

## A Rule-based Neural Stock Trading Decision Support System

Seng-cho Timothy Chou  
Department of Information Management  
National Taiwan University  
Taipei, Taiwan, R.O.C.  
chou@ccms.ntu.edu.tw

Chau-chen Yang  
Department of Finance, NTU

Chi-huang Chen  
Department of Computer Science and Information Engineering, NTU

Feipei Lai  
Department of Electrical Engineering and Department of CSIE, NTU

### Abstract

In this paper, we propose an intelligent stock trading decision support system that can forecast the buying and selling signals according to the prediction of short-term and long-term trends using rule-based neural networks. A rule-based neural network allows us to use domain knowledge in the form of inference rules to set up the initial structure of the neural network, and to extract refined domain knowledge from the trained network. With this information, users can understand why and how a decision is made by the system without the need to trust the output of the network blindly. The performance of the proposed system was evaluated by trading the TSEWPI (Taiwan Stock Exchange Weighted Price Index) from 1992 to 1995, and the result was encouraging.

### 1 Introduction

Many factors affect the price trend of a stock. One may attempt to explore the relationship between these factors and stock prices in order to predict the price trend. It is however very difficult for even a human expert to handle this job well because such a relationship is always complex and implicit. Experts often turn to powerful tools to assist them in getting a clearer picture of the relationship. Neural network has emerged as a popular technology in this regard. Neural network is good at establishing the relationship between its input and output even in the presence of noisy data. For this reason, many studies on the application of neural network and related techniques for building up stock trading decision support systems have been proposed [1,2,3,4,5,6,7,8].

While neural network has gained its popularity in the past few years for its powerful learning and non-linear mapping capabilities, it still suffers from two major drawbacks. One is difficulties in embedding existing knowledge into the system. The other is lack of explanation capability. Unlike expert systems, where it is relatively easy to incorporate expert knowledge, it is not easy to build a neural network with embedded a priori information about a stock market. The complex operations involved in training a neural network further hinder us from understanding how a decision is reached by the network. Hence, we have to trust the output of the network blindly, which, however, does not endear the neural network technology to investors or traders. The rule-based neural network model [9] adopted in this paper can avoid the two drawbacks of the traditional neural network model.

To simply put, a rule-based neural network model maps a rule-based inference system into a neural network architecture. As a result, it allows us to embed existing knowledge into the system directly. The resulting system is as powerful as the traditional network in its non-linear mapping capability as well. On the other hand, because of its rule-based architecture, one can also extract rules and the corresponding certainty factors from the trained network. With this information, users can understand why and how a decision is made by the system without the need to trust the output of the network blindly.

In this paper, we propose an intelligent stock trading decision support system that can forecast the buying and selling signals according to the prediction of short-term and long-term trends using rule-based neural networks. This system can provide the user not only the stock trading suggestion but also the rules in the trained network. Users can make use of these rules as the initial rules for other rule-based inference systems as well.

## 2 Rule-based Neural Network

Very briefly, a rule-based neural network is a network combining both connectionistic and symbolic reasoning [9,17]. The main advantages of a rule-based neural network are two-fold: we can use domain knowledge in the form of inference rules to set up the initial structure of the neural network and we can extract refined domain knowledge from the trained network.

Without loss of generality, a rule-based inference system is composed of rules of the following form:

Rule  $R_i$ : If  $\alpha_1(cf_1)$  and  $\alpha_2(cf_2), \dots, \text{and } \alpha_n(cf_n)$  then  $\beta(CF)$   
 where  $\alpha_i$  is a condition,  $\beta$  a conclusion,  
 $cf_i$  the certainty factor of the  $i$ th condition, and  $CF$  the rule's certainty factor.

A set of rules of the form just mentioned can be mapped into a three-layer neural network architecture as follows. We use neurons in the input layer to represent conditions of the rules. Those in the hidden layer represent rules themselves. The conclusions of rules make up the output layer of the network. The interconnection weights between the input layer and the hidden layer come from the certainty factors of corresponding conditions in rules. Finally, the rules' certainty factors become the interconnection weights from the hidden layer to the output layer. With this mapping, we convert a rule-based inference system to a neural network and employ the domain knowledge captured to initialize such a network. The subsequent refinement of this knowledge is done through the powerful learning mechanism of neural networks.

## 3 System Architecture

The proposed stock trading decision support system is composed of five components, namely, preprocessor, dual rule-based neural network module, postprocessor, initial rule generator and rule extractor. The system architecture is shown in Fig. 1.

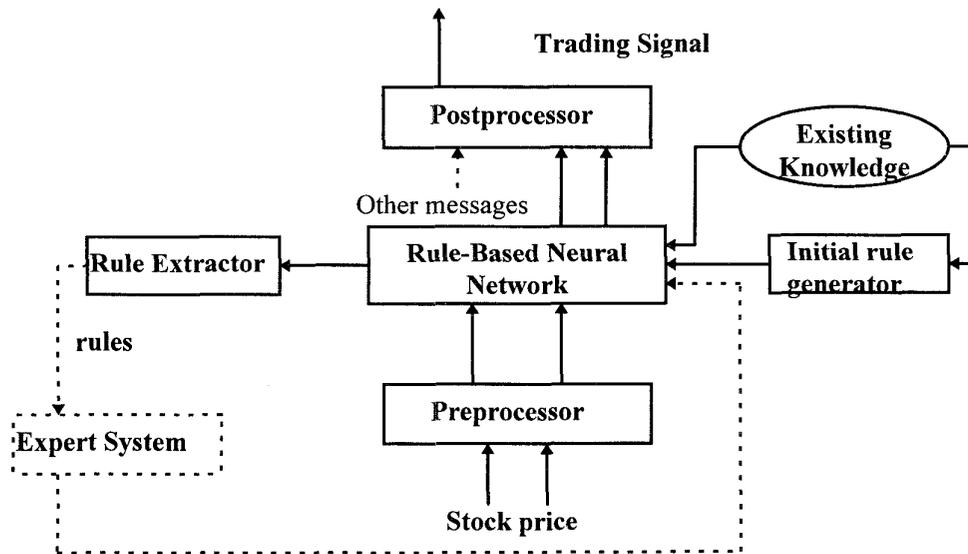


Fig. 1. System Architecture

### 3.1 Preprocessor

The daily information of a stock can be defined as a 5-tuple  $\langle O, H, L, C, V \rangle$ , where  $O$ ,  $H$ ,  $L$  and  $C$  are the opening, highest, lowest, and closing prices of the trading day, respectively, and  $V$  is the daily trading volume. The preprocessor is responsible for turning the raw input stock data into technical indexes to be used by the dual rule-based neural network component. The technical indexes used by these two neural networks include RSI, WMS, KD, MA, PSY and others as listed in Appendix A.

### 3.2 Dual rule-based neural networks

The neural network component is composed of two rule-based neural networks. They are used to predict the price trend of stocks, allowing us to take both the long-term and short-term trends of the market into consideration. The long-term module is responsible for the prediction of long-term price fluctuations with technical indexes computed from the weekly stock data. The desired output is  $WMS_n^{4,1}$  as defined below. This module can predict where the next week's closing price will be located in the price fluctuation range of six consecutive weeks. The short-term module is responsible for the short-term price trend prediction with technical indexes coming from daily stock data. The desired output of this network module is  $PFE_n^{2,3}$ , which is to predict whether an up or down trend exists in the coming three days. Knowing the price fluctuation pattern and the up or down trends of the market allows us to take a long or short position accordingly.

#### 3.2.1 WMS%R

WMS was originally proposed by Larry Williams [13]. This index provides a simple way for us to measure where today's closing price will fit into the recent trading range. We define a forward-calculated  $WMS_n^{j,k}$  as follows:

$$WMS_n^{j,k} = \frac{\text{Max}_{i=n-j}^{n+k}(H_i) - C_n}{\text{Max}_{i=n-j}^{n+k}(H_i) - \text{Min}_{i=n-j}^{n+k}(L_i)}$$

Where  $H_i$  and  $L_i$  are the highest and lowest prices of the  $i$ th trading day respectively.

$WMS_n^{j,k}$  reveals where today's closing price stands in the period starting from  $j$  days in the past to  $k$  days in the future.  $WMS_n^{j,k}$  will be close to 0 (1) when today's closing price is near the peak (trough) of the price fluctuation range in these  $(j+k+1)$  trading days.

#### 3.2.2 PFE

Polarized Fractal Efficiency [14] is a tool to measure the fractal dimension of the price movement over a period of time. It can be used to show how trendy or congested the recent price action is. The forward-calculated  $PFE_n^{j,k}$  is defined below:

$$PFE_n^{j,k} = \frac{(\text{sgn})\sqrt{(C_{n+k} - C_{n-j})^2 + (j+k+1)^2}}{\sum_{i=n-j}^{n+k} \sqrt{(C_{i+1} - C_i)^2 + 1^2}}$$

where  $C_i$  is the closing price of the  $i$ th trading day, and

$$\text{sgn} = \begin{cases} +1 & \text{when } C_{n+k} > C_{n-j} \\ -1 & \text{otherwise} \end{cases}$$

The  $PFE_n^{j,k}$  defined discloses what the price action will be in the next  $k$  trading days.  $PFE_n^{j,k}$  will be close to 1 when the trend of the market is bullish in the future. It drops down to -1 when the market is bearish. At a tied market,  $PFE_n^{j,k}$  is close to 0.

#### 3.2.3 Network architecture and training scheme

The neural network module of the proposed system employs a modified back-propagation algorithm as the learning algorithm. The architecture of the neural network and the initial weights between the input and hidden layers and between the hidden and output layers are dictated by the given domain knowledge in the form of inference rules as discussed above. A sample inference rule is:

If RSI16 is Low (0.922) and WMS12 is high (0.797) then PFE6 is Low (0.846).

To capture the temporal characteristics of a stock market, our neural network adopts a moving-window training scheme [1].

### 3.3 Postprocessor

The postprocessor is responsible for generating trading signals, i.e., buy, sell, or hold, based on the output information from the dual rule-based neural network modules. It makes its suggestion according to some trading rules. An example trading strategy is:

```
If (PFE > upper threshold && WMS < upper threshold) then {
    if ( SHORT exists) then {
        SHORT Exit;
        LONG Entry;
    }
    else if (LONG exists) then HOLD;
    else LONG Entry;
}
```

```
If (PFE < lower threshold && WMS > lower threshold) then {
    if ( LONG exists) then {
        LONG Exit;
        SHORT Entry;
    }
    else if (SHORT exists) then HOLD;
    else SHORT Entry;
}
```

Otherwise HOLD;

which says that if it is an up-trend and it is not near the peak then buy; if it is a down-trend and it is not near the trough then sell.

### 3.4 Rule generator

The purpose of the rule generator is to provide a set of useful rules to initialize the rule-based neural network. We can make use of known rules or, as in our system, take the genetic algorithms approach to extract rules from the historic stock data. Genetic algorithms are search algorithms based on the mechanics of natural selection and natural genetics [11, 15, 16].

### 3.5 Rule extractor

After training, the refined rules or knowledge stored in the network can be extracted easily. Just treat each hidden node as a rule, the input of this hidden node as its conditions, the output of this hidden nodes as the rule's conclusion. The certainty factors are copied from the respective interconnection weights directly.

## 4 Performance Evaluation

Our system was implemented in MS Visual C++ under MS Windows 3.1 on a Pentium PC. The performance of the system was evaluated by trading the TSEWPI of the Taiwan stock market from January 1992 to May 1995. All rates of return were calculated after taking the actual transaction costs for each transaction into consideration. The evaluation criteria included the total number of trades over the test period, the number of profitable trades, the percentage of profitable trades among the total number of trades, the average rate of return per profitable(unprofitable) trade, the maximum rate of return of the profitable(unprofitable) trade, the maximum draw-down, i.e., the cumulative rate of

returns of consecutive unprofitable trades, and the annual rates of return over the test period. The performance of our system and the four major closed-end mutual funds issued in Taiwan are listed in Table 1.

| Period  | 1992    | 1993    | 1994   | 1995    |
|---|---------|---------|--------|---------|
| Total Number of Trades  | 4       | 7       | 9      | 3       |
| Number of Profitable Trades   | 3       | 4       | 6      | 2       |
| Percentage of Profitable Trades (%)                                   | 75%     | 57%     | 67%    | 67%     |
| Average Rate of Return per Profitable Trade(%)                        | 13.39%  | 23.23%  | 6.41%  | 4.10%   |
| Average Rate of Return per Unprofitable Trade(%)                      | -5.88%  | -2.60%  | -3.56% | -1.22%  |
| Maximum Rate of Return per Profitable Trade(%)                        | 17.71%  | 48%     | 9.79%  | 7.87%   |
| Maximum Rate of Return per Unprofitable Trade(%)                      | -5.88%  | -3.80%  | -5.41% | -1.22%  |
| Maximum Draw-down (%)   | -5.88%  | -5.59%  | -5.41% | -1.22%  |
| Annual Rate of Return (%)   | 36.92%  | 102.77% | 29.78% | 6.90%   |
| Total Rate of Return of CSITC Growth Fund(%)                          | -15.39% | 57.73%  | 40.23% | N/A     |
| Total Rate of Return of Kwang Hua Growth Fund(%)                      | -13.42% | 62.12%  | 53.06% | N/A     |
| Total Rate of Return of NITC Fu Yuan Fund(%)                          | -6.29%  | 45.62%  | 51.03% | N/A     |
| Total Rate of Return of Citizens Securities Investment Trust Fund (%) | -7.77%  | 29.10%  | 27.25% | N/A     |
| Annual TSEWPI Change (%)  | -26.60% | 79.76%  | 16.40% | -17.76% |

Table 1. System Performance

The simulation results of buying and selling the daily TSEWPI since 1992 are shown in Fig. 2a and Fig. 2b. Explanations follow.

In 1992, our system detected an up-trend of stock price in January; it therefore took a long entry. When a down-trend was detected in February, the system turned to take a short entry. The systems made a false alarm in August but it had been corrected immediately within a few days. In Table 1, we can see our system yielded a higher return than the four closed-end funds in the bearish market of 1992.

The early part of 1993 was an overwhelming bullish market in Taiwan. The up-trend lasted till April when the stock price sank again. In the last month of 1993, there was another rally, which substantially contributed to the annual gain of the TSEWPI. As shown in Fig 2a, the presence of some false alarms made us suffer some loss but the return gained by our system from the two rallies was substantial enough to cover the loss. So the annual return was superior to all the closed-end funds and the simple buy-and-hold strategy.

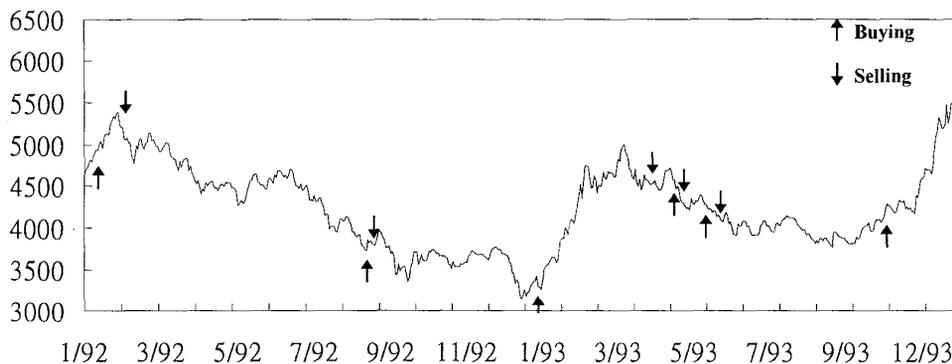


Fig. 2a. Simulation Result from 1992 to 1993

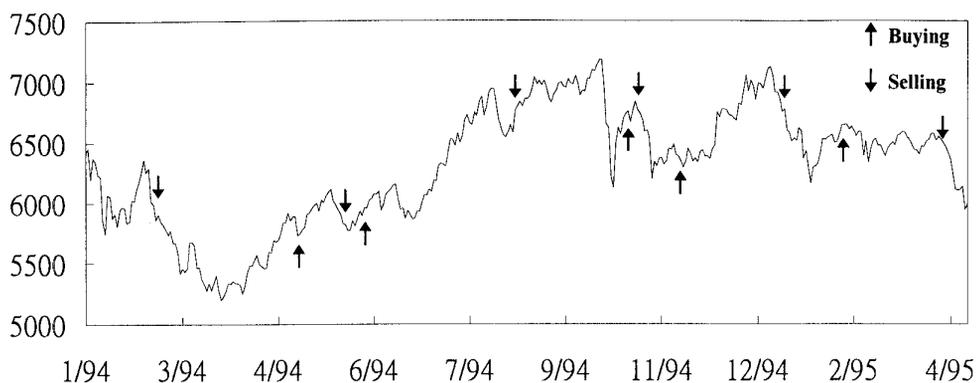


Fig. 2b. Simulation Result from 1994 to 1995

In 1994, the annual return generated by our system beat the market but was lower than some closed-end funds. This was due to the high performance of some particular stocks in that year. While we kept trading the TSEWPI, fund managers of the closed-end funds can adjust their portfolio to increase the rates of returns by selecting those stocks with higher expected returns. In fact, a simple buy-and-hold strategy alone on these few stocks can result in an annual return of 80% to 90%. This explains why closed-end funds performed better than our system in this year.

The first quarter of 1995 was a tied market. It turned to be bearish from April; the TSEWPI collapsed 14% within these few weeks. Our system took a short entry in late April, and got a 6.9% of return in the first four months of 1995.

Finally, a close look at the major rules extracted from the trained neural network module shows that most of the rules match the common sense of human investors. This reflects our network did learn something from the historical stock data. In addition, the rules extracted from our system are superior to what a human expert can give in their explicitness. Rules offered by human experts are often not so well-defined. In particular, for a given set of rules, one might have no idea about how to weigh their importance. By contrast, our system does not suffer from this problem; the explicit certainty factors enable the system to correctly weigh the importance of the rules employed.

## 5 Conclusion

A rule-based neural network is superior to traditional neural network in its explanation capabilities. By extracting and analyzing the knowledge stored in a trained neural network, we know what the network has learned and therefore need not trust the suggestion made by the system blindly. The rules extracted from our system can also be used to support other stock trading systems.

The result of trading TSEWPI of the Taiwan stock market in the period from 1992 to 1995 using the proposed system was encouraging. Our future work focuses on making the system more adaptive to the application domain and applying the system to other markets.

## Appendix A

The definition of the technical indexes used in our system are listed below:

1. Relative Strength Index:

$$RSI_n^k = \frac{\sum_{i=0, C_{n-i} > C_{n-i-1}}^{k-1} (C_{n-i} - C_{n-i-1})}{\sum_{i=0}^{k-1} |C_{n-i} - C_{n-i-1}|}$$

2. Williams Overbought/Oversold Index (WMS%R):

$$WMS_n^k = \frac{\text{Max}_{i=n-k+1}^n(H_i) - C_n}{\text{Max}_{i=n-k+1}^n(H_i) - \text{Min}_{i=n-k+1}^n(L_i)}$$

3. Moving Average and BIAS:

$$MA_n^k = MA_{n-1}^k + \frac{C_n - C_{n-k}}{k}$$

$$BIAS_n^k = \frac{C_n - MA_n^k}{MA_n^k}$$

4. KD

$$K_n^k = \frac{C_n - L_n}{H_n - L_n} \times \frac{1}{k} + K_{n-1}^k \times \frac{k-1}{k}$$

$$D_n^k = K_n^k \times \frac{1}{k} + D_{n-1}^k \times \frac{k-1}{k}$$

5. PSY

$$PSY_n^k = \frac{\sum_{i=n-k+1, C_i > C_{i-1}}^n 1}{k}$$

## References

- [1] Jang, Gia-shuh and Lai, Feipei, "Intelligent Stock Market Prediction System Using Dual Adaptive-Structure Neural Network," Proc. *Second Int'l Conf. on AI Applications On Wall Street*, New York, 1993, pp. 288-294.
- [2] Jang, Gia-shuh and Lai, Feipei, "An Intelligent Trend Prediction and Reversal Recognition System Using Dual Module Neural Network," Proc. *First Int'l Conf. on AI Application On Wall Street*, New York, 1991, pp. 42-51.
- [3] Kimoto, T., Asakawa, K., Yoda, M. and Takeoka, M., "Stock Market Prediction System with Modular Neural Networks," Proc. *IEEE International Joint Conference on Neural Networks*, 1990, pp.1-6.
- [4] Kimoto, T., and Tanigawa, T., " Stock Price Pattern Recognition : A Recurrent Neural Network Approach," Proc. *IEEE International Joint Conference on Neural Networks*, 1990, pp. 215-221.
- [5] Yoda, M., " Predicting the Tokyo Stock Market," *Trading on the Edge*, John Wiley & Sons, Inc, 1994, pp. 66-79.
- [6] Wong, Francis, " Time Series Forecasting Using Back-propagation Neural Networks," *Neurocomputing* , Vol 2, 1991, pp. 147-159.
- [7] Wong, Francis, and Lee, David , "Hybrid Neural Network for Stock Selection," Proc. *Second International Conference on Artificial Intelligence Applications On Wall Street*, New York, 1993, pp. 294 - 301.
- [8] Wong, Francis, and Tan, Clarence , "Hybrid Neural, Genetic, and Fuzzy System," *Trading on the Edge*, John Wiley & Sons, Inc, 1994, pp. 243-261.
- [9] Fu, L. M. , "Knowledge-Based Neural Network", *Neural Network in Computer Intelligence*, McGraw-Hill, 1994, pp. 115 - 143.
- [10] Lacher, R.C., Hruska, S. I., and Kuncicky, D. C., "Back-Propagation Learning in Expert Networks", *IEEE Trans. on Neural Network*, Vol. 3, No. 1, Jan 1992, pp. 62 - 72.
- [11] Goldberg, D.E. , *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [12] Edwards, R. D. and Magee, J., *Technical Analysis of Stock Trends* ,John Magee Inc, 1971.
- [13] Kaufman, P. J., *The New Commodity Trading Systems and Methods* , Wiley: New York, 1987.
- [14] Huannula, H., "Polarized Fractal Efficiency," *Technical Analysis of Stock & commodities* Jan. 1994, pp. 76-81.
- [15] Oliver, J., "Finding Decision Rules with Genetic Algorithms," *AI Expert*, March 1994, pp 33 - 39.
- [16] Bauer, R. J., Jr., *Genetic Algorithms and Investment Strategies*, John Wiley & Sons, Inc, 1994.
- [17] Fu, L.M., "Knowledge-based connectionism for revising domain theories," *IEEE Transactions on Systems, Man , And Cybernetics*, Vol 23, No 1, pp. 173-181.