ELSEVIER

Decision Support Systems 23 (1998) 161–174

Decision Support Systems

Forecasting S & P 500 stock index futures with a hybrid AI system

Ray Tsaih [a,*], Yenshan Hsu [b], Charles C. Lai [a]

[a] Department of Management Information Systems, National Chengchi University, Taipei, Taiwan
[b] Department of Finance, National Chengchi University, Taipei, Taiwan

Received 1 August 1996; revised 1 June 1997; accepted 1 January 1998

Abstract

This study presents a hybrid AI (artificial intelligence) approach to the implementation of trading strategies in the S & P 500 stock index futures market. The hybrid AI approach integrates the rule-based systems technique and the neural networks technique to accurately predict the direction of daily price changes in S & P 500 stock index futures. By highlighting the advantages and overcoming the limitations of both the neural networks technique and rule-based systems technique, the hybrid approach can facilitate the development of more reliable intelligent systems to model expert thinking and to support the decision-making processes. Our methodology differs from other studies in two respects. First, the rule-based systems approach is applied to provide neural networks with training examples. Second, we employ Reasoning Neural Networks (RN) instead of Back Propagation Networks. Empirical results demonstrate that RN outperforms the other two ANN models (Back Propagation Networks and Perceptron). Based upon this hybrid AI approach, the integrated futures trading system (IFTS) is established and employed to trade the S & P 500 stock index futures contracts. Empirical results also confirm that IFTS outperformed the passive buy-and-hold investment strategy during the 6-year testing period from 1988 to 1993. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Hybrid AI system; Rule-based system; Reasoning Neural Networks; Back Propagation Networks; S & P 500 stock index futures

1. Introduction

1.1. Problem statements

In modern finance, derivatives such as futures and options play increasingly prominent roles not only in risk management activities but also in price speculative activities. Owing to the high-leverage characteristic involved in derivative tradings, investors can gain enormous profits with a small amount of capital if they can accurately predict the market's direction. However, many factors influence financial markets, including political events, general economic conditions, and traders' expectations. Therefore, predicting the financial market's movements is quite difficult.

Increasingly, according to academic investigations, movements in market prices are not random.

* Corresponding author.

PII: S0167-9236(98)00028-1

Rather, they behave in a highly nonlinear, dynamic manner. The standard random walk assumption of futures prices may merely be a veil of randomness that shrouds a noisy nonlinear process [2,6,7]. To remove this veil and to make the forecasting of futures prices more reliable, the application of expert systems and neural networks have received extensive attention [1,3,8,10,11,16,17].

Rule-based systems and neural networks originate from work in the field of AI. Rule-based expert systems, which emerged in the early 1970s, have received wide interest and have represented the bulk of expert systems applications today. The knowledge of rule-based systems is stored or presented primarily in the form of rules and as problem-solving procedures. Knowledge embedded in the system is easy to read because it uses explicit rules (condition-and-action relationships). In addition, providing explanations for the decisions made by rule-based systems is also feasible since the antecedents of rules specify exactly what conditions activate the rule. Therefore, the primary advantage of rule-based systems over their neural networks counterpart is the 'readability' of the process that the system utilizes to make decisions [23].

In summary, rule-based systems are quite appropriate for frequently recurring problems that are naturally well-handled by rules. However, the rule-based systems approach is limited by its requirement for explicit rules. This problem arises because most human knowledge is implicit, particularly experts' knowledge.

In contrast to rule-based systems, neural networks attempt to emulate the biological system of the human brain in learning and identifying patterns. Moreover, neural networks can more aptly recognize poorly defined patterns. Instead of extracting explicit rules from domain experts, the neural networks approach employs a learning algorithm to autonomously: (a) extract the functional relationship between input and output, which is embedded in a set of historical data (called training examples), and (b) encode it in connection weights. Training examples that are readily available allow neural networks to capture a large volume of information in a rather short period of time and to continuously learn throughout its lifespan. Furthermore, neural networks have the ability to not only deal with noisy, incomplete, or previously unseen input patterns, but to also generate a reasonable response. However, reading and understanding the knowledge in neural networks is difficult because knowledge is distributed over the entire network.

By highlighting the advantages and overcoming the limitations of both the neural networks technique and rule-based systems technique, the hybrid approach can facilitate the development of more reliable intelligent systems to model expert thinking and to support the decision-making processes. To our knowledge, two studies (Bergerson and Wunsch [1] and Trippi and Desieno [16]) have integrated both the neural networks technique and rule-based systems technique to implement their trading strategies in the S&P 500 index futures market.

Bergerson and Wunsch [1] constructed a rule-based daily trading system augmented by a neural network market predictor. They used Back Propagation networks (BP) to predict the market and also employed a risk-management rule to curtail trading losses. However, they expended the majority of their efforts in picking out training examples and in selecting parameters for the neural network system's architecture. Hence, their method relies heavily on manual procedures.

In a related work, Trippi and DeSieno [16] designed a neural network-based intraday trading system assisted by a set of composite trading decision rules. Their system consisted of several trained neural networks and a set of rules for combining the neural networks' results to generate a composite recommendation for the current day's position. However, in contrast to Bergerson and Wunsch [1], they merely fed neural networks with massive amounts of historical data.

Studies of Bergerson and Wunsch [1] and Trippi and DeSieno [16] are inhibited in two distinct ways. First, the ways of picking out training examples adopted in both studies are rather inefficient. One involves picking out training examples manually, the other, feeds in massive amounts of historical data. Second, BP is used in both studies. Although the multi-layered networks with the back propagation learning algorithm [15] have excited the connectionists and rehabilitated confidence in neural networks, several associated undesirable predicaments have caused their effectiveness to deteriorate: (1) exactly how many hidden nodes are necessary for a specific problem is generally unknown; and (2) learning might converge to an undesired attractor (e.g., a relatively optimal network solution), despite the existence of the desired network solution.

These undesirable predicaments might significantly hinder BP's performance. Appendix A provides a brief explanation of such predicaments. Also, more detailed information can be found in Ref. [9]. However, Refs. [1,16] do not address how to resolve the above predicaments.

1.2. Proposed approaches

By overcoming the above drawbacks, this study presents a superior approach to establishing a hybrid AI system that can implement the trading strategies in the S&P 500 stock index futures market. Our approach is as follows: first, the rule-based systems provide training examples to neural networks, in contrast to feeding in massive amounts of historical data or manually picking out training examples. Second, Reasoning Neural Networks (RN) [18–22] is adopted herein to avoid the undesired predicaments encountered in BP. RN has been applied to several categorization problems [5,13,20]. Empirical results in these studies demonstrate the following:

1. RN's learning is always completed;
2. RN's number of required hidden nodes is less than BP's;
3. RN's learning speed is much faster than the back propagation learning algorithm; and
4. RN's internal representation is more acceptable, compared with BP's.

Restated, RN can resolve the undesired predicaments associated with BP.

This study presents dual forecast models (that is, a futures forecast model (FFM) and an extended futures forecast model (EFFM)) to accurately predict the direction of daily price changes in S&P 500 index futures. In addition, FFM is set up with the conventional rule-based systems technique, while EFFM is set up with the neural networks technique. Moreover, the integrated futures trading system (IFTS), which integrates FFM and EFFM, is designed to provide investors with daily trading suggestions on the S&P 500 index futures contracts.

2. The proposed hybrid AI system

2.1. Futures forecast model

FFM adopts the conventional rule-based systems. We derive the knowledge and rules from scholars and experts who are specializing in trading S&P 500 stock index futures. The following description explains the derived rules which are embedded in our FFM.

S&P 500 stock index futures contracts expire four times annually. Daily closing price (closes) data from the nearby contracts are constructed over the period from December 1983 to December 1993. Since we attempt to forecast the direction of daily price change in S&P 500 index futures, only technical indicators are used as inputs. A variety of technical indicators are available. Some technical indicators are effective under trending markets and others perform better under nontrending or cyclical markets. The linear regression analysis and the relative strength index (RSI) are employed to derive 10 variables of inputs from the daily closes under trending and nontrending markets, respectively. [1] Table 1 lists the definitions of 10 derived input variables. SP, SN, LU and LD are derived from the following linear regression model with 14 days of data: [2]

$$C_t = \alpha + \beta * t + \varepsilon_t, \qquad t = 1, 2, \ldots, 14 \tag{1}$$

where C_t denotes the close on day t, α and β are unknown parameters, t represents the time variable, and ε_t is the random disturbance term.

Output of the regression model gives values for α (intercept) and β (slope). The slope is positive when prices are rising and negative when they are falling. Linear regression analysis also provides the coefficient of determination, denoted by r^2, to measure

[1] Please refer to Ref. [4] for further details.

[2] This study tries to forecast the daily price changes in S&P 500 stock index futures. If we employ too much historical data (for example, 30 days of data) in the linear regression model, we might include some outdated data which could bias the daily forecast. On the other hand, if we employ too little data (for example, 7 days of data), we might not be able to generate reliable output from the linear regression model. Therefore, we chose to employ 14 days of data in the linear regression analysis although the choice is arbitrary.

Table 1
The definitions of 10 derived variables

Variable	Description
SP	Equals 1 when the slope of price trend is significantly positive[a]; -1, otherwise.
SN	Equals 1 when the slope of price trend is significantly negative[a]; -1, otherwise.
LU *	Equals 1 when the oscillator[b] crosses upward over its 3-day moving average[c]; -1 otherwise.
LD *	Equals 1 when the oscillator[b] crosses downward over its 3-day moving average[c]; -1 otherwise.
UD	Equals 1 when the closing price of the index is up at the present day; -1, otherwise.
AUD	Equals 1 when the closing prices are either up or down consecutively for at least 3 days; -1 otherwise.
RSI1 *	Equals 1 when the stochastic RSI[d] falls from 100; -1 otherwise.
RSI2 *	Equals 1 when the stochastic RSI[d] rises from 0; -1 otherwise.
RSI3	Equals 1 when the stochastic RSI[d] is greater than 90; -1 otherwise.
RSI4	Equals 1 when the stochastic RSI[d] is less than 10; -1 otherwise.

* These variables serve as triggers.
[a] We test whether the slope is statistically significant (positive or negative) based on a 5% significance level; see Eq. (1).
[b] The forecast oscillator (FO) is computed according to Eq. (2).
[c] The 3-day moving average line of the forecast oscillator is computed according to Eq. (3).
[d] The stochastic RSI is computed according to Eq. (4).

the strength of the linear relationship. We test whether the slope is statistically significant (positive or negative) by comparing the r^2 value to a critical value based on a 5% significance level. We designed the SP and SN variables to show the direction as well as the strength of the price trend. The SP variable is set to 1 when the slope is significantly positive. Meanwhile, the SN variable is set to 1 when the slope is significantly negative.

For LU and LD variables, we develop an oscillator based on the regression forecast. We compute the forecast oscillator (FO) according to the following equation:

$$\mathrm{FO} = 100 * \frac{C - C_{\mathrm{f}}}{C} \tag{2}$$

where C denotes the daily close and C_{f} represents today's forecast close from the previous 14 daily closes based on Eq. (1). Hence the forecast oscillator is constructed on the basis of the difference of the realized close and the forecast close from the regression. Then, we develop a 3-day moving average line of the forecast oscillator:

$$\mathrm{MA}_t(3) = \frac{\mathrm{FO}_t + \mathrm{FO}_{t-1} + \mathrm{FO}_{t-2}}{3} \tag{3}$$

The LU variable is set to 1 if the oscillator crosses upward over its 3-day moving average and the LD variable set to 1 if the oscillator crosses downward under its 3-day moving average. That the oscillator moves above (or below) its 3-day moving average signals the potential trend change in prices. In facing the down trend, selling the futures might be desirable; when facing an up trend, purchasing the futures might be desirable. Thus, LU and LD are designed to signal the trading opportunities, and serve as triggers [3] of our hybrid AI system.

Both UD and AUD variables also attempt to capture the current market trend at 1 day and 3 days, respectively. UD is set to 1 if today's close exceeds yesterday's; otherwise, it is set to -1. AUD is set to 1 if the close moves upward or downward consecutively for three trading days; otherwise, it is set to -1.

RSI1, RSI2, RSI3 and RSI4 belong to the stochastic RSI oscillator (stochRSI). The relative strength index (RSI) is quite effective in extracting price information for a nontrending market. A stochRSI oscillator is computed as follows:

$$\mathrm{stochRSI} = 100 * \frac{\mathrm{RSI} - \mathrm{RSI_L}}{\mathrm{RSI_H} - \mathrm{RSI_L}} \tag{4}$$

where $\mathrm{RSI} = 100 * (S_{\mathrm{u}})/(S_{\mathrm{u}} - S_{\mathrm{d}})$; S_{u} = the sum of up-day momentum over a 14-day period; the up-day momentum is zero if today's close is less than yesterday's; otherwise, it is the absolute difference between the two closes; S_{d} = the sum of down-day momentum over a 14-day period; the down-day mo-

[3] Trigger variables are used to determine whether the trading should be initiated on a particular trading day. If values of these variables are all -1, no action is recommended on that trading day. If some of these triggers are not -1, the hybrid AI system is employed to generate the trading suggestions. In facing a down trend, you might want to sell (short) futures; while facing an up trend, you might want to purchase (long) futures.

mentum is zero if today's close is greater than yesterday's; otherwise, it is the absolute difference between the two closes; RSI_H = the highest RSI among the current RSI and the preceding 13 RSIs; RSI_L = the lowest RSI among the current RSI and the preceding 13 RSIs.

The stochRSI measures the location of RSI within its recent range, indicating short-term momentum extremes. According to Eq. (4), stochRSI will be 100 if today's RSI is the highest one and it will be zero if today's RSI is the lowest one. A stochRSI value of 100 provides an excellent entry point into the down trend. In contrast, a stochRSI value of zero provides an excellent entry point into the up trend. Both RSI1 and RSI2 variables serve as triggers in detecting the trading opportunities.

Assume that you sold the futures when the stochRSI was 100; then the up trend resumes and the stochRSI again moves above 90 toward 100. Reversing the short position to a long position would be desirable. On the other hand, assume that you purchased the futures when the stochRSI was zero; the down trend then resumes and the stochRSI again moves below 10 toward zero. Reversing the long position to a short position would be desirable. With this in mind, both RSI3 and RSI4 are designed and used to capture the current market trend.

In summary, among these 10 variables, four (i.e., LU, LD, RSI1, and RSI2) serve as triggers and the other six (i.e., SP, SN, UD, AUD, RSI3, and RSI4) capture the tendencies of the current futures market.

For each trading day, we not only derive the values of these 10 variables from the closing data, but also use the derived information to describe the features of the current futures market. Notably, the same information (i.e., values of the 10 variables) can be derived from the price data observed at different time epochs. If each distinct set of derived information is called a case, the market's daily conditions can be divided into at most 1024 cases with 10 variables. Restated, we categorize the daily conditions of the futures market into a variety of cases through processing futures historical data. According to the derived case information, each trading day becomes a case observation. During a particular time period, one case might recur several times, thereby allowing for several observations of this case. Those observations provide further insight into the frequency of there being an up-day [4] (and a down-day [5]) for the case occurring during the observed period. For instance, assume that a case has 10 observations observed previously, among which six are up days and four are down days. Thus, the observations suggest that previously the case occurred with a 60% frequency on an up day and a 40% frequency on a down day. Once a case is

Table 2
The definitions of rules used to categorize the cases

Case group	Definition
Trigger_OFF	LU = −1 and LD = −1 and RSI1 = −1 and RSI2 = −1.
Trigger_ON:	LU = 1 or LD = 1 or RSI1 = 1 or RSI2 = 1.
Obvious_LONG	(LU = 1 or LD = 1 or RSI1 = 1 or RSI2 = 1) and (at least 55% of the observations. in the past 4 years are up-days)[a].
Obvious_SHORT	(LU = 1 or LD = 1 or RSI1 = 1 or RSI2 = 1) and (at least 55% of the observations in the past 4 years are down-days)[a].
Obvious_WAIT	(LU = 1 or LD = 1 or RSI1 = 1 or RSI2 = 1) and (observations of past 4 years are exactly 50% up-days and exactly 50% down-days)[a].
Non-obvious	(LU = 1 or LD = 1 or RSI1 = 1 or RSI2 = 1) and otherwise.
Unobserved	case not observed from the observations of past 4 years.

If the next day's closing price exceeds the current day's, we define that as an up-day occurrence; if the next day's closing price is less than the current day's, we define that as a down-day occurrence.

[a] We have evaluated the forecasting performances with various values of Y (the length of the observed period) and P (the threshold percentage used to classify the case). Y had been 3, 4, 5, or 6; P had been 50%, 55%, 60%, or 65%. If Y is too short, the amount of samples is so small that the result of analyzing the observed cases is statistically insufficient. However, if Y is too long, the result of the analysis may be biased due to including too much historical data. If P is too low, the new obvious cases behave the same as their past counterparts. If P is too high, there are too few obvious cases. From the testing results, $Y = 4$ and $P = 55\%$ is the best one. So $Y = 4$ and $P = 55\%$ is adopted for the ongoing research. For the detailed information, please refer to Ref. [12].

[4] If the next day's closing price exceeds the current day's, we define that as an up-day occurrence.

[5] If the next day's closing price is less than the current day's, we define that as a down-day occurrence.

Table 3
The algorithm for constructing the RB system

Step 1	Set U1, U2, U3, U4, D1, D2, D3, D4 equal 0 initially; If LU = 1 then count the number of up-days (U1) and down-days (D1) from the observations of cases during the past 4 years whose LU = 1 and SP, SN, UD, AUD, RSI3 and RSI4 are the same as the case; If LD = 1 then count the number of up-days (U2) and down-days (D2) from the observations of cases during the past 4 years whose LD = 1 and SP, SN, UD, AUD, RSI3 and RSI4 are the same as the case; If RSI1 = 1 then count the number of up-days (U3) and down-days (D3) from the observations of cases during the past 4 years whose RSI1 = 1 and SP, SN, UD, AUD, RSI3 and RSI4 are the same as the case; If RSI2 = 1 then count the number of up-days (U4) and down-days (D4) from the observations of cases during the past 4 years whose RSI2 = 1 and SP, SN, UD, AUD, RSI3 and RSI4 are the same as the case.
Step 2	Let U# = U1 + U2 + U3 + U4 and D# = D1 + D2 + D3 + D4; Let U% = U#/(U# + D#) and D% = D#/(U# + D#); If U# = D# = 0 then case_result = Unobserved; If U% > = 55% then case_result = Obvious_LONG; If D% > = 55% then case_result = Obvious_SHORT; If U% = D% = 50% then case_result = Obvious_WAIT; If ((55% > U% > 50%) or (55% > D% > 50%)) then case_result = Non_Obvious.

presented, rules shown in Table 2 are used to categorize it. These rules are implemented in the algorithm of Table 3 which is employed to analyze the derived cases information of the previous 4 years to establish the rule-based system (RB). This RB system is used to identify cases in the upcoming year. For instance, a case that is Obvious_LONG (_SHORT) implies that at least 55% of the observations of this case in the previous 4 years are up (down) days. A circumstance in which the market conditions only slightly change in the near future makes the information of being Obvious_LONG (or Obvious__SHORT) quite useful for forecasting the direction of price changes in the future. [6] This idea is implemented in the FFM algorithm, as presented in Table 4.

Table 4
The algorithm of FFM

Step 1	If LU = LD = RSI1 = RSI2 = − 1 then case_result = Trigger_OFF and take a rest.
Step 2	case_result = RB(case)
Step 3	If case_result = Obvious_LONG then long (buy) a futures contract; If case_result = Obvious_SHORT then short (sell) a futures contract; If (case_result = Obvious_WAIT or case_result = Non_Obvious or case_result = Unobserved or case_result = Trigger_OFF) then take a rest.

Step 1 checks whether there is any activated trigger; if there is none, the system makes no suggestion about the trading decision. In Step 2, the RB system identifies the case. In Step 3, the system makes the trading suggestion for the case of either Obvious_LONG or Obvious_SHORT.

2.2. Extended futures forecast model

The information source of FFM is primarily the accumulation of previous case observations (i.e., the RB). Basically, RB provides information regarding previous obvious cases to forecast the obvious cases of the future, and information regarding previous non-obvious cases to forecast the non-obvious cases of the future. Thus, RB provides useful trading suggestions for the obvious cases, but does not adequately handle the non-obvious cases since the previous non-obvious case cannot offer sufficient information to accurately forecast the future non-obvious case. The empirical testing results also verify this intuition. [7] Therefore, FFM utilizes RB to merely forecast the obvious cases of the future. Moreover, EFFM is designed to facilitate FFM in dealing with the non-obvious cases.

[6] For a case with a higher frequency, to have an up-day in the past makes it more possible to have an up-day in the future; in contrast, for a case with higher frequency, to have a down-day in the past makes it more possible to have a down-day in the future.

[7] Detailed information can be found in Ref. [12].

A circumstance in which (a) the obvious cases are the majority of the market and (b) the previous obvious case can provide good trading suggestions for the obvious cases allows us to assume that the mechanism embedded in the obvious case can capture the majority of the market trend. Restated, the information embedded in the set of obvious cases is helpful in dealing with the non-obvious cases. Therefore, how to obtain and utilize the useful implicit information embedded in the set of obvious cases are relevant tasks. To achieve such tasks, the neural networks technique is applied toward EFFM.

The main components of EFFM are the four artificial Neural Networks (ANN) systems. Once a non-obvious case is presented, each of its four trigger variables (i.e., LU, LD, RSI1, and RSI2) is used respectively to trigger an ANN. Restated, each of the four ANNs is executed only when its corresponding trigger is on. Next, the values of the current day's six status variables (i.e., SP, SN, UD, AUD, RSI3, and RSI4) are fed into ANN as inputs. Two output nodes are used here; the output vector (which $\in [-1, 1]^2$) will show the recommendation opinion (see Table 5). The triggered ANNs forecast the direction of price changes. A voting mechanism is used to combine the opinions of these ANNs. Thereafter, EFFM recommends what the trading decision should be for the non-obvious case. Table 6 displays the algorithm corresponding to EFFM.

In the set-up stage of EFFM, the previous obvious cases picked up from the RB system are used as training examples of ANN. Among those cases are included the Obvious_LONG cases, the Obvious_SHORT cases, and the Obvious_WAIT cases. Each case is translated into the format of a training input–output vector. The input vector denotes the values of the six status variables; Table 7 defines the desired output vector. The value of each trigger variable determines whether a training case should be assigned to its corresponding ANN. With the associated training examples, each of the four ANNs develops its own network structure.

Notably, the above design does not specify the type of ANN model; adopting a different ANN model in EFFM would yield a different forecasting performance. To select an appropriate ANN model for EFFM, we evaluate the forecasting performance of the three ANN models: Perceptron (PN) [14], BP [9], and RN [19]. The three ANNs models (i.e., PN, BP and RN) are trained with the same obvious cases

Table 5
The output values of ANN and their corresponding recommendations

Output	Recommendation
(1, −1)	LONG
(−1, 1)	SHORT
(−1, −1)	WAIT
(1, 1)	No comment

Table 6
The algorithm of EFFM

Step 1 Set #L, #S equal 0 initially;
If LU = 1 then
Run the ANN for LU;
If the output of the ANN is LONG then #L = #L + 1;
If the output of the ANN is SHORT then #S = #S + 1;
If LD = 1 then
Run the ANN for LD;
If the output of the ANN is LONG then #L = #L + 1;
If the output of the ANN is SHORT then #S = #S + 1;
If RSI1 = 1 then
Run the ANN for RSI1;
If the output of the ANN is LONG then #L = #L + 1;
If the output of the ANN is SHORT then #S = #S + 1;
If RSI2 = 1 then
Run the ANN for RSI2;
If the output of the RN is LONG then #L = #L + 1;
If the output of the RN is SHORT then #S = #S + 1.
Step 2 If #L > #S then recommendation = LONG
Else if #L < #S then recommendation = SHORT
Else recommendation = WAIT;
Step 3 If recommendation = LONG, then long futures contract;
If recommendation = SHORT, then short futures contract;
If recommendation = WAIT, then take a rest.

Step 1 says that an ANN is run when its corresponding trigger is activated. In Step 2, a voting mechanism is used to combine the four ANNs' opinions. Step 3 makes the trading suggestion for the case.

Table 7
The definition of the desired output values for the ANNs

Case	Desired output values
Obvious_LONG	(1, −1)
Obvious_SHORT	(−1, 1)
Obvious_WAIT	(−1, −1)

in the preceding 4 years; the trained ANNs are then placed into EFFM. Next, EFFMs are applied to the non-obvious cases of the following year to assess their performance throughout the period from 1984 to 1993. The fact that different initial weights may yield different learning results in ANNs accounts for why we make five independent replications (runs) of the simulation.

Table 8 summarizes the forecasting performances of the various EFFM configurations. In Table 8, the number of test equals the difference of the amount of the observed non-obvious cases and the 'No Comment' cases, and success is achieved when the actual output is the same as the desired output. The success rate equals the ratio of the number of successes to the number of tests. Table 8 reveals that most annual success rates exceed 50%.

With all average success rates exceeding 50%, EFFM exhibits the ability to handle the non-obvious cases. Furthermore, empirical results demonstrate that

Table 8
The forecasting performances of the various EFFM configurations

	Year	1988	1989	1990	1991	1992	1993	1988–1993
Panel A: EFFM with RN[a]								
Replication 1	Number of tests	15	17	16	15	14	18	95
	Success rate (%)	66.67	76.47	56.25	33.33	71.43	55.56	60.00
Replication 2	Number of tests	21	20	14	15	14	18	102
	Success rate (%)	66.67	55.00	57.14	33.33	71.43	55.56	61.17
Replication 3	Number of tests	21	20	15	15	14	18	103
	Success rate (%)	66.67	80.00	53.33	33.33	71.43	55.56	61.17
Replication 4	Number of tests	21	17	14	15	14	18	99
	Success rate (%)	66.67	76.47	57.14	33.33	71.43	55.56	60.61
Replication 5	Number of tests	21	17	12	15	11	16	92
	Success rate (%)	66.67	82.35	75.00	33.33	72.73	62.50	65.22
Panel B: EFFM with BP[b]								
Replication 1	Number of tests	15	20	17	15	11	16	94
	Success rate (%)	66.67	50.00	52.94	66.67	72.73	68.75	61.70
Replication 2	Number of tests	21	14	17	15	14	16	97
	Success rate (%)	52.38	78.57	47.06	33.33	71.43	43.75	53.61
Replication 3	Number of tests	21	20	16	15	14	16	102
	Success rate (%)	61.90	65.00	68.75	33.33	71.43	37.50	56.86
Replication 4	Number of tests	14	15	13	15	14	9	80
	Success rate (%)	71.43	73.33	69.23	33.33	71.43	22.22	58.75
Replication 5	Number of tests	12	12	14	15	15	10	78
	Success rate (%)	58.33	75.00	64.29	33.33	66.67	60.00	58.97
Panel C: EFFM with PN								
Replication 1	Number of tests	14	6	17	15	14	15	81
	Success rate (%)	50.00	16.67	52.94	33.33	71.43	66.67	51.85
Replication 2	Number of tests	20	17	13	15	11	16	92
	Success rate (%)	55.00	70.59	38.46	66.67	36.36	56.25	55.44
Replication 3	Number of tests	15	3	16	15	14	14	77
	Success rate (%)	60.00	33.33	56.25	33.33	71.43	64.29	55.84
Replication 4	Number of tests	6	18	7	6	5	7	49
	Success rate (%)	33.33	55.56	71.43	50.00	60.00	85.71	59.18
Replication 5	Number of tests	9	15	4	3	5	2	38
	Success rate (%)	44.44	66.67	50.00	66.67	60.00	0.00	55.26

The last column presents the average success rates and total number of tests in each replication over the 6 years.
[a]Based upon five replications, the maximal, minimal, and average numbers of hidden nodes used are 8, 2, and 3.33, respectively.
[b]The numbers of hidden nodes used is 4.

Table 9
The μ_1 and μ_2 values of the various EFFM configurations

	EFFM with RN	EFFM with BP	EFFM with PN
μ_1 (%)	60.69	57.87	55.19
μ_2	98.2	90.2	67.4

the RN model is better than the BP and PN according to the following considerations. One consideration is based on the mean of the average success rates (μ_1) and the mean of total successful test samples (μ_2). These two means are calculated from the five replications. An ANN model having a higher μ_1 and higher μ_2 is better. As Table 9 reveals, EFFM with RN has a higher μ_1 and higher μ_2 than BP and PN.

The other consideration is based on the mean (μ_3) and standard deviation (Std.) of annual success rates. μ_3 and Std. measure the variability of forecasting performances caused by different initial weights. An ANN model having higher μ_3 and lower Std. is desired. According to Table 10, from the perspective of μ_3 and Std., RN outperforms BP and PN.

A possible explanation for BP and PN's poor performance is their associated drawbacks. PN's poor performance is possibly due to PN's inability to handle nonlinearly separable problems [9]. Although BP can handle nonlinearly separable problems, BP is usually trapped in a local minimal learning result. Since RN outperforms BP and PN, we adopt EFFM with RN.

Table 10
The μ_3 and Std. values of the various EFFM configurations

Year	1988	1989	1990	1991	1992	1993
EFFM with RN						
μ_3 (%)	66.67	73.63	59.15	33.33	71.64	56.82
Std. (%)	0.00	10.12	7.29	0.00	0.48	2.68
EFFM with BP						
μ_3 (%)	61.44	66.67	59.74	40.00	70.59	47.76
Std. (%)	6.63	10.60	9.03	13.34	2.14	15.63
EFFM with PN						
μ_3 (%)	51.56	57.63	52.63	46.30	61.22	62.96
Std. (%)	7.69	16.47	9.66	15.27	14.09	15.22

2.3. Integrated futures trading system

Based on FFM and EFFM, we develop the integrated futures trading system (IFTS) for trading S&P 500 stock index futures contracts. IFTS consists of two units: the periodical off-line training unit and the daily on-line prediction unit.

The periodical off-line training unit attempts to construct two subsystems corresponding to FFM and EFFM. From the time series of price data of the previous 4 years (P_h), the derived case information (D_h) is calculated first. D_h consists of the case observations. We use D_h and the algorithms listed in Tables 3 and 4 to construct the subsystem (S_{FFM}) corresponding to FFM. Then, using S_{FFM}, we select the obvious cases from D_h, which consist of the training examples for RNs (T_h). Then, we use T_h, the learning algorithm of RN, and the algorithm listed in Table 6 to construct the subsystem (S_{EFFM}) corresponding to EFFM. Notably, S_{FFM} and S_{EFFM} are adopted in the daily on-line prediction unit.

The daily on-line prediction unit provides futures investors with trading recommendations. From the time series of the current price data (P_c), the daily on-line prediction unit calculates the derived case information (D_c) for the current day. Then, D_c is inputted into S_{FFM}. Next, S_{FFM} analyzes D_c. If D_c is the non-obvious case, S_{FFM} triggers S_{EFFM}. More specifically, S_{FFM} handles the obvious cases, S_{EFFM} handles the non-obvious cases, and the trading recommendation for the current day (R_c) is generated by either S_{FFM} or S_{EFFM}.

3. The performance evaluation

We use the daily S&P 500 stock index futures price data from 1984 to 1993 for the evaluation. The simulation strategy is as follows: 4-year daily data, from 1984 to 1987, are used to construct an IFTS. This IFTS is then applied to forecast the price movements in 1988. We replicate this strategy throughout the period from 1989 to 1993. In other words, the testing period in the simulation is from 1988 to 1993. This period covers one bear market (1989–1990) and one bull market (1991–1993). Therefore, we can investigate the performance of our IFTS under dif-

Table 11
The forecasting performances of FFM and EFFM in each testing year

Year	1988	1989	1990	1991	1992	1993
FFM						
Success rate (%)	57.69	58.67	56.38	59.38	58.95	58.76
Number of tests	78	75	94	96	95	97
EFFM						
Success rate (%)	66.67	76.47	56.25	33.33	71.43	55.56
Number of tests	15	17	16	15	14	18
Both FFM and EFFM						
Success rate (%)	59.14	61.96	56.36	55.86	60.55	58.26

Table 12
The summary over 1988–1993 and their associated binomial tests

	(a) Total number of tests	(b) Total number of successes	Success rate (%): b/a	Binomial test[a] Z-value
FFM	535	312	58.32	3.90[c]
EFFM	95	57	60.00	1.99[b]
FFM and EFFM	630	369	58.57	4.37[c]

The binomial test is to test whether the success rate is significantly different from 50%.
[a]The binomial test is based on Eq. (5).
[b]Significant at the 5% level.
[c]Significant at the 1% level.

ferent market characteristics. Regardless of what trades are executed, they occur at a fixed time interval during the trading day. More specifically, IFTS enters the market just before the close of trading for that day, and unwinds its position before the close of trading for the next day. [8] IFTS, which does not use a stop-loss mechanism, always carries the futures position for one day (24 h).

Table 11 presents the above evaluation results. [9] Notably, the success rate for each testing year exceeds 50% except for that of EFFM in 1991. Table 12 summarizes the results for the entire period. The binomial test is employed to test whether the success rate varies significantly from 50%. The binomial test proceeds as follows.

Let N and T be the total number of tests and the total number of successes in the simulation. Under the null hypothesis that the success rate is equal to 50%, the test statistic, T, has approximately a normal distribution with a mean, $E(T) = N/2$, and a variance, $\mathrm{VAR}(T) = N/4$ if each test of the simulation is independent from each other. When N is large, T, when standardized, has approximately a unit normal distribution. The binomial test is based on the following Z-statistic:

$$Z = \frac{T - \frac{N}{2}}{\sqrt{\frac{N}{4}}} \tag{5}$$

The binomial tests indicate that all of the success rates dramatically exceed 50%. The collaboration of FFM and EFFM is quite effective in accurately forecasting the direction of daily price change in the S&P 500 index futures market.

The trading performance of IFTS is also evaluated by simulating the purchasing and selling of the S&P 500 stock index futures contracts from 1988 to 1993. Fig. 1 depicts the holding period returns (HPRs) of IFTS and of the buy-and-hold (B&H) strategy. With a round-trip transaction cost of \$60, [10] the holding period return is computed as follows:

holding period return

$$= \prod_{\text{all transactions}} \frac{500C_{t+1} - 60}{500C_t} - 1 \tag{6}$$

where C_t denotes the close on day t. At the end of 1993, IFTS had a holding period return of 94.96%, exceeding that of the buy-and-hold strategy, 56.89%.

[8] Since only the closing prices are available to us, all trades are assumed to be at the closing price in our simulations. Although it might not be possible to trade the futures at the close for every trade, the impact of this assumption on the simulation results is negligible.

[9] The various FFM and EFFM configurations have been evaluated; however, only the test results of FFM (with $Y = 4$ and $P\% = 55\%$) are reported herein. The evaluation results are available from authors upon request.

[10] The amount of the transaction cost according to Trippi and DeSieno [10].

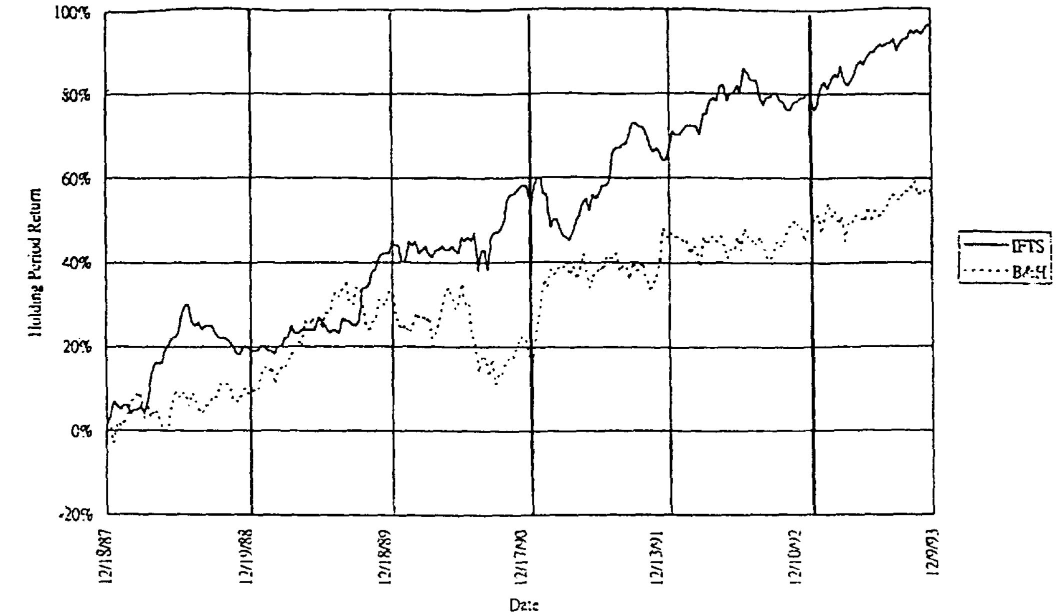


Fig. 1. The holding period returns with respect to IFTS and B&H for S&P 500 index futures trading from 1988 to 1993.

Table 13 displays the annual success rates and holding period returns for IFTS as well as for the buy-and-hold strategy. According to Panel A, by considering only the obvious cases, the trading performance of IFTS (56.09%) is worse than that of the buy-and-hold strategy (56.89%). However, as Panel C indicates, by considering both the obvious cases and the non-obvious cases, the trading performance of IFTS (94.96%) is better than that of the buy-and-hold strategy (56.89%). Based on the trading simula-

Table 13
The yearly success rates and holding period returns with respect to IFTS and buy-and-hold strategies

Year	1988	1989	1990	1991	1992	1993	1988–1993
Panel A: Using IFTS to handle the obvious cases							
Success rate	57.69%	58.67%	56.38%	59.38%	58.95%	58.76%	58.32%
HPR	11.42%	11.96%	7.27%	4.67%	3.89%	7.27%	56.09%
Panel B: Using IFTS to handle the non-obvious cases							
Success rate	66.67%	76.47%	56.25%	33.33%	71.43%	55.56%	60.00%
HPR	7.41%	6.08%	3.70%	0.52%	3.08%	2.02%	24.90%
Panel C: Using IFTS to handle both the obvious cases and non-obvious cases							
Success rate	59.14%	61.96%	56.36%	55.86%	60.55%	58.26%	58.57%
HPR	19.67%	18.77%	11.24%	5.22%	7.09%	9.44%	94.96%
Panel D: the buy-and-hold strategy							
HPR	11.55%	17.49%	−7.96%	12.02%	11.49%	4.78%	56.89%

The last column presents the average success rates and total holding period returns from 1988 to 1993.

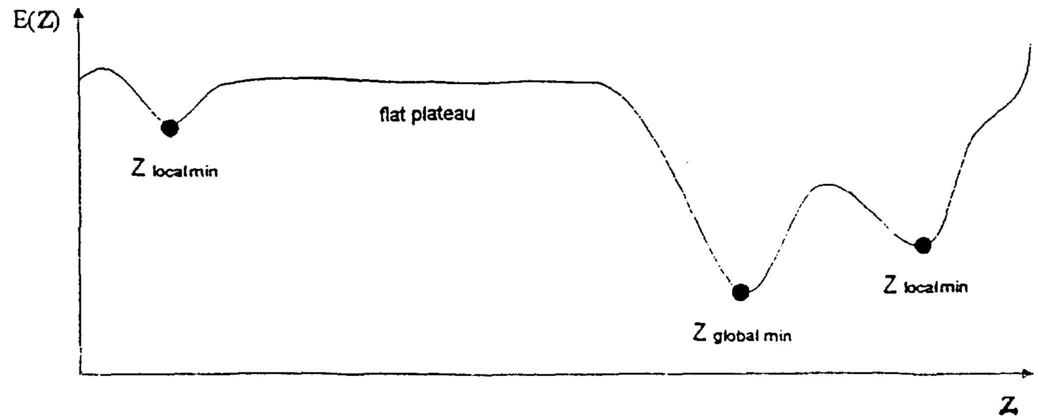


Fig. 2. Function surface on the weight space.

tion results, IFTS outperforms the passive buy-and-hold strategy in the S&P 500 stock index futures market.

4. Conclusions and future work

This paper presents a hybrid AI approach to implement trading strategies in the S&P 500 stock index futures market. Based on the results presented, the following suggestions can be offered.

(1) Predict and trade in the futures market by adopting the hybrid AI approach which integrates both the rule-based systems technique and neural networks technology.

(2) Process the futures price data to derive the case information for prediction. Among the variables of the case information, some are trigger variables and others are status variables. Trigger variables determine the timing of when to provide the trading suggestion; meanwhile, status variables capture the information of the current futures market. For each trading day, we derive case information, which is either an observation (in the past) or an occurrence (at the present) of a certain case.

(3) Propose FFM to analyze previous case observations, which offer information about the frequencies of being an up day (and a down day) for all cases observed previously. The observed frequencies are primarily to classify the observed cases into two groups: the obvious cases and the non-obvious cases. The obvious cases are those having higher frequencies of being an up day or a down day in the past; the others are the non-obvious cases. FFM handles the obvious cases based on the proposition of their consistency [11] in the future.

(4) Propose EFFM to handle the non-obvious cases. The main components in EFFM are the four ANNs, which are trained by using the previous obvious cases picked out via in FFM. EFFM's success is based on two propositions: (1) the information contained in the past obvious cases is sufficient and (2) the generalization ability of ANN is adequate.

(5) EFFM with RN outperforms BP and PN. A possible explanation for the poor generalization of both BP and PN is their associated drawbacks. The poor performance of the PN model is possibly due to PN's inability to handle the nonlinearly separable problems. Although BP can handle the nonlinearly separable problems, BP is generally trapped at a relatively minimal result during the learning stage.

Our research can be extended in the following directions. First, we can compare the prediction performance of our forecast models with other forecast techniques. Second, the same methodology can be applied to examine either long term forecasts or financial markets.

[11] The consistency implies that the emerging obvious cases behave similarly to their previous counterparts.

Acknowledgements

The authors would like to thank the National Science Council of R.O.C. for supporting this work under grant No. NSC 85-2418-H-004-008. Thanks are also due to the editors and two anonymous reviewers for useful comments on an earlier draft of this paper. Several of the references might not be widely available (these include Refs. [19,20,5,13]), but the authors would provide copies of these papers upon request.

Appendix A. Limitations of back propagation

During the learning stage, to minimize the value of the objective function, the back propagation learning algorithm adopts an optimization technique (typically the steepest descent method) to adjust the connection weights. Ideally, a global minimum is desirable. For a more complicated learning task as Fig. 2 depicts, however, the function surface on the weight space usually has numerous local minimums. Thus, it is unavoidable that the learning might be trapped at a local minimum when adopting the steepest descent method.

The optimal network architecture is critical for developing good generalization ability. However, the back propagation learning algorithm cannot determine the optimal hidden layer size by itself. The hidden layer size must be given before the learning starts. The fact that the optimal network architecture is unknown accounts for why the decision can be made only by trial and error.

References

[1] K. Bergerson, D.C. Wunsch, A commodity trading model based on a neural network—Expert system hybrid, Proceedings of the IEEE International Conference on Neural Networks, 1991, pp. I289–I293.

[2] S.C. Blank, Chaos in futures market? a nonlinear dynamical analysis, J. Futures Markets 11 (1991) 711–728.

[3] W.E. Bosarge, Adaptive processes to exploit the nonlinear structure of financial markets, The Santa Fe Institute of Complexity Conference: Neural Networks and Pattern Recognition in Forecasting Financial Markets, Feb. 1991.

[4] T.S. Chande, S. Kroll, The New Technical Trader, Wiley, New York, 1994.

[5] Y. Chiou, S. Liu, R. Tsaih, Applying reasoning neural networks to the analysis and forecast of Taiwan's stock index variation, Taipei Econ. Inquiry, Taipei 34 (2) (1996) 171–200.

[6] G.P. DeCoster, W.C. Labys, D.W. Mitchell, Evidence of chaos in commodity futures prices, J. Futures Markets 12 (1992) 291–305.

[7] M. Frank, T. Stengos, Measuring the strangeness of gold and silver rates of return, Rev. Econ. Studies 56 (1989) 553–567.

[8] G. Grudnitski, L. Osburn, Forecasting S&P and gold futures prices: an application of neural networks, J. Futures Markets 13 (6) (1993) 631–643.

[9] J. Hertz, A. Krogh, R. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley, Reading, MA, 1991.

[10] J.M. Hutchinson, A.W. Lo, T. Poggio, A nonparametric approach to pricing and hedging derivative securities via learning networks, J. Finance 49 (3) (1994) 851–889.

[11] W.Y. Kee, A. Koh, Technical analysis of Nikkei 225 stock index futures using an expert system advisor, Proceedings of the CBOT Conference, 1994.

[12] C. Lai, Forecasting Foreign Stock Index Futures: An Application of Neural Networks, Department of Management Information Systems, National Chengchi Univ., Taipei, Master Thesis, 1996.

[13] H.W. Lin, R. Tsaih, R. Jee, Exploring the relative abilities of neural networks and VAR models in forecasting Taiwan bond prices, Rev. Securities Futures Markets, Taipei 9 (1) (1997) 63–113.

[14] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (1958) 386–408.

[15] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, Parallel Distributed Processing, Vol. 1, MIT Press, Cambridge, MA, 1986, pp. 318–362.

[16] R. Trippi, D. DeSieno, Trading equity index futures with a neural network, J. Portfolio Manage. Fall, (1992) 27–33.

[17] R. Trippi, E. Turban, Neural Networks in Finance and Investing, Probus Publishing, Chicago, 1993.

[18] R. Tsaih, The softening learning procedure, Math. Comput. Modelling 18 (1993) 61–64.

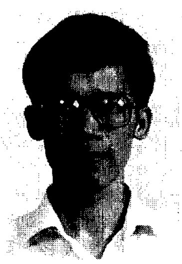
[19] R. Tsaih, The softening learning procedure for the networks with multiple output nodes, MIS Rev. 4 (1994) 89–93.

[20] R. Tsaih, Learning procedure that guarantees obtaining the desired solution of the 2-classes categorization learning problem, The 1st Asia-Pacific Conference on Simulated Evolution and Learning, Korea, 1996, pp. 446–453.

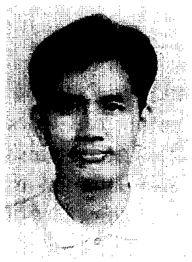
[21] R. Tsaih, Reasoning neural networks, in: S. Ellacott, J. Mason, I. Anderson (Eds.), Mathematics of Neural Networks: Models, Algorithms and Applications, Kluwer Academic Publishers, London, 1997, pp. 366–371.

[22] R. Tsaih, An explanation of reasoning neural networks, Math. Comput. Modelling, 1997, accepted.

[23] Y. Yoon, T. Guimaraes, G. Swale, Integration artificial neural networks with rule-based expert system, Decision Support Syst. 11 (1994) 497–507.



Ray Tsaih, also known as Rua-Huan Tsaih, is an associate professor at National Chengchi University, Taipei, Taiwan. He received his B.S. from National Tsinghua University Taipai, and received his M.S. and Ph.D. in Operations Research in 1991 from University of California, Berkeley. His research interests are Developing new neural networks, applying Neural Networks to Finance, and Interior-point algorithm. His most recent work has been published in Mathematical and Computer Modelling, Review of Securities and Futures Markets, MIS REVIEW, Taipei Economic Inquiry, and Sun Yat-sen Management Review.



Yenshan Hsu is a Professor of Finance at National Chengchi University, Taipei, Taiwan, where he taught courses in investment theory and portfolio management. He received a Ph.D. in Finance from the University of Iowa. His research ares include investment finance, stock market regulations, portfolio management, and the applications of artificial neural networks in finance. His most recent work has been published in Journal of Financial Studies and Pacific-Basin Finance Journal.Charles Lai received a M.S. in Management Information Systems from National Chengchi University, Taipei in 1996.