	1.21	0.37	4	0.3034	1.02	2.56	1.54	289	0.0000000000
23	0.61	0.24	− 0.37	3	0.3582	1.01	2.54	1.53	292	0.0000000000
24	0.77	1.22	0.45	6	0.1664	1.00	2.51	1.52	298	0.0000000000
25	0.73	2.52	1.79	20	0.0000	0.99	2.51	1.53	318	0.0000000000
26	0.66	1.58	0.92	37	0.0155	0.97	2.42	1.44	355	0.0000000000
27	1.07	–	–	0	–	0.98	2.42	1.44	355	0.0000000000
28	0.14	− 0.42	− 0.55	24	0.0566	0.95	2.24	1.29	379	0.0000000000
29	0.77	–	–	0	–	0.94	2.24	1.30	379	0.0000000000
30	0.30	–	–	0	–	0.92	2.24	1.32	379	0.0000000000
31	− 1.05	–	–	0	–	0.86	2.24	1.38	379	0.0000000000
32	0.83	–	–	0	–	0.86	2.24	1.38	379	0.0000000000
33	0.68	1.84	1.16	14	0.0059	0.85	2.22	1.37	393	0.0000000000
34	2.49	3.09	0.60	36	0.0006	0.90	2.30	1.40	429	0.0000000000
35	1.92	2.11	0.19	70	0.2394	0.93	2.27	1.34	499	0.0000000000
36	2.27	3.18	0.91	56	0.0147	0.96	2.36	1.40	555	0.0000000000
37	1.37	0.29	− 1.08	4	0.0889	0.98	2.35	1.37	559	0.0000000000
38	0.49	4.79	4.30	8	0.0000	0.96	2.38	1.42	567	0.0000000000
39	2.53	3.24	0.70	48	0.0542	1.00	2.45	1.44	615	0.0000000000
40	1.75	8.14	6.39	14	0.0000	1.02	2.57	1.55	629	0.0000000000
41	2.87	4.82	1.95	29	0.0006	1.07	2.67	1.61	658	0.0000000000
42	1.03	3.26	2.23	46	0.0000	1.07	2.71	1.65	704	0.0000000000
43	3.60	4.37	0.77	20	0.1266	1.12	2.76	1.63	724	0.0000000000
44	− 1.99	− 0.13	1.86	29	0.0175	1.05	2.65	1.59	753	0.0000000000
45	2.19	–	–	0	–	1.08	2.65	1.57	753	0.0000000000
46	2.19	3.03	0.84	58	0.0402	1.10	2.67	1.57	811	0.0000000000
47	− 0.03	2.49	2.52	28	0.0001	1.08	2.67	1.59	839	0.0000000000

See the caption for Table 4 for explanation of headings.

ence a price increase of less than 0.0% in the following 20 trading days; (2) those which are expected to experience a 20-day price increase greater than or equal to 0.0% but less than or equal to 1.0%; and (3) those which are expected to experience a 20-day price increase greater than 1.0%. The neural network has three output nodes, one for each of the classes. The output value for each node is a prediction confidence value for each class, respectively, C_1, C_2, and C_3. The design and tuning of a neural network system is more art than science and many decisions are involved. We use the default settings supplied by our commercial neural network tool, identified in a previous section, in all cases. We do no optimization or "snooping" with these settings.

For Experiment D1 we use a trading rule of "buy if $C_3 > C_1$," and for Experiment D2 we us a trading rule of "buy if $C_3 > 3 \times C_1$." Table 4 shows the results for Experiment D1, and Table 5 shows the results for Experiment D2. The trading rule for Experiment D2 is more discriminating than the one in Experiment D1, and the number of trading days that receive buy recommendations is considerably smaller, but the profit percentage realized on that smaller number of days is higher, and the significance is higher. The rightmost five columns in each table contain cumulative results, with overall cumulative results for all folds and all trading days in the bottom line of the table. Results are significant and fail to confirm the null hypothesis, which is that the 20-day horizon profit realized using the trading rule will be no better than the overall average, which is equivalent to the weak form of the EMH.

The sliding-window cross-validation mechanism of Experiment D can be deployed directly in a decision support context to make a purchasing decision for a current day. The training set would end 21 days before the current day. The testing set, now a forecast set, would be reduced to 1 trading day, the current day. Each day the training and forecast sets are moved 1 trading day forward, instead of 80 as in Experiment D, to make use of the latest information.

7. Discussion

The representation of the time dimension in neural network models may be approached with recurrent node architectures to give the network memory. Recurrent node architectures are complex [13]. The approach illustrated in this work brings to bear the universal approximation power of the neural net to a time series forecasting problem in a different way. The technical analysis pattern heuristic paradigm embodies a spatial representation of time, the 10×10 template, which we use to transform the price forecasting problem into an isolated pattern recognition problem out-of-time for the neural network. Then we apply the resulting neural network pattern recognizer to each trading day individually, to re-introduce the dimension of time.

A financial trader is interested in a trading recommendation, rather than in a forecast. Researchers and developers of financial trading applications are interested in the results of applying the resulting trading rule—and they are not interested in the quality of the forecast by conventional forecasting error measures. This difference in orientation and its implications is discussed in Ref. [32]. The output of a price forecasting system may be converted to a trading rule by translating forecasts greater than some minimum percentage into a stock purchase recommendation. We use a forecasted increase of more than 0.0% to signal purchase in Experiment B and two different trading rules based on classification confidence to signal purchase in Experiment D.

A trading rule system is a classification system, matching moments in time with a trading recommendation. The use of multiple classifiers to make a single recommendation, such as is the case in Experiment B, results in a *multiple classifier system*. Multiple classifier systems can be organized as a "conditional," "serial", "hybrid", or "parallel" combinations of pattern recognition, neural network, and/or other classifier methods [31]. In the conditional architecture, a primary classifier, which has low cost or generally applicability, is used first, and if it fails to come to a decision with the desired level of confidence, then a secondary classifier is used. In the serial topology, classifiers are applied in succession, with the output of one classifier used as the input to the next. Ref. [29] is an example of the use of the serial topology. The hybrid organization uses the output of one classifier to indicate which classifier to use subsequently. Examples of this are Refs. [21,41]. A parallel multiple classifier system applies all of the classifiers simultaneously to the problem, and the individual results are combined to produce a final decision. A parallel system whose individual decisions are combined by some system of weighted or unweighted voting is an ensemble classifier [8]. Recently developed ensemble combination methods include bagging and boosting [2], and ensemble classifiers generally perform better than their constituent classifiers alone [2]. These multiple classifier methods may be generalized to include multiple forecasting systems and multiple combinations of classifier and forecasting systems.

The pattern recognizer in Experiment A is a single classifier system. Our work in Experiment B is an example of use of the serial architecture for a multiple classifier system, as only the trading days which are identified as purchase opportunities by the pattern recognizer are given to the neural network for learning, testing, and decision-making. Experiment C involves model optimization rather than classification. Experiment D integrates the pattern recognition and the neural network into a single classifier—perhaps constituting a hybrid multiple classifier organization, if you analyze the resulting neural network, but we are not sure this distinction is useful in this case.

The experimental mechanism in Experiment D is directly adaptable as a deployable decision support system. That decision support system could be extended to a multiple classifier form. Knowledge developed from the use of multiple window widths, multiple price horizons, and/or multiple pattern heuristics could be integrated into the decision support system by a parallel, ensemble method. The possibility exists in this stock trading domain to purchase the net summed value of the recommendations of all of the recommendations rules which might apply at a particular point in time, possibly based on a weighting by some learned confidence in the different windows, horizons, and/or pattern heuristics. Contradictory actions would cancel each other out in the net recommended purchase (or sale). Pattern heuristic selection might be supplied by an expert system, such as is used in Ref. [51] to adaptively select models to use in a forecasting system. A case-based approach (such as is used in Ref. [29]) might be used to coordinate the application of the different technical analysis pattern heuristics. An expert system for cash management could be included. All of this is an area of research and development that we consider to be in the nature of future work.

8. Contributions

Many years of accumulated technical analysis knowledge are available in thousands of books and articles in the practitioner literature. This knowledge was developed by "data mining" and through the application of the "knowledge discovery in databases" (KDD) process [15], even though we academics had not coined those terms or "invented" these methods when this technical analysis development work was done. Most of this technical analysis knowledge is in subjective and qualitative forms, such as anecdotes and cases, and is not in the sort of precise algorithmic form that is required by hard computing automation. This knowledge is more amenable to application through the tools of soft computing, such as fuzzy rules, neural networks, and genetic algorithms. It remains to engineer this knowledge so that it may be tested, evaluated and applied in a consistent manner. The work in this paper is an early step in the direction of engineering technical analysis lore so that they it may be tested and deployed.

The method we devise for using neural networks to improve the effectiveness of template matching is novel. The isolation of the individual column cross-correlation values as variables, which are input to the neural network, and the modularity of the hybrid pattern recognition and neural network method as a whole allow the application of the genetic algorithm to identify the most useful column variables, as is illustrated in Experiment C.

We test the bull flag price and volume pattern heuristic in a rigorous way, which has not been done before. Experiments A, B, and D have robust experimental design frameworks, and all have significant out-of-sample results that fail to confirm the null hypothesis that the markets are EMH weak form efficient. The realization that the weak form of the EMH fails to hold in the case of momentum is beginning to dawn in the academic finance community. New theory is being invented to account for this, based on information diffusion rates [23] or behavioral phenomena, such as "herding" [5] or "overconfidence" [6]. Perhaps our purely empirical paper, which comprises "measurement without theory" [30], will hasten the acceptance of technical analysis and accelerate its development from its present state as a "weak-theory domain" (defined in Ref. [39]). The passing of the EMH paradigm is an interesting story in itself. The reversed order [17] and extended delay between theory and practice in the case of technical analysis is certainly not typical of other modern technologies.

We survey the currently active research area of combining classifiers. The development and understanding of this technology is critical if we are to realize the decision support potential of the new soft computing tools. Experiments B, C, and D demonstrate application of multiple tools, and Experiments B and D illustrate the power in multiple classifier systems.

We are continuing this work to the investigation of:

1) Other window widths, forecasting horizons, and technical analysis pattern heuristics.
2) Architectures for combining the use of multiple window widths, forecasting horizons, and technical analysis pattern heuristics into a single multiple classifier system. There is opportunity here for the integration of fuzzy and expert systems technology.
3) Regression to refine the effectiveness of the template matching, instead of neural networks.
4) Wavelets analysis as a pattern identifier, instead of template matching.
5) A genetic algorithm based method to mine for new technical patterns.

9. Conclusion

We foresee that research and practice in DSS is entering a romantic period characterized by the use of combinations of recently developed and exotic-seeming decision analysis and data modeling tools, highly nonlinear and connectionist models, machine learning techniques, the employment of high-performance computing and gigantic data warehouses on and from the desktop, and by iconoclastic attitudes toward existing theories and accepted beliefs. The experiments reported herein combine pattern recognition, neural network, and genetic algorithm techniques in a novel way to forecast price changes for the NYSE Composite Index, illustrate the romantic style of DSS, and exemplify the superior decision-making results that may be achieved through this approach. In addition, the high quality of the results attained in the experiments reported in this paper spotlights the potential value of stock market technical analysis, particularly the pattern heuristics of technical analysis.

References

[1] B. Back, T. Laitinen, K. Sere, Neural networks and genetic algorithms for bankruptcy predictions, Expert Systems With Applications 11 (4) (1996) 407–413.
[2] E. Bauer, R. Kohave, An empirical comparison of voting classification algorithms: bagging, boosting, and variants, Machine Learning 36 (1999) 105–139.
[3] H. Bessembinder, K. Chan, Market efficiency and the returns to technical analysis, Financial Management 27 (2) (1998) 5–17.
[4] H.K. Bhargava, S. Sridhar, C. Herrick, Beyond Spreadsheets: Tools for Building Decision Support Systems, Computer, IEEE, New York, March 1999, pp. 31–39.
[5] E.C. Chang, J.W. Cheng, A. Khorana, An examination of herd behavior in equity markets: an international perspective, Journal of Banking and Finance 24 (10) (2000) 1651–1679.
[6] K. Daniel, S. Titman, Market efficiency in an irrational world, Financial Analysts Journal (November/December 1999) 28–40.
[7] F.X. Diebold, Forecast combination and encompassing: reconciling two divergent literatures, International Journal of Forecasting 5 (1989) 589–592.
[8] T. Dietterich, Ensemble Methods in Machine Learning, Proceedings/MCS 2000, Cagliari, Italy, June 21–23, 2000. Springer-Verlag, Berlin, 1998, pp. 1–15.
[9] J. Downes, J. Goodman, Dictionary of Finance and Investment Terms, 5th edn., Barron's Educational Series, New York, 1998.
[10] R. Duda, P. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[11] S. Dutta, S. Shekhar, W. Wong, Decision support in non-conservative domains: generalization with neural networks, Decision Support Systems 11 (1994) 527–544.
[12] R. Edwards, J. Magee, Technical Analysis of Stock Trends, 7th edn., Amacom, New York, 1997.
[13] J.L. Elman, Finding structure in time, Cognitive Science 14 (1990) 179–211.
[14] E. Fama, Efficient capital markets: II, Journal of Finance 46 (5) (1991) 1575–1617.
[15] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, The KDD process for extracting useful knowledge from volumes of data, Communications of the ACM 39 (11) 1996, pp. 27–34.
[16] R. Gencay, The predictability of security returns with simple technical trading rules, Journal of Empirical Finance 5 (1998) 347–359.
[17] R.L. Glass, The temporal relationship between theory and practice, The Journal of Systems and Software 10 (1989) 65–67.
[18] D. Goldberg, Genetic and evolutionary algorithms come of age, Communications of the ACM 37 (3) (March 1994) 113–119.
[19] B. Graham, D. Dodd, Security Analysis, classic 1934 edn., McGraw-Hill, New York, 1997.

[20] G. Grudnitski, L. Osburn, Forecasting S&P and gold futures prices: an application of neural networks, The Journal of Futures Markets 13 (6) (1993) 631–643.
[21] T. Hill, W. Remus, Neural network models for intelligent support of managerial decision making, Decision Support Systems 11 (1994) 449–459.
[22] H. Hong, J. Stein, A unified theory of underreaction, momentum trading, and overreaction in asset markets, The Journal of Finance LIV (6) (1999) 2143–2184.
[23] H. Hong, T. Lim, J. Stein, Bad news travels slowly: size, analyst coverage, and the profitability of momentum strategies, The Journal of Finance LV (1) (2000) 265–295.
[24] R. Hoptroff, The principles and practice of time series forecasting and business modelling using neural nets, Neural Computing and Applications 1 (1993) 59–66.
[25] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366.
[26] M.Y. Hu, G. Zhang, C. Jiang, B.E. Patuwo, A cross-validation analysis of neural network out-of-sample performance in exchange rate forecasting, Decision Sciences 30 (1999) 197–216.
[27] R. Kennedy, Y. Lee, B. Van Roy, C. Reed, R. Lippman, Solving Data Mining Problems Through Pattern Recognition, Prentice-Hall, Upper Saddle River.
[28] S. Kim, S. Chun, Graded forecasting using an array of bipolar predictions: application of probabilistic neural networks to a stock market index, International Journal of Forecasting 14 (3) (1998) 323–337.
[29] S. Kim, H.J. Noh, Predictability of interest rates using data mining tools: a comparative analysis of Korea and the US, Expert Systems with Applications 13 (2) (1997) 85–95.
[30] T.C. Koopmans, Measurement without theory, The Review of Economic Statistics 29 (3) (1947) 161–172.
[31] L. Lam, Classifier combinations: implementations and theoretical issues, Multiple Classifier Systems: First International Workshop; Proceedings/MCS 2000, Cagliari, Italy, June 21–23, 2000, Springer-Verlag, Berlin, 1998, pp. 77–86.
[32] G. Leitch, J.E. Tanner, Economic forecast evaluation: profits versus the conventional error measures, American Economic Review 81 (1991) 580–590.
[33] L.P. Lukac, B.W. Brorsen, The usefulness of historical data in selecting parameters for technical trading systems, The Journal of Futures Markets 9 (1) (1989) 55–65.
[34] S. Makridakis, Why combining works? International Journal of Forecasting 5 (1989) 601–603.
[35] V. Maniezzo, Genetic evolution of the topology and weight distribution of neural networks, IEEE Transactions on Neural Networks 5 (1) (January 1994) 39–53.
[36] R. Martinelli, B. Hyman, Cup-with-handle and the computerized approach, Technical Analysis of Stocks and Commodities 16 (10) (1998) 63–66.
[37] L. Medsker, E. Turban, R. Trippi, Neural network fundamentals for financial analysts, Journal of Investing (Spring 1993) 59–68.
[38] T. Plummer, Forecasting Financial Markets, Wiley, New York, 1990.
[39] B.W. Porter, R. Bareiss, R.C. Holte, Concept learning and heuristic classification in weak-theory domains, Artificial Intelligence 45 (1990) 229–263.
[40] M. Qi, Financial applications of artificial neural networks, Handbook of Statistics 14 (1996) 529–552.
[41] T.-S. Quah, C.-L. Tan, K.S. Raman, B. Srinivasan, Towards integrating rule-based expert systems and neural networks, Decision Support Systems 17 (1996) 99–118.
[42] P. Raghubir, S.R. Das, A case for theory-driven experimental enquiry, Financial Analysts Journal (November/December 1999) 56–79.
[43] A. Refenes, Neural Networks in the Capital Markets, Wiley, Chichester, 1995.
[44] D. Rumelhart, G. Hinton, R. Williams, Learning representations by back-propagating errors, Nature 335 (9) (1986) 533–536.
[45] E. Saad, D. Prokhorov, D. Wunsch, Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks, IEEE Transactions on Neural Networks 9 (6) (1998) 1456–1470.
[46] S. Schocken, G. Ariav, Neural networks for decision support: problems and opportunities, Decision Support Systems 11 (1994) 393–414.
[47] W. Siedlecki, J. Sklansky, A note on genetic algorithms for large-scale feature selection, Pattern Recognition Letters 10 (1989) 335–347.
[48] J.L. Treynor, R. Ferguson, In defense of technical analysis, Journal of Finance XL (3) (July 1985) 757–775.
[49] R. Trippi, E. Turban, The impact of parallel and neural computing on managerial decision making, Journal of Management Information Systems 6 (3) (1989) 85–98.
[50] R. Trippi, E. Turban (Eds.), Neural Networks in Finance and Investing, Probus Publishing, Chicago, 1993.
[51] A. Venkatachalam, J. Sohl, An intelligent model selection and forecasting system, Journal of Forecasting 18 (1999) 167–180.
[52] W. Wang, P. Jones, D. Partridge, Diversity between neural networks and decision trees for building multiple classifier systems, First International Workshop; Proceedings/ MCS 2000, Cagliari, Italy, June 21–23, 2000, Springer-Verlag, Berlin, 1998, pp. 240–249.
[53] R.L. Winkler, Combining forecasts: a philosophical basis and some current issues, International Journal of Forecasting 5 (4) (1989) 605–609.
[54] B.K. Wong, Y. Selvi, Neural network applications in finance: a review and analysis of literature (1990–1996), Information and Management 34 (1998) 129–139.
[55] G. Zhang, B. Patuwo, M. Hu, Forecasting with artificial neural networks: the state of the art, International Journal of Forecasting 14 (1998) 35–62.

Dr. William Leigh (www.bus.ucf.edu/leigh), Professor of MIS at the University of Central Florida, completed a BS in mathematics at Millsaps College in 1968 and then worked for several years in the computer industry as a software developer and manager, including 5 years with the IBM, earning an MS in computer science from

Rensselaer Polytechnic Institute and an MBA in industrial management from the University of Cincinnati along the way. Returning to academia, Dr. Leigh completed a PhD in information systems at the University of Cincinnati in 1984 and has worked as a professor in computer science and information systems departments since. Dr. Leigh has co-authored 13 textbooks, several educational and research software products, and over 30 articles in journals in computer science and information systems. A current project is the production of an introductory course in management information systems completed by over 900 undergraduates per term, delivered through the internet.

Dr. Russell Purvis (PhD, Florida State, 1994) is an assistant professor at Clemson University in the Department of Management teaching graduate and undergraduate courses in MIS, IT management and E-commerce. Current research interests include managing the development and implementation of IT applications within organizations. Dr. Purvis has had papers accepted for publication in Management Science, Organization Science, IEEE Transactions in Engineering Management and Information and Management, among others.

Dr. James M. Ragusa has completed Masters and Doctorate degrees in Business Administration from the Florida State University and holds a BS in Mechanical Engineering from the University of Illinois. Since 1987, he has served as Associate Professor on the faculties of both the Colleges of Business and Engineering at the University of Central Florida. Since 1988, Dr. Ragusa has directed as Principal Investigator (PI) or Co-PI sponsored grant activities valued in excess of $1,098,000. During this period, Dr. Ragusa has participated in numerous doctoral dissertation, master’s thesis, and graduate independent studies as advisor or committee member. Dr Ragusa has prepared and presented more than 100 conference papers, and has written 19 journal articles and three books. He has served as special editor for the Expert Systems Journal and Communications of the ACM.