This article has been accepted for inclusion in a future issue.

Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach

Kai Keng Ang, *Student Member, IEEE*, and Chai Quek, *Member, IEEE*

***Abstract*—This paper investigates the method of forecasting stock price difference on artificially generated price series data using neuro-fuzzy systems and neural networks. As trading profits is more important to an investor than statistical performance, this paper proposes a novel rough set-based neuro-fuzzy stock trading decision model called stock trading using rough set-based pseudo outer-product (RSPOP) which synergizes the price difference forecast method with a forecast bottleneck free trading decision model. The proposed stock trading with forecast model uses the pseudo outer-product based fuzzy neural network using the compositional rule of inference [POPFNN-CRI(S)] with fuzzy rules identified using the RSPOP algorithm as the underlying predictor model and simple moving average trading rules in the stock trading decision model. Experimental results using the proposed stock trading with RSPOP forecast model on real world stock market data are presented. Trading profits in terms of portfolio end values obtained are benchmarked against stock trading with dynamic evolving neural-fuzzy inference system (DENFIS) forecast model, the stock trading without forecast model and the stock trading with ideal forecast model. Experimental results showed that the proposed model identified rules with greater interpretability and yielded significantly higher profits than the stock trading with DENFIS forecast model and the stock trading without forecast model.**

***Index Terms*—Forecasting theory, fuzzy neural networks, rough set theory, stock market, time series.**

I. Introduction

THERE are two major approaches to the analysis of stock market price prediction: Fundamental and technical analyses. Fundamental analysis is the approach of studying the overall economy, industry, financial conditions and management of companies to measure the intrinsic value of a particular security (please refer to [1] for a classical guide to fundamental analysis). This approach uses revenues, earnings, future growth, return on equity, profit margins, and other data to determine a company's underlying value and the potential for future growth of its security. Technical analysis, on the other hand, does not attempt to measure a security's intrinsic value. This approach evaluates securities by analyzing statistics generated by market activity, such as past prices and volume (please refer to [2] for a modern guide to technical analysis). The pioneering technical analysis technique is attributed to C. Dow back in the late 1800s [3]. The efficient market hypothesis (EMH) [4] is generally interpreted to imply that the technical approach to forecasting stock price is invalid, but recent literature presented from a behavioral finance perspective [5] and statistical inference from computational algorithms [6] further exemplified the evidence on the predictability of financial market using technical analysis. Thus technical analysis has recently enjoyed a renaissance and most major brokerage firms publish technical commentary on the market and individual securities.

The main approach in financial forecasting is to identify trends at an early stage in order to maintain an investment strategy until evidence indicates that the trend has reversed. Predictability of security from past real-world data using two of the simplest and most popular trading rules, namely moving average and trading range break-out rules, were first investigated in [3] on the Dow Jones Index. Other techniques used include regression methods and the ARIMA models [7], but these models fail to give satisfactory forecast for some series because of their linear structures and some other inherent limitations [8]. Although there are also ARCH/GARCH models [9] to deal with the nonconstant variance, these models also fail to give satisfactory forecast for some series [8] (please refer to [10]–[12] for a modern guide to the statistical approach to technical financial forecasting). Increasingly applications of artificial intelligence (AI) techniques, mainly artificial neural networks, have been applied to technical financial forecasting [13]–[15] as they have the ability to learn complex nonlinear mapping and self-adaptation for different statistical distributions (please refer to [16] for a review and evaluation of neural networks in technical financial forecasting). An investigation on the nonlinear predictability of security returns from past real-world returns using single layer feed-forward neural network and moving average rules was presented in [17] on the Dow Jones Index. The results showed that evidence of nonlinear predictability in stock market returns can be found by using the past buy and sell signals of the moving average rules. Application of AI techniques in financial forecasting is not restricted only to the technical analysis approach, but has also been applied to the fundamental approach. For example, in the work of [18], a genetic algorithm based fuzzy neural network is trained with additional political, financial, economic factors etc. to formulate trading decisions. A number of research investigations have been published on the application of AI techniques in the technical analysis approach of forecasting stock price, but only a few presented quantitative results on trading performance using real world stock market data [19]–[24]. Saad *et al.* [21] performed analysis of predictability based on a history of closing price of a number of high volatility stocks and consumer stocks using time delay, recurrent and probabilistic neural networks. Leigh *et al.* [22] used neural

Manuscript received May 12, 2005; revised August 11, 2005. This work was supported by the Singapore Millennium Foundation.

The authors are with the Centre for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, Singapore 639798, Singapore (e-mail: ashcquek@ntu.edu.sg).

Digital Object Identifier 10.1109/TNN.2006.875996

networks and genetic algorithm to perform pattern recognition of the bull flag pattern and to learn the trading rules from price and volume of the NYSE Composite Index. Results showed that the forecasting method yielded statistically significant returns that are better than the overall average 20-day horizon price increase. Moody *et al.* [23], [25] used recurrent reinforcement learning without forecasting to train a trading system to trade using past prices of S&P500 stock index while accounting for the effects of transaction costs (please refer to [26] for details on reinforcement learning). Chen *et al.* [24] used a probabilistic neural network (PNN) to forecast the direction of price moment of the Taiwan Stock Index and presented two PNN-guided investment strategies to translate the predicted direction to trading signals. Field and Singh [20] used Pareto evolutionary neural network (Pareto-ENN) to forecast 37 different international stock indexes.

Although neural networks possess the properties required for technical financial forecasting, they cannot be used to explain the causal relationship between input and output variables because of their black box nature. Neuro-fuzzy hybridization synergizes neural networks and fuzzy systems by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks. Neuro-fuzzy hybridization is widely termed as fuzzy neural networks (FNNs) or neuro-fuzzy systems (NFSs) in the literature [27]. NFSs incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF–THEN fuzzy rules. Thus the main strength of NFSs is that they are universal approximators [28]–[30] with the ability to solicit interpretable IF–THEN rules [31]. In recent years, increasing number of research applied NFSs in financial engineering [32]. Some works that applied NFSs in forecasting stock price are [8], [21], [33]–[35].

This paper proposes a novel rough set-based neuro-fuzzy stock trading decision model called stock trading using rough set-based pseudo outer-product (RSPOP). Section II reviews the two main NFSs and outlines the proposed rough set-based neuro-fuzzy approach. Section III reviews the commonly used time-delayed price forecast approach and the time-delayed price difference forecast approach in forecasting stock prices. Section IV presents experimental results of forecasting stock price difference using various neuro-fuzzy systems and neural networks on artificially generated price series data. Section V reviews existing trading models with and without forecast and presents the proposed forecast bottleneck free stock trading with RSPOP forecast model. Section VI presents extensive experimental results using the proposed stock trading with RSPOP forecast model on real world stock market data. The trading profits in terms of portfolio end values are presented and compared against the stock trading with dynamic evolving neural-fuzzy inference system (DENFIS) [36] forecast model, the stock trading without forecast model and the stock trading with ideal forecast model. Finally, Section VII concludes this paper.

II. Rough Set-Based Neuro-Fuzzy Approach

The strength of neuro-fuzzy systems involves two contradictory requirements in fuzzy modeling: interpretability verses accuracy. In practice, one of the two properties prevails. The fuzzy modeling research field is divided into two areas: *linguistic fuzzy modeling* that is focused on interpretability, mainly the Mamdani model [37] given in (1); and *precise fuzzy modeling* that is focused on accuracy, mainly the Takagi–Sugeno–Kang (TSK) model [38] given in (2) and (3) (please refer to [39] for a recent comprehensive coverage on interpretability issues of fuzzy modeling) [31], [39], [40]. Both of these models are investigated in this paper, specifically the Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) (please refer to Section III in [36] for a description of DENFIS) which is based on the precise fuzzy model and the pseudo outer-product based fuzzy neural network (POPFNN) (please refer to [41, Sec. II] for a brief description of POPFNN) which is based on the linguistic fuzzy model

$$R_k: \text{IF } x_1 \text{ is } A_{k,1} \text{ AND} \ldots x_{n_1} \text{ is } A_{k,n_1} \text{ THEN } y \text{ is } B_k \tag{1}$$

$$R_k: \text{IF } x_1 \text{ is } A_{k,1} \text{ AND} \ldots x_{n_1} \text{ is } A_{k,n_1} \text{ THEN } y_k = \mathbf{a_k x} + b_k \tag{2}$$

$$y = \sum_{k=1}^{n_3} w_k(x) y_k \tag{3}$$

where $\mathbf{x}$, y are the input vector $x = [x_1, \ldots, x_{n1}]$ and output value, respectively; $A_{k,i}$, B_k are the linguistic labels with fuzzy sets associated defining their meaning; n_1 number of inputs; and n_3 are tbe number of rules.

The consequents in (1) are linear functions of the inputs whereas the consequents in (2) are simply linguistic labels. Therefore, the TSK model has decreased interpretability but increased representative power compared against the Mamdani model [39]. Recently, a number of research work addressing the issues of the TSK model have been reported [42]–[44]. In contrast, an increased number of fuzzy rules is needed in the Mamdani model to yield the same representative power of the TSK model. Recently, Rough Set methods [45] have been shown to significantly reduce pattern dimensionality and proven to be viable data mining techniques in [46]. This motivated the investigation of the rough set-based neuro-fuzzy approach, which uses the RSPOP algorithm [41] to identify the fuzzy rules in POPFNN. The use of the RSPOP algorithm reduced computational complexity, improved the interpretability by identifying significantly fewer fuzzy rules as well as improved the accuracy of the POPFNN (please refer to [41, Sec. IV] for a detailed description of the RSPOP algorithm).

The RSPOP algorithm consists of three parts. The first part *Rule Identification* identifies only one most influential rule instead of all possible rules from one instant of the training data. The second part *Attribute Reduction* performs feature selection through the reduction of redundant attributes using the concept of knowledge reduction from rough set theory [45]. As there are many possible reducts for a given rule set, this part uses an objective measure to identify reducts that improve rather than deteriorate the inferred consequence after attribute reduction. The third part *Rule Reduction* performs partial feature selection by extending the reduction to rules without redundant attributes.

This article has been accepted for inclusion in a future issue.

TABLE I
FUZZY RULES IDENTIFIED USING RSPOP ON NAKANISHI DATA SET (A) FROM [41]

Rules from RSPOP Rule Indentification

k	$\mathbf{C}_k^I$ c_1	c_2	c_3	c_4	$\mathbf{D}_k^V$ d_1
1	2	3	1	1	1
2	2	1	2	1	3
3	1	2	3	1	2
4	4	2	4	1	1
5	3	2	1	2	2
6	3	3	1	2	1
7	1	1	2	2	4
8	3	1	2	2	2
9	2	2	2	2	2
10	2	2	3	2	2
11	4	3	3	2	1
12	3	2	4	2	2
13	1	3	1	3	2
14	4	3	1	3	1
15	1	2	2	3	3
16	3	2	2	3	1
17	3	3	2	3	1
18	1	1	3	3	3
19	2	1	4	3	2
20	2	2	1	4	2
21	1	2	2	4	3
22	4	3	2	4	1

Rules after RSPOP Attribute Reduction

k	$\mathbf{C}_k^I$ c_1	c_2	c_3	c_4	$\mathbf{D}_k^V$ d_1
1	2	3	0	1	1
2	2	1	0	1	3
3	1	2	0	1	2
4	4	2	0	1	1
5	3	2	0	2	2
6	3	3	0	2	1
7	1	1	0	2	4
8	3	1	0	2	2
9	2	2	0	2	2
10	4	3	0	2	1
11	1	3	0	3	2
12	4	3	0	3	1
13	1	2	0	3	3
14	3	2	0	3	1
15	3	3	0	3	1
16	1	1	0	3	3
17	2	1	0	3	2
18	2	2	0	4	2
19	1	2	0	4	3
20	4	3	0	4	1

Rules after RSPOP Rule Reduction

k	$\mathbf{C}_k^I$ c_1	c_2	c_3	c_4	$\mathbf{D}_k^V$ d_1
1	2	3	0	1	1
2	2	1	0	1	3
3	1	2	0	1	2
4	4	2	0	1	1
5	3	2	0	2	2
6	3	3	0	0	1
7	1	1	0	2	4
8	3	1	0	2	2
9	2	2	0	2	2
10	4	3	0	0	1
11	1	3	0	3	2
12	1	2	0	3	3
13	3	2	0	3	1
14	1	1	0	3	3
15	2	1	0	3	2
16	2	2	0	4	2
17	1	2	0	4	3

An example from [41] is used to help illustrate the RSPOP rule identification and reduction process. Table I shows the initial and reduced rules identified using RSPOP on the Nakanishi example (a) data set [35]. The sets $\mathbf{C}^I = \{c_1, c_2, c_3, c_4\}$ and $\mathbf{D}^V = \{d_1\}$ represent the condition and consequent attributes, respectively, where $c_1 \in \{1,2,3,4\}$, $c_2 \in \{1,2,3\}$, $c_3 \in \{1,2,3,4\}$, $c_4 \in \{1,2,3,4\}$, and $d_1 \in \{1,2,3,4\}$ represent the membership functions. The first part *Rule Identification* identified an initial 22 rules. The second part *Attribute Reduction* identified one reduct $\left(\left(\mathbf{C}^I - \{c_3\}\right), \mathbf{D}^V\right)$ that resulted in no deterioration of the objective measure. Thus, the attribute c_3 is redundant and the number of rules is reduced from 22 to 20 after removing duplicated rules. The third part *Rule Reduction* identified groups of reducible rules that resulted in no deterioration of objective measures and reduced the number of rules from 20 to 17.

III. STOCK PRICE FORECAST

Time series prediction has a diverse range of applications [7], and forecasting stock price is one such application. As time series prediction is concerned with the estimation of equations containing stochastic components, it is modeled by an nth-order linear difference equation with constant coefficients given in (4) [10]

$$y(T) = a_0 + \sum_{i=1}^{n} a_i y(T-i) + x(T) \tag{4}$$

where y is the time series where $y(T)$ represents a value at time instant T; a_1 is one of the arbitrary parameters that do not depend on any values of y or x; and $x(T)$ is the term called the *forcing process* where it can be any function of time, current and lagged values of other variables and/or stochastic disturbances.

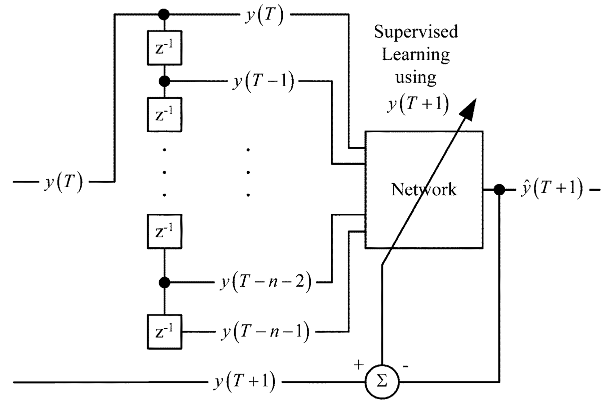


Fig. 1. Time-delayed price forecast approach.

If $x(T)$ is just a sequence of unspecified exogenous variables represented by a stochastic variable $\varepsilon(T)$ instead, then (4) resembles an autoregressive equation [10] and time series prediction is formulated as: Given values $y(T-n), y(T-n+1), \ldots y(T)$; predict $y(T+1)$ as $\hat{y}(T+1)$. Some called this time series prediction using neural networks as time-delayed neural network [21]; some called it recurrent neural network [33]. In the case of nonlinear time series prediction, it is known as the time delay embedding technique [47] and n denotes the embedding dimension [48], [49]. The linear time series prediction in (4) is henceforth referred to as the *time-delayed approach* to discern it from recurrent networks with feedback connections. Several works applied this approach using neural networks or neuro-fuzzy systems [21], [33], [50]–[52]. As a benchmark, the best estimation of (4) is the Random Walk model, which suggests that the day-to-day price change of a stock should have a mean value of zero [10]. Formally, the Random Walk model asserts that the value of the time series should follow the stochastic difference equation in (5) [10]. Therefore, the predicted value $\hat{y}(T)$ using the Random Walk model is simply as given in (6)

$$y(T) = y(T-1) + \varepsilon(T) \tag{5}$$

$$\hat{y}(T) = y(T-1) \tag{6}$$

where $\varepsilon(T)$ is a normal random deviate with an expected mean of zero.

Fig. 1 illustrates the forecasting of stock price using the time-delayed approach. This architecture employs n previous values of the series as input to the network and the predicted value as an output from the network. Each z^{-1} node represents a lag operation for one time instant. The network is trained using a series of known $y(T)$ prior to its use as a time series predictor. Specifically, assume that $y(1), y(2), \ldots, y(N)$ are known, the training data set is formed using $N - n$ of these input–output data tuples.

Incidentally, quite a number of research publications using neural networks based on the time-delayed price forecast approach did not compare against the random walk model (see [16, App. A]), except a recent work in [24]. This is because the time-delayed approach performs direct forecast on a stock price series that is usually non-stationary. A time series is stationary, which is an important property in time series analysis, when the mean value of the series remains constant over the time series. Since many econometric time series are non-stationary, the time-delayed price forecast approach tend to include linear trends and thus deterministic shifts in the out-of-sample forecast period tend to exacerbate forecast errors [12] when compared against the random walk model.

Although financial analysts often adopt trading practices that rely on forecasting the price levels of financial instruments, Sanders and Ritzman [53] showed that people focus on the use of judgemental approach as a basis for forecasting and this approach is about as accurate as the best statistical approaches. O'Connor *et al.* [54] examined the way people forecast trends in time series and how they respond to changes in the trend and presented studies of laboratory experiments that showed individuals have different tendencies and behaviours for upward and downward series. A recent study in [24] suggested that trading strategies guided by forecasts on the direction of price change, which is also the forecast of trends, is more effective and generate greater profits. Empirical results on the index of Taiwan Stock Exchange in [24] showed that this approach using probabilistic neural network obtained higher returns than other investment strategies. In addition, the recent development in the theory of economic forecasting in [11], [12] showed that transforming nonstationary econometric time series to stationarity by differencing and cointegration helped robustify the price forecasts and avoid deterministic shifts in the out-of-sample forecasts which exacerbate the forecast errors. Thus the notion of the human inherent ability in forecasting price trends and the recent development in the theory of economic forecasting motivate the investigation of forecasting of the stock price difference instead of price level.

From (4), the approach in forecasting the price difference is formulated as: Given values $y(T-n), y(T-n+1), \dots y(T)$, compute $\Delta y(T-n+1), \Delta y(T-n+2), \dots \Delta y(T)$, and predict the change in $y(T+1)$ represented as $\Delta\hat{y}(T+1)$ where Δ is the difference operator.

Subtracting $y(T-1)$ from (4) gives

$$\begin{aligned}\Delta y(T) &= a_0 + (a_1-1)y(T-1) + a_2 y(T-2)\cdots \\ &\quad + a_n y(T-n) + \varepsilon(T) \\ &= a_0 + (a_1-1)(y(T-1)-y(T-2)) \\ &\quad + (a_2+a_1-1)y(T-2)\cdots \\ &\quad + a_n y(T-n) + \varepsilon(T) \\ &= a_0 + a_1'\Delta y(T-1) + a_2'\Delta y(T-2)\cdots \\ &\quad + a_n'\Delta y(T-n) + \varepsilon(T) \qquad (7)\end{aligned}$$

where a_i' is another arbitrary parameter; $\varepsilon(T)$ is a stochastic random error term; and $\Delta y(T) = y(T) - y(T-1)$.

Clearly, (7) is a modified version of (4) and $\hat{y}(T+1)$ can then be obtained using (8)

$$\hat{y}(T+1) = y(T) + \Delta\hat{y}(T+1). \qquad (8)$$

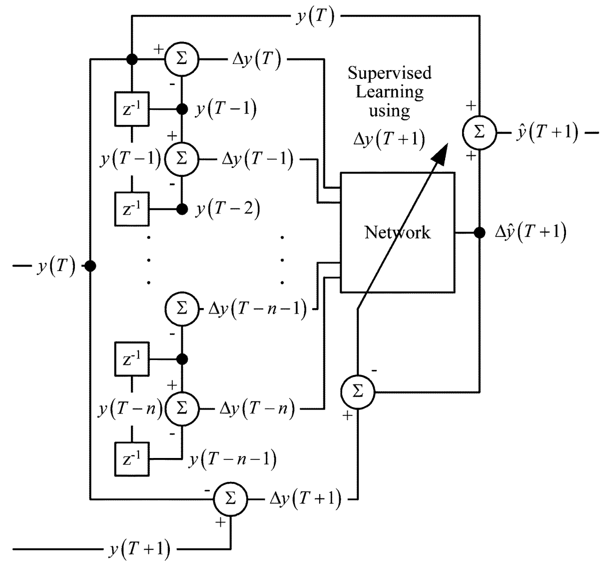


Fig. 2. Time-delayed price difference forecast approach.

This approach uses the difference operator Δ compared to time-delayed price forecast approach in Fig. 1, so we called this the *time-delayed price difference forecast approach*. This approach is illustrated in Fig. 2.

IV. Experiment on Artificial Price Data

In this section, an artificial experimental stock price data set is constructed given in (9) and (10) to investigate the predictive performance of the time-delayed price difference forecast approach using various networks

$$y(T) = y(T-1) + \beta(T-1) + k\varepsilon(T) \qquad (9)$$

$$\beta(T) = \alpha\beta(T-1) + v(T) \qquad (10)$$

where α and k are constants; $\varepsilon(T)$ and $v(T)$ are normal random deviates with zero mean and unit variance.

The stochastically generated artificial experimental data set, which consists of 10000 data samples, is shown in Fig. 3. This data set is constructed similar to the simulation model used in [25] with the parameters $\alpha = 0.9$, $k = 3$ and initial conditions $y(0) = 0$ and $\beta(0) = 0$ using (9) and (10). In this experiment, the first 3000 data samples of the differenced price series are used to train the following networks and the remaining data samples of the differenced price series are used for testing.

1) Feed-forward neural network trained using back-propagation (FFNN-BP) [55].
2) Radial basis function networks (RBFN) [56].
3) Adaptive-network-based fuzzy inference systems (ANFISs) [57] with subtractive clustering [58].
4) Evolving fuzzy neural networks (EFuNNs) [59].
5) Dynamic evolving neural-fuzzy inference system (DENFIS) [36].
6) Pseudo outer-product fuzzy neural network using compositional rule of inference and singleton fuzzifier [60] using RSPOP [41] to identify fuzzy rules (abbreviated as RSPOP-CRI).

This article has been accepted for inclusion in a future issue.

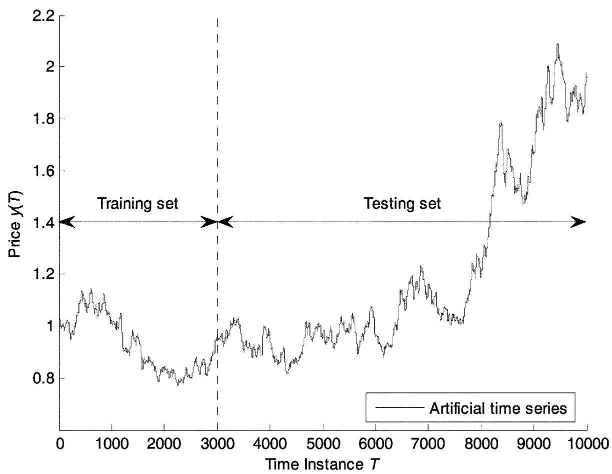


Fig. 3. Artificially generated data set.

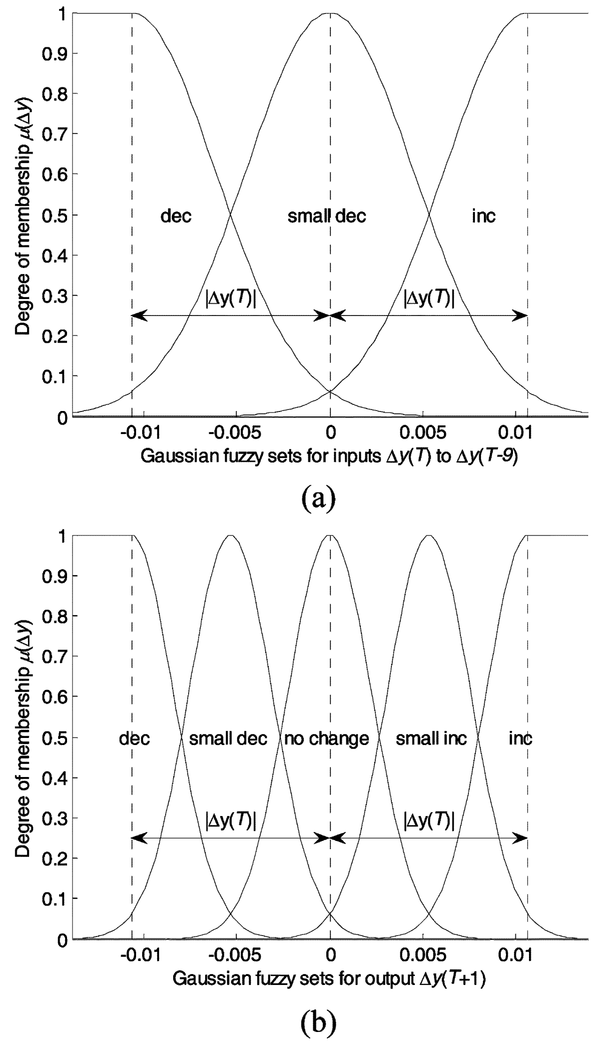


Fig. 4. (a) Input fuzzy sets. (b) Output fuzzy sets of RSPOP-CRI using the time-delayed price difference forecast approach on the artificial data set.

FFNN-BP is configured with 10 input neurons, 30 hidden neurons and 1 output neuron using bipolar sigmoid activation function and is trained for 1000 training iterations using a learning rate of 0.025. RBFN is configured with a default spread of 0.1. ANFIS is configured with a range of influence of 0.3 for subtractive clustering and a maximum epoch of 100. EFuNN and DENFIS are both configured with default parameters. To remove heuristic tuning of training parameters in fuzzy membership learning algorithms of RSPOP-CRI, its fuzzy membership functions are constructed using three input fuzzy sets and five output fuzzy sets that span the range of $\pm|\Delta y(T)|$ from the training data. The fuzzy sets are constructed with open tails so that they cover values that are outside the training range. The fuzzy sets constructed are illustrated in Fig. 4.

This method of constructing the fuzzy sets facilitates the semantic interpretation of each fuzzy set (please refer to [61] and [62] for more details on fuzzy sets) and also adheres more with the fuzzy concept of expressing observations and measurement uncertainties. In this case, the magnitude of the change in price value of the inputs $\Delta y(T)$ to $\Delta y(T+9)$ is represented as three linguistic concepts: $\{$ decrease, no change, increase $\}$ as labelled in Fig. 4(a). Similarly, the magnitude of the predicted change in price value of the output $\Delta y(T+1)$ is represented by five linguistic concepts: $\{$ decrease, small decrease, no change, small increase, increase $\}$ as labelled in Fig. 4(b). The fuzzy rules are derived using the RSPOP algorithm [41] from the training data. RSPOP is used instead of POP specifically for its rapid learning and feature selection capability on large dimension input without heuristic selection of training parameters. The experimental results of using the aforementioned networks for both recall and prediction of the training and test data sets respectively are presented in Table II in terms of the commonly used *mean square error* (*MSE*) and *Pearson product-moment correlation coefficient* (r).

Table II shows that RBFN yielded a higher r, but RBFN yielded almost the same *MSE* as random walk. These results show that the value of r alone is not directly indicative of predictive performance and the value of *MSE* in comparison with random walk is a more suitable performance measure for the prediction of price difference. The results show that the predicted price difference values using neuro-fuzzy systems such as ANFIS, EFuNN, DENFIS and RSPOP-CRI yielded better results than FFNN-BP and RBFN. The results also show that among the networks, EFuNN and RSPOP-CRI yielded the best recall and predictive results respectively. Although EFuNN yielded the best recall results, its predictive results suffered, demonstrating an over-trained phenomenon. Among all the networks, RSPOP-CRI and DENFIS yielded recall and prediction results that are superior to the Random Walk model in terms of MSE.

An examination of the recall and prediction results using only 200 time instances in Figs. 5 and 6 show that the time-delayed price difference forecast approach using DENFIS and RSPOP-CRI are suitable for predicting the price difference of the artificial price series compared against a zero price difference from the Random Walk model. However, the predictive results from just one artificially generated data set is not sufficiently convincing to conclude that DENFIS and RSPOP-CRI are superior choices compared against the other networks.

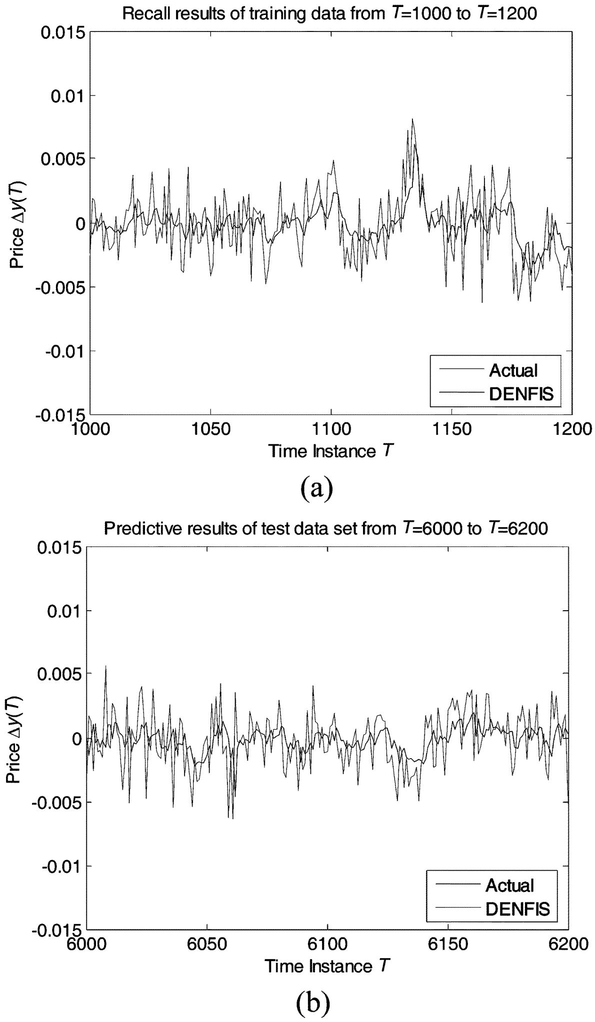


Fig. 5. (a) Magnified recall results and (b) predictive results of forecasting price difference on the artificial data set using DENFIS.

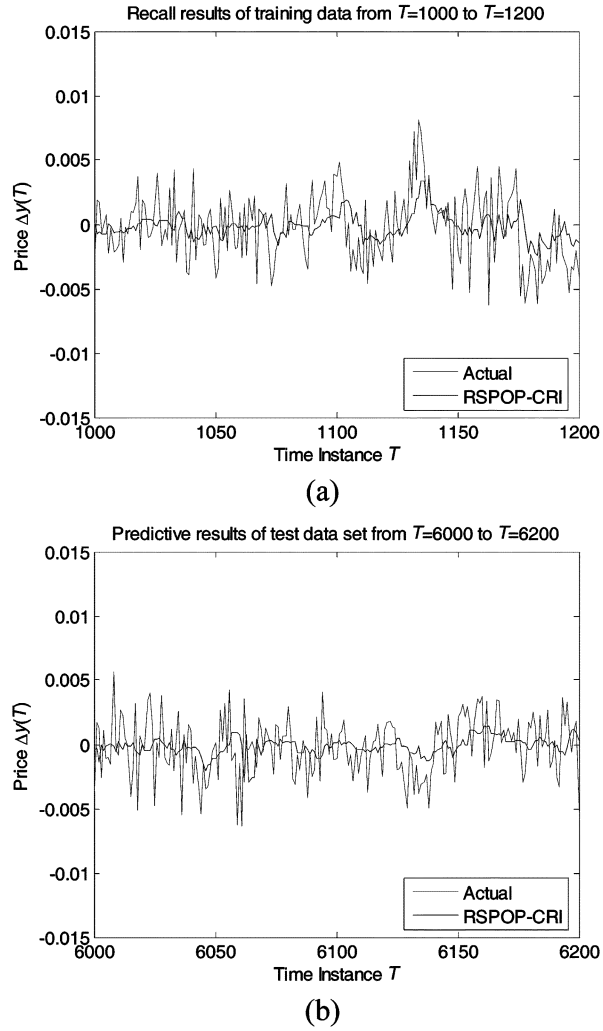


Fig. 6. (a) Magnified recall results and (b) predictive results of forecasting price difference on the artificial data set using RSPOP-CRI.

TABLE II
SUMMARY OF RESULTS OF FORECASTING PRICE DIFFERENCE ON THE ARTIFICIAL DATA SET

Networks	Recall results		Predictive results		Number of fuzzy rules
	MSE (x 10^{-6})	*r*	*MSE* (x 10^{-6})	*r*	
FFNN-BP	16138	0.014	50141	0.060	NaN
RBFN	5.230	0.389	9.530	0.404	NaN
ANFIS	2.613	0.708	15.303	0.139	29
EFuNN	1.698	0.846	14.136	0.132	2167
DENFIS	4.200	0.444	8.655	0.336	153
RSPOP-CRI	4.657	0.331	8.504	0.327	80
Random Walk	5.231	0.000	9.525	0.000	NaN

TABLE III
AVERAGED RESULTS OF FORECASTING PRICE DIFFERENCE ON 20 ARTIFICIALLY GENERATED DATA SET

Networks	Recall results average *MSE* (x 10^{-6})	Predictive results average *MSE* (x 10^{-6})	Average number of fuzzy rules
FFNN-BP	14994 ± 3525	31981 ± 28795	N.A.
RBFN	7.805 ± 5.523	9.694 ± 7.172	N.A.
ANFIS	4.202 ± 4.072	13.064 ± 11.562	37.9 ± 47.7
EFuNN	2.482 ± 2.104	14.844 ± 10.148	2138.2 ± 139.5
DENFIS	5.944 ± 4.122	8.599 ± 6.299	191.6 ± 51.4
RSPOP-CRI	6.846 ± 4.844	8.808 ± 6.545	184.8 ± 64.4
Random Walk	7.816 ± 5.526	9.683 ± 7.172	N.A.

Therefore, further experimental results are collected from additionally generated artificial stock price data sets using (9) and (10). Table III presents the accuracy of the recall and predictive results in terms of average *MSE* and the interpretability of the neuro-fuzzy systems used in terms of average number of fuzzy rules on 20 artificially generated data sets. Standard deviations of the average *MSE* and number of fuzzy rules are also included.

Table III shows that DENFIS and RSPOP-CRI yielded a MSE that is lower than the Random Walk model. It also shows that DENFIS yielded a slightly lower MSE and slightly higher number of fuzzy rules than RSPOP-CRI. However, DENFIS is based on the TSK model [38], [40] used for *precise fuzzy modeling*, which generally yield decreased interpretability but increased accuracy [39]. In contrast, POPFNN-CRI(S) [60] is based on the Mamdani model [37] used for *linguistic fuzzy modeling*, which is focused on interpretability. Thus, in terms of interpretability, RSPOP-CRI is a superior choice than DENFIS since both networks used almost the same number of fuzzy rules. In terms of accuracy, both RSPOP-CRI and

This article has been accepted for inclusion in a future issue.

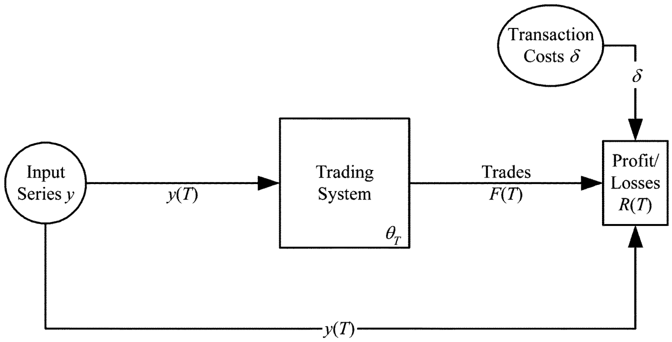


Fig. 7. Trading decision model on technical analysis approach.

DENFIS yielded superior predictive performance in terms of averaged *MSE* compared against the random walk model. Therefore, these results motivate the use of RSPOP-CRI in the time-delayed price difference forecast approach.

This section presented the results based on artificial stock price data set using the time-delayed price difference forecast approach with several networks. Results showed that the use of DENFIS [36] and a novel rough set-based neuro-fuzzy system, namely POPFNN-CRI(S) [60] with fuzzy rules identified using the RSPOP algorithm [41] (abbreviated as RSPOP-CRI), on artificially generated data sets yielded superior results than the Random walk model. A trading model and experimental trading results on real market data is presented in Sections V and VI to assess the trading performance of RSPOP-CRI and DENFIS using the time-delayed price difference forecast approach.

V. Financial Trading Systems

In order to assess the trading performance of the proposed approach, a trading decision model where profits or losses are computable from trading decisions is required. A realistic yet simple computation of profits or losses of a trading decision model based on recurrent reinforcement learning is presented in [25]. Based on this model, two commonly used trading decision models based on the technical analysis approach will be discussed; they are, namely: with price forecast and without price forecast.

A block diagram for a generic trading decision model based on the technical analysis approach is shown in Fig. 7. The technical trading analysis in this model is based on the premise that the market price patterns are assumed to recur in the future [17]. In Fig. 7, the price value of a security is represented as a time series $y(T)$, where y represents a value at time instant T. The action of the trading system is assumed to be one of short, neutral or long, and this action is, respectively, represented by $F(T)$ where $F \in \{-1, 0, 1\}$ [25]. The trading system return is subsequently modeled by portfolio end value using a multiplicative return given in (11) where the transaction cost is assumed to be a fraction δ of the transacted price value [25]

$$R(T) = \{1 + F(T-1)r(T)\}\{1 - \delta|F(T) - F(T-1)|\} \quad (11)$$

where $r(T) = (y(T)/y(T-1)) - 1$; $F(T)$ is the action from the trading system; and δ is the transaction rate.

A number of techniques can be used to generate buy and sell signals using technical analysis techniques. One of the simplest and popular trading rules for deciding when to buy and sell in a security market is the moving average rule [3], [17]. A widely used variant of the moving average rule is the moving average convergence/divergence (MACD) oscillator originally developed by Gerald Appel [5]. MACD uses the crossovers of a *Fast* signal given in (12) and a *Slow* signal given in (13) to indicate a buy or sell signal. The *exponential moving average* (EMA) of a price series is given in (14) [5]. The *Fast* signal in (12) is computed from the difference between the τ_{long} EMA and the τ_{short} EMA of $y(T)$ using the closing price where $\tau_{\text{long}} > \tau_{\text{short}}$. The *Slow* signal in (13) is computed from the τ_{slow} EMA of the *Fast* signal

$$\text{fast}(T) = \text{EMA}^{y}_{\tau_{\text{short}}}(T) - \text{EMA}^{y}_{\tau_{\text{long}}}(T) \quad (12)$$

$$\text{slow}(T) = \text{EMA}^{\text{fast}}_{\tau_{\text{slow}}}(T) \quad (13)$$

$$\text{EMA}^{z}_{\tau} = Kz(T) + (1-K)\text{EMA}^{z}_{\tau}(T-1) \quad (14)$$

where $k = (2/\tau + 1)$; τ is the number of time instance of the moving average; $y(T)$ is the price at the current time instance T; and $\text{EMA}_r(T)$ is the τ EMA of time instance T.

Instead of using the crossover of the fast and slow signal to generate the buy/sell signal, simpler moving average trading rules can be generated using just the slow signal given in (15), where the slow signal as given in (13) is a moving average of the fast signal and the fast signal is the two moving averages of the price level: a long-period average and a short-period average [3]

$$F(T) = sign(\text{slow}(T)) \quad (15)$$

where $F(T)$ is the action from the trading module for time instant T; and $\text{slow}(T)$ is given in (13).

Fig. 8 shows the box plots summarizing the multiplicative profits using just simple moving average trading rules without using RSPOP-CRI price forecast for 100 experiments on artificially generated price series data generated using (9) and (10). The multiplicative profits are computed using (11) and the trading rules are constructed using (12)–(14) and (15) with a heuristically chosen $\text{long} = 26$, $\text{short} = 12$ and $\text{slow} = 9$ (please refer to [3], [63] on the choice of moving average parameters). Comparing the results in Fig. 8 against similar experiments using recurrent reinforcement learning (RLL) neural network in [23], the results in the former yielded a median of about 25 and 6 times while the latter yielded a median of about 6 and 4 times for transaction costs of 0.2% and 0.5% respectively. These results further exemplified the high returns obtainable using simple moving average trading rules compared against other trading strategies as shown in [3].

A block diagram for a generic trading decision model based on the technical analysis approach with forecast is shown in Fig. 9 [25]. In such a system, the forecasting module is trained using supervised learning on a training data set and is then used to forecast an out-of-sample data set. The forecasts are then used as input to a trading module to generate trading signals.

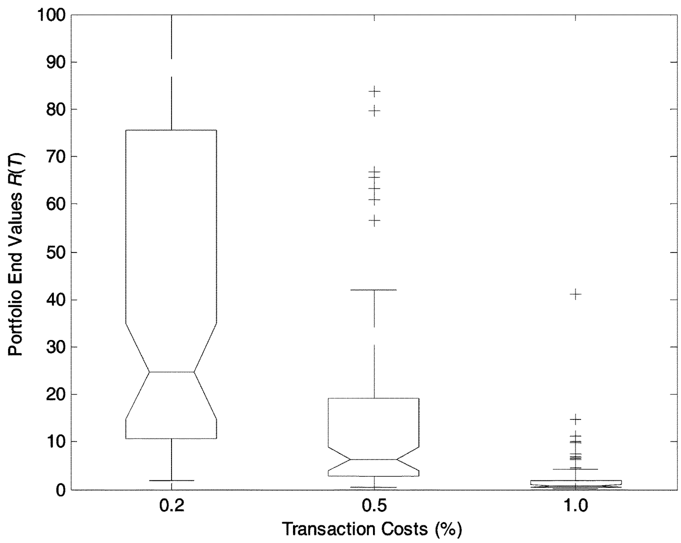


Fig. 8. Box plots of portfolio end values against transaction costs over 100 artificial data set using just simple moving average trading rules.

This common practice of using only the forecasts as input to the trading module results in loss of information or a forecast bottleneck that may lead to suboptimal performance [25]. An example on the use of this model is the work of [24] where trading decision signals are generated using the degree of certainty on the directional forecast of price movements obtained from a trained PNN. Comparing the models in Figs. 7 and 9, the former allows the generation of trading signals from simple trading rules like the moving average trading rules while the latter only allows the generation of trading rules based on the forecasts. Despite the high profitable returns obtainable through the use of simple moving average rules as shown previously, this forecast bottleneck as indicated in [25] impedes the use of the price history of the input series $y(T)$ in the generation of trading signals. Hence, this forecast bottleneck encumbers the trading performance of the forecast module. This motivates the design of a forecast bottleneck free trading decision model that in turn facilitates the research using a synergy of various forecasting methods and trading decision methods.

A novel rough set-based neuro-fuzzy forecast bottleneck free stock trading decision model is proposed and shown in Fig. 10. We called the proposed trading decision model *stock trading using RSPOP*, which is based on the technical analysis approach. Comparing the models in Figs. 9 and 10, the latter simply includes an additional signal of the input series $y(T)$ to the trading module and uses the time-delayed price difference forecast approach. Although this model appears to be an overly simplistic solution to resolve the forecast bottleneck, this model facilitates the synergy of the time-delayed price difference forecast approach with the use of the moving average trading rules. The computation of the trading signal $F(T)$ using both the forecast value $\hat{y}(T+1)$ and $y(T)$ is given in (19). The computation of the moving average signals is given in (16)–(18)

$$\text{fast}'(T+1) = \text{EMA}^{\hat{y}}_{\tau_{\text{short}}}(T+1) - \text{EMA}^{\hat{y}}_{\tau_{\text{long}}}(T+1) \quad (16)$$

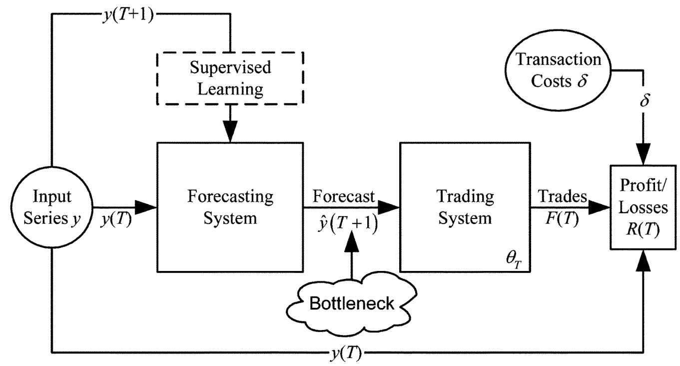


Fig. 9. Trading decision model on technical analysis approach with forecast adapted from [25].

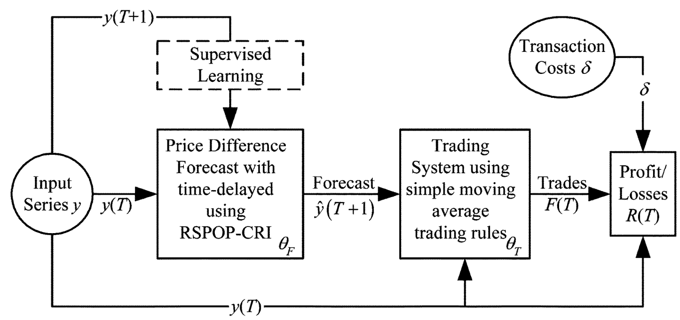


Fig. 10. Stock trading using RSPOP: A forecast bottleneck free trading decision model and RSPOP predictive model.

$$\text{slow}'(T+1) = \text{EMA}^{\text{fast}}_{\tau_{\text{slow}}}(T+1) \quad (17)$$

$$\text{EMA}^{\hat{y}}_{\tau}(T+1) = K\hat{y}(T+1) + (1-K)\text{EMA}^{y}_{\tau}(T) \quad (18)$$

$$F(T) = sign(\text{slow}'(T+1)) \quad (19)$$

where $k = (2/\tau + 1)$; τ is the number of time instance of the moving average; $F(T)$ is the action from the trading module for time instant T; and $\hat{y}(T+1)$ is the forecast price value from the forecast module.

VI. Empirical Results of Stock Trading Using RSPOP

This section presents experimental results in the trading of stock in real world market using the proposed stock trading using RSPOP model shown in Fig. 10. The trading performances of Stock Trading using RSPOP (labelled *with RSPOP forecast*) are benchmarked against Stock Trading using DENFIS (labelled *with DENFIS forecast*), trading decision model without forecast (labelled *without forecast*) and trading decision model with ideal forecast (labelled *with ideal forecast*) on 2 sets of real world market data; namely, Neptune Orient Lines (NOL) and Development Bank of Singapore (DBS). In these experiments, the trading signals for stock trading without forecast model is computed using (12)–(15). The trading signals for stock trading with DENFIS and RSPOP forecast models are computed using (16)–(19) where $\hat{y}(T+1)$ is the forecast obtained using DENFIS and RSPOP respectively. The trading signals for stock trading with ideal forecast model are also computed using (16)–(19), but the forecast $\hat{y}(T+1)$ is replaced with the actual price value of the next time instance $y(T+1)$. All portfolio end values are computed using (11).

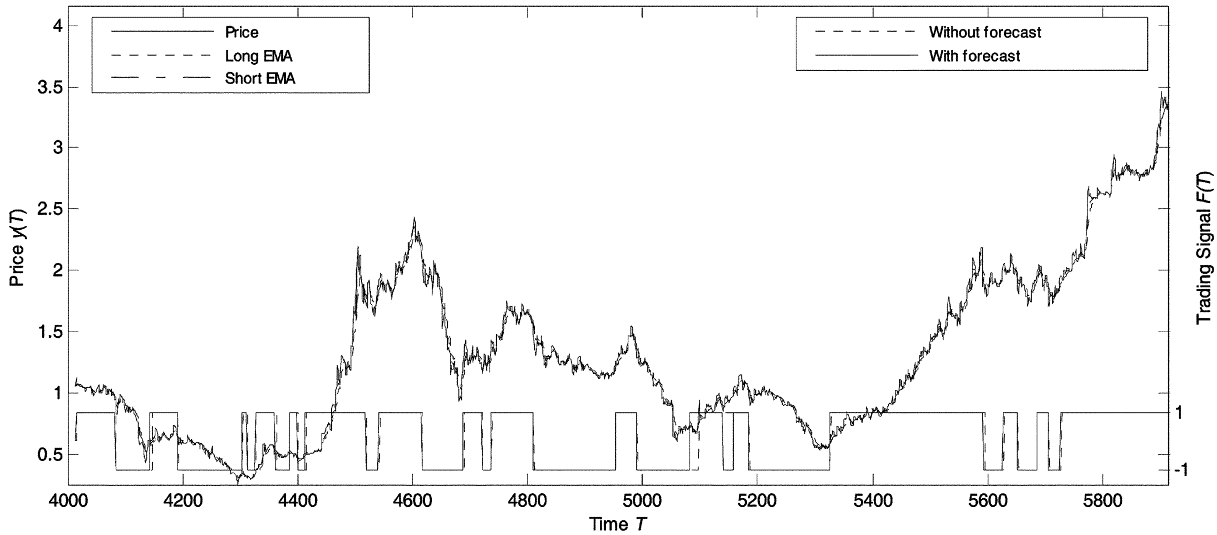


Fig. 11. Price and trading signals on NOL using the stock trading with RSPOP forecast model.

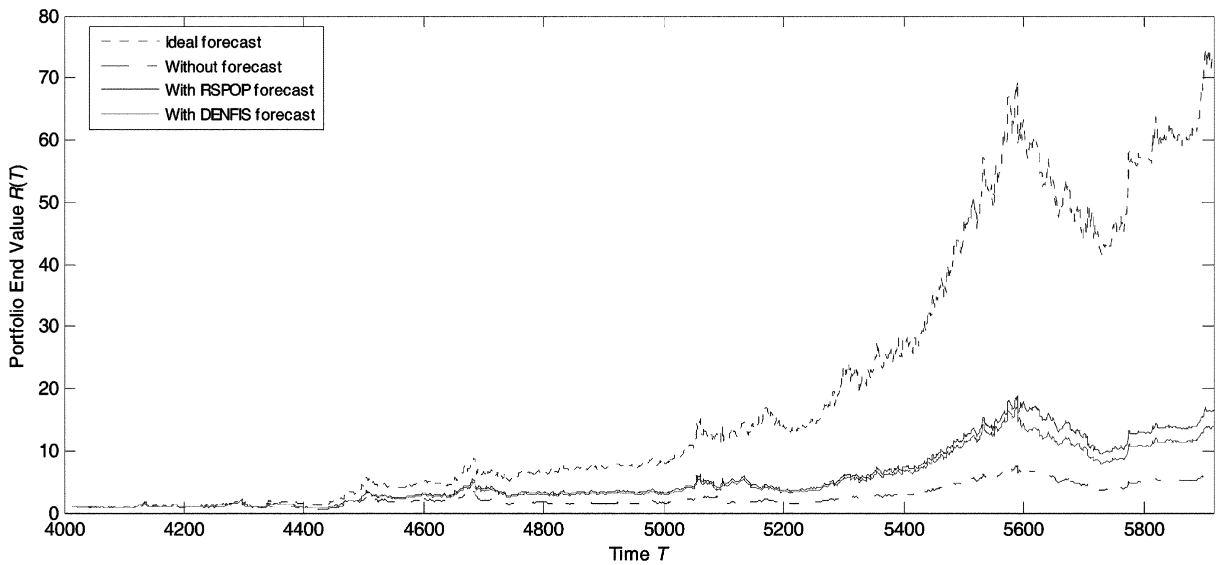


Fig. 12. Portfolio end values $R(T)$ on NOL.

A. NOL

In this experiment, the trading performance of using the proposed stock trading with RSPOP forecast model on the daily closing price of the NOL is compared against the stock trading with DENFIS forecast model, the stock trading without forecast model and the stock trading with ideal forecast model. The forecast models are trained with $n = 10$ previous values of the differenced price series as inputs. The experimental price series consists of 5917 price values obtained from the Yahoo Finance website on the counter N03.SI from the period of January 2, 1980 to March 1, 2005. The in-sample training data set is constructed using the first 4000 data points and the out-of-sample test data set is constructed using the more recent 1917 data points. Trading signals are generated using heuristically chosen moving average parameters $\text{long} = 12$, $\text{short} = 8$, $\text{slow} = 5$ and the portfolio end values $R(T)$ are calculated with a transaction cost of $\delta = 0.2\%$. The moving average trading rule is modified by the introduction of a 1% moving average band. The introduction of this hysteresis band reduces the number of trading signals by eliminating the "whiplash" signals when the short-period and long-period moving averages are close [3].

Figs. 11 and 12 show the out-of-sample price series, the long and short EMAs, the trading signals and the portfolio end values $R(T)$. Starting with a portfolio value of $R(T) = 1.0000$, at the end of the experiment, the stock trading without forecast model yielded a portfolio end value of $R(T) = 6.3027$, the proposed stock trading with RSPOP forecast model yielded $R(T) = 16.5980$, the stock trading with DENFIS forecast model yielded $R(T) = 13.8675$ and the stock trading with ideal forecast model yielded $R(T) = 73.2888$. The results show that the stock trading with forecast models yielded higher returns than the stock trading without forecast model and yielded lower returns than the stock trading with ideal forecast model. Overall, the use of the proposed stock trading with RSPOP forecast model yielded an increase of $R(T) = 10.2953$ in

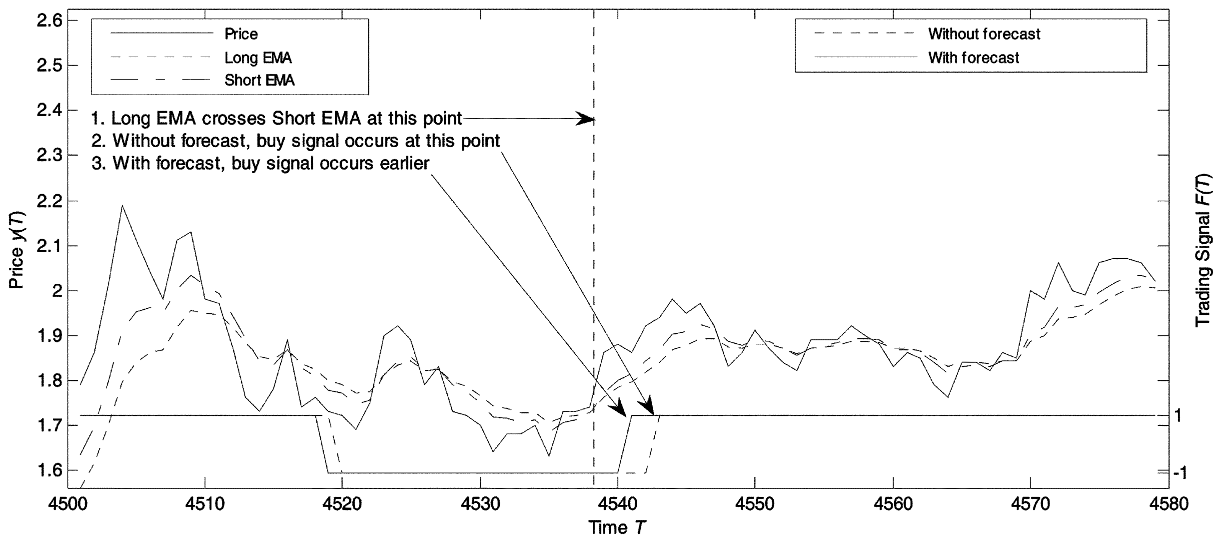
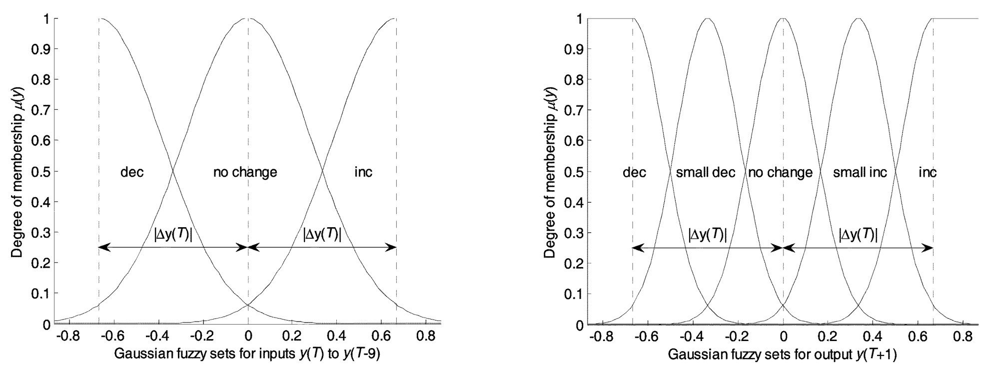


Fig. 13. Magnified results of Fig. 11 from time $T = 4400$ to $T = 4600$.



(a) (b)

Rule 1: if ($y(T\text{-}2)$ is no change)	and ($y(T\text{-}1)$ is dec)	and ($y(T)$ is dec)	then $y(T+1)$ is small inc
Rule 2: if ($y(T\text{-}2)$ is no change)	and ($y(T\text{-}1)$ is no change)	and ($y(T)$ is dec)	then $y(T+1)$ is small inc
Rule 3: if ($y(T\text{-}2)$ is dec)	and ($y(T\text{-}1)$ is dec)	and ($y(T)$ is no change)	then $y(T+1)$ is no change
Rule 4: if ($y(T\text{-}2)$ is dec)	and ($y(T\text{-}1)$ is no change)	and ($y(T)$ is no change)	then $y(T+1)$ is small dec
Rule 5: if ($y(T\text{-}2)$ is no change)	and ($y(T\text{-}1)$ is no change)	and ($y(T)$ is no change)	then $y(T+1)$ is no change
Rule 6: if ($y(T\text{-}2)$ is inc)	and ($y(T\text{-}1)$ is no change)	and ($y(T)$ is no change)	then $y(T+1)$ is no change
Rule 7: if ($y(T\text{-}2)$ is dec)	and ($y(T\text{-}1)$ is inc)	and ($y(T)$ is no change)	then $y(T+1)$ is no change
Rule 8: if ($y(T\text{-}2)$ is no change)	and ($y(T\text{-}1)$ is dec)	and ($y(T)$ is inc)	then $y(T+1)$ is no change

(c)

Fig. 14. (a) Fixed input fuzzy membership functions. (b) Fixed output fuzzy membership functions. (c) Fuzzy rules identified using RSPOP in the stock trading with RSPOP forecast model on NOL.

portfolio end value trading NOL stock compared with the stock trading without forecast model. The stock trading with DENFIS forecast model yielded a lower increase of $R(T) = 7.5648$ in portfolio end value trading NOL stock compared with the stock trading without forecast model.

Fig. 13 shows the magnified results of Fig. 11 from time $T = 4500$ to $T = 4580$. The moving average trading signals are generated from the Slow EMA ($\tau_{\text{slow}} = 5$) of the difference between the Long EMA ($\tau_{\text{long}} = 12$) and Short EMA ($\tau_{\text{short}} = 8$) of the NOL price values. When the Short EMA crosses the Long EMA from below as indicated in Fig. 13, it indicates a rising price trend and, thus, the a buy signal ($F(T) = 1$) is generated from the Slow EMA. The stock trading with RSPOP forecast model generated the buy signal earlier through the use of the forecast value $\hat{y}(T+1)$. Hence the proposed stock trading with RSPOP forecast model and the stock trading with DENFIS forecast model yielded a much higher portfolio end value than the stock trading without forecast model. However, since the stock trading with forecast models are unable to forecast with exact accuracy, these models yielded a lower portfolio end value than the stock trading with ideal forecast model. This result is intuitive since it is impossible to obtain higher returns than the stock trading with ideal forecast model, which uses $y'(T+1) = y(T+1)$ the actual future price value.

Fig. 14 shows the fixed fuzzy membership functions and fuzzy rules identified using RSPOP in the stock trading with RSPOP forecast model. A total of eight fuzzy rules are identified by RSPOP in contrast to nine fuzzy rules identified by DENFIS. Since POPFNN-CRI(S) neuro-fuzzy system [60] using RSPOP [41] is a Mamdani fuzzy model [37] whereas

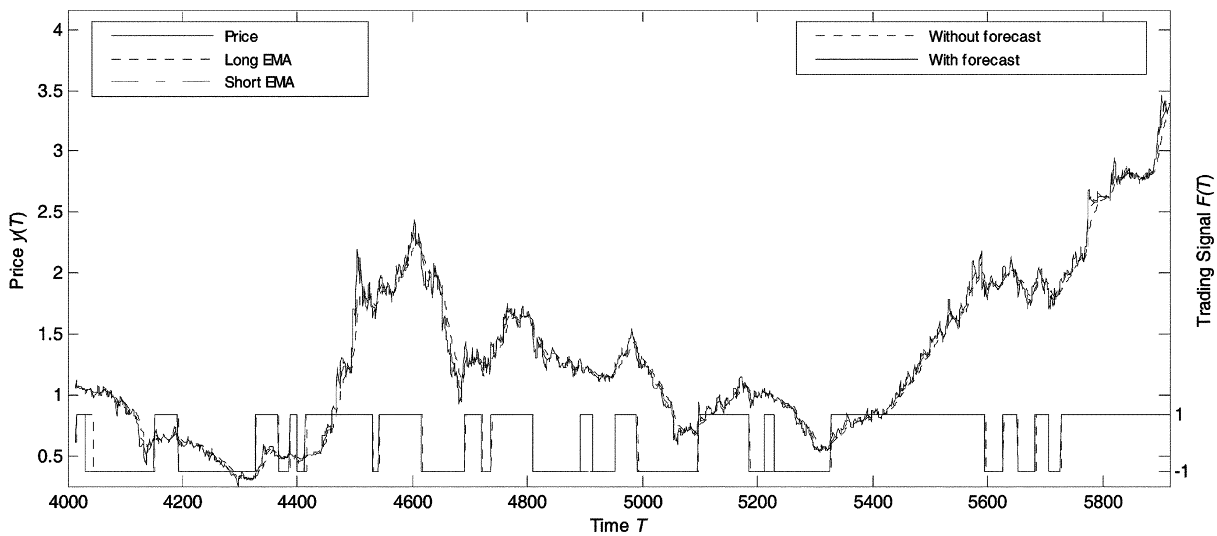


Fig. 15. Price and trading signals on NOL using the stock trading with RSPOP forecast model but using another set of moving average trading parameters.

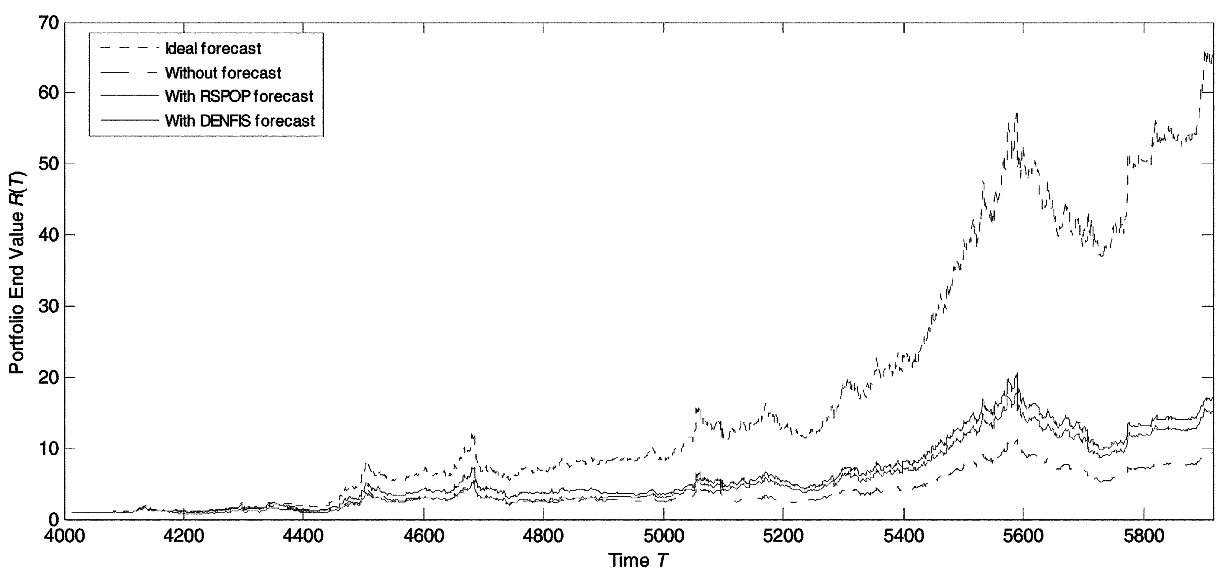


Fig. 16. Portfolio end values $R(T)$ on NOL using another set of moving average parameters.

DENFIS neuro-fuzzy system [36] is a TSK fuzzy model [40], the rules identified in the former are more interpretable than DENFIS model [39]. Moreover, Fig. 14 shows that only the three of the time-delayed price values ($y(T)$, $y(T-1)$ and $y(T-2)$) are used in the stock trading with RSPOP forecast model instead of the all the $n = 10$ time-delayed price values to forecast the future price values. Since the heuristically chosen value of $n = 10$ does not represent the embedding dimension of the NOL price series, the results show that the use of the RSPOP algorithm enabled the computation of an appropriate value of the embedding dimension. In contrast, the fuzzy rules in the stock trading with DENFIS forecast model used all ten time-delayed price values and the fuzzy rules are too complex to be listed. Therefore, the results showed that the proposed stock trading with RSPOP forecast model not only yielded a higher portfolio end value than the stock trading with DENFIS forecast model, but also yielded more interpretable rules by computing an appropriate embedding dimension compared against the stock trading with DENFIS forecast model.

B. NOL—Different Trading Parameters

The experiment is repeated using another set of moving average parameters $\mathrm{long} = 20$, $\mathrm{short} = 10$, $\mathrm{slow} = 4$, and the portfolio end values $R(T)$ are calculated with the same transaction cost of $\delta = 0.2\%$. Figs. 15 and 16 show the out-of-sample price series with long and short EMAs, the trading signals and the portfolio end values $R(T)$ using this set of moving average parameters.

Starting with a portfolio value of $R(T) = 1.0000$, at the end of the experiment, the stock trading without forecast model yielded a portfolio end value of $R(T) = 9.2673$, the proposed stock trading model RSPOP forecast model yielded $R(T) = 17.0362$, the stock trading with DENFIS forecast model yielded $R(T) = 15.2643$ and the stock trading with

This article has been accepted for inclusion in a future issue.

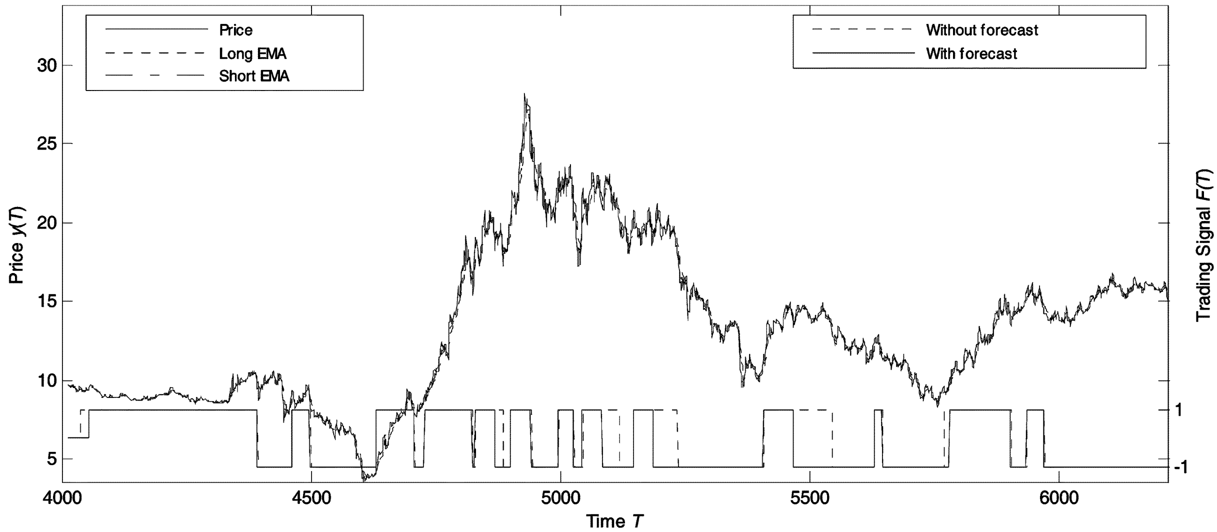


Fig. 17. Price and trading signals on DBS using the stock trading with RSPOP forecast model.

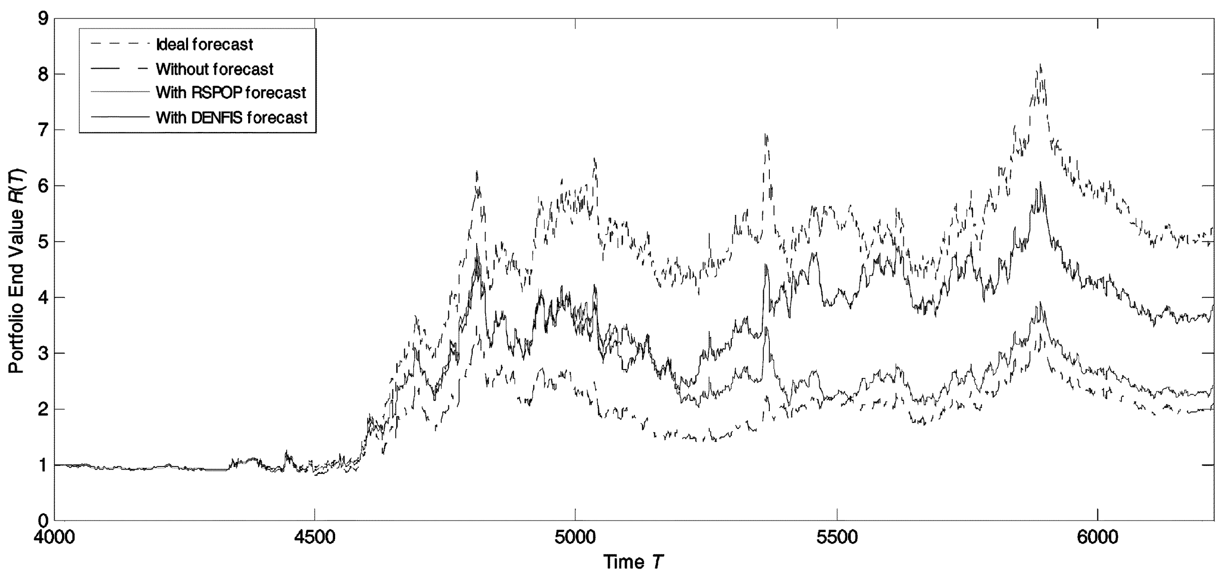


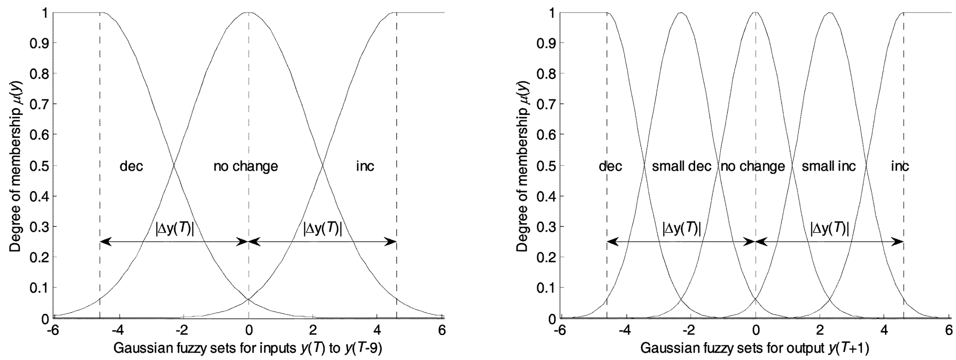
Fig. 18. Portfolio end values $R(T)$ on DBS.

ideal forecast model yielded $R(T) = 65.0778$. Overall, the use of the proposed stock trading with RSPOP forecast model yielded an increase of $R(T) = 7.7689$ in portfolio end value trading NOL stock compared with the stock trading without forecast model. The stock trading with DENFIS forecast model yielded a lower increase of $R(T) = 5.9970$ in portfolio end value trading NOL stock compared with the stock trading without forecast model. Comparing the results from this experiment using moving average parameters $\mathrm{long} = 20, \mathrm{short} = 10$, $\mathrm{slow} = 4$ against the results from the previous experiment using moving average parameters $\mathrm{long} = 12$, $\mathrm{short} = 8$, $\mathrm{slow} = 5$; the results showed that the use of different moving average parameters yielded different portfolio end values for the stock trading without forecast model, the stock trading with forecast models as well as the stock trading with ideal forecast model. The results also showed that the stock trading with forecast models yielded higher returns than the stock trading without forecast model and yielded lower returns than the stock trading with ideal forecast model. Most significantly, the results showed that the proposed stock trading with RSPOP forecast model yielded a higher portfolio end value than the stock trading without forecast model as well as the stock trading with DENFIS forecast model, despite using a different set of moving average parameters.

C. Development Bank of Singapore (DBS)

In this experiment, the trading performance of using the proposed stock trading with RSPOP model on the daily closing price of the DBS is compared against the stock trading with DENFIS forecast model, the stock trading without forecast model and the stock trading with ideal forecast model. The forecast models are trained with $n = 10$ previous values of the differenced price series as inputs. The experimental price series consists of 6222 price values obtained from Yahoo Finance web site on the counter D05.SI from the period of January 2,

This article has been accepted for inclusion in a future issue.



(a) (b)

Rule 1: if ($y(T$-1) is dec)	and ($y(T)$ is dec)	then $y(T$+1) is small inc
Rule 2: if ($y(T$-1) is no change)	and ($y(T)$ is dec)	then $y(T$+1) is dec
Rule 3: if ($y(T$-1) is no change)	and ($y(T)$ is no change)	then $y(T$+1) is no change
Rule 4: if ($y(T$-1) is inc)	and ($y(T)$ is no change)	then $y(T$+1) is no change
Rule 5: if ($y(T$-1) is dec)	and ($y(T)$ is inc)	then $y(T$+1) is small dec

(c)

Fig. 19. (a) Fixed input fuzzy membership functions. (b) Fixed output fuzzy membership functions. (c) Fuzzy rules identified using RSPOP in the stock trading with RSPOP forecast model on DBS.

1980 to March 1, 2005. The in-sample training data set is constructed using the first 4000 data points and the out-of-sample test data set is constructed using the more recent 2222 data points. Trading signals are generated using heuristically chosen moving average parameters $\mathrm{long} = 12$, $\mathrm{short} = 8$, $\mathrm{slow} = 5$ with a 1% moving average band and the portfolio end values $R(T)$ are calculated with a transaction cost of $\delta = 0.2\%$.

Figs. 17 and 18 show the out-of-sample price series, the long and short EMAs, the trading signals and the portfolio end values $R(T)$. Starting with a portfolio value of $R(T) = 1.0000$, at the end of the experiment, the stock trading without forecast model yielded a portfolio end value of $R(T) = 2.0821$, the proposed stock trading with RSPOP forecast model yielded $R(T) = 3.8582$, the stock trading with DENFIS forecast model yielded $R(T) = 2.4138$ and the stock trading with ideal forecast model yielded $R(T) = 5.3635$. The results show that the stock trading with forecast models yielded higher returns than the stock trading without forecast model and yielded lower returns than the stock trading with ideal forecast model. Overall, the use of the proposed stock trading with RSPOP forecast model yielded an increase of $R(T) = 1.7760$ in portfolio end value trading DBS stock compared with the stock trading without forecast model. The stock trading with DENFIS forecast model yielded a lower increase of $R(T) = 0.3316$ in portfolio end value trading DBS stock compared with the stock trading without forecast model. Fig. 19. shows the fixed fuzzy membership functions and fuzzy rules identified using RSPOP in the stock trading with RSPOP forecast model.

Fig. 19 showed that a total of 5 fuzzy rules are identified by RSPOP whereas a total of 10 fuzzy rules are identified by DENFIS. Fig. 19. shows that only two of the time-delayed price values ($y(T)$ and $y(T-1)$) are used in the stock trading with RSPOP forecast model instead of the all the $n = 10$ time-delayed price values to forecast the future price values. Since the heuristically chosen value of $n = 10$ does not represent the embedding dimension of the NOL price series, the results showed again that the use of the RSPOP algorithm enabled the computation of an appropriate value of the embedding dimension. In contrast, the fuzzy rules in the stock trading with DENFIS forecast model used all 10 time-delayed price values and the fuzzy rules are too complex to be listed.

The results obtained on the NOL and DBS stock prices show that the proposed stock trading with RSPOP forecast model yielded portfolio end values of $R(T) = 17.0362$ and $R(T) = 3.8582$, respectively, using the same moving average parameters $\mathrm{long} = 12$, $\mathrm{short} = 8$, and $\mathrm{slow} = 5$. This translates to a profit of 16.0362 and 2.8582 times the initial capital invested in NOL and DBS, respectively. Although a significant amount of profit is obtained, the results once again exemplified the fact that profit or loss strongly depends on the trading module and its parameters, as well as the stock counter selected. These results obtained showed a fair representation of the real world market because the effectiveness of the moving average trading rule relies heavily on the heuristic choice of moving average parameters for a particular data set. Hence, it is fair to generalize that profit or loss strongly depends on the trading module. Nevertheless, the results obtained using the proposed stock trading using RSPOP forecast model in all three experiments using real market data yielded significantly higher portfolio end values than the stock trading without forecast model. This showed that using the proposed stock trading with RSPOP forecast model improved the portfolio end value yielded compared against a stock trading without forecast model. The results also showed that the use of RSPOP algorithm enables the identification of the embedding dimension of the stock price series, which produced a smaller number of interpretable rules.

VII. Conclusion

A novel rough set-based neuro-fuzzy stock trading decision model called *stock trading using RSPOP* is proposed in this paper. The proposed stock trading model circumvents the forecast bottleneck and synergizes the time-delayed price difference forecast approach with simple moving average rules for generating trading signals. Experimental results on forecasting stock price difference on artificially generated price series data showed that two neuro-fuzzy systems, namely DENFIS and a novel rough set-based neuro-fuzzy system called the RSPOP FNN yielded superior predictive performance than the well-established random walk model. As the trading profits is more important to an investor than statistical performance, these neuro-fuzzy systems are incorporated as the underlying predictor model in forecasting stock price difference with a forecast bottleneck free trading decision model using moving average trading rules. Experimental results based on real world stock market data, namely the stock prices of NOL and DBS, are presented. Experimental trading profits in terms of portfolio end values of the proposed stock trading with RSPOP forecast model are benchmarked against the stock trading with DENFIS forecast model, the stock trading without forecast model and the stock trading with ideal forecast model. Experimental results showed that the proposed stock trading with RSPOP forecast model identified fewer fuzzy rules as well as yielded higher portfolio end values compared against the stock trading with DENFIS forecast model. Thus results showed that using RSPOP as the underlying predictive model identified rules with greater interpretability and accuracy. Experimental results also provided evidence that profitability strongly depends on the trading module and its parameters as well as the stock counter selected. Despite the different yield in profits, the experimental results consistently showed that the proposed stock trading with RSPOP forecast model yielded significantly higher profits than the stock trading without forecast model.

The potential of the proposed stock trading using RSPOP model discussed in this paper is notable as it demonstrates the feasible application of POPFNN in the area of financial prediction. Although the proposed approach is not proven to assure profits from real world markets, this paper presented a forecast bottleneck free trading decision model that opened new opportunities for the synergy of this approach with various trading decision schemes. Design of intelligent trading decision schemes that synergize with forecasting to yield optimal trading performance on real world stock market data now appears to be a promising research area. The research presented in this paper is in line with the research direction of Centre for Computational Intelligence (C2i) [64], formerly the Intelligent Systems Laboratory, at the Nanyang Technological University, Singapore. C2i undertakes active research in intelligent neuro-fuzzy systems for the modelling of complex, nonlinear and dynamic problem domains. Examples of neural and neuro-fuzzy systems developed at C2i are MCMAC [65], GenSoFNN [66], and POPFNN [67].

REFERENCES

[1] B. Graham and D. Dodd, *Security Analysis: Principles and Techniques*. New York: McGraw-Hill, 1940.

[2] R. D. Edwards, J. Magee, and W. H. C. Bassetti, *Technical Analysis of Stock Trends*. Boca Raton, FL: Saint Lucie Press, 2001.

[3] W. Brock, J. Lakonishok, and B. LeBaron, “Simple technical trading rules and the stochastic properties of stock returns,” *J. Finance*, vol. 47, no. 5, pp. 1731–1764, Dec. 1992.

[4] E. F. Fama, “Efficient capital markets: II,” *J. Finance*, vol. 46, no. 5, pp. 1575–1617, Dec. 1991.

[5] T. Plummer and A. Ridley, *Forecasting Financial Markets: The Psychology of Successful Investing*. London, U.K.: Kogan, 2003.

[6] A. W. Lo, H. Mamaysky, and J. Wang, “Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation,” *J. Finance*, vol. 55, no. 4, pp. 1705–1765, Aug. 2000.

[7] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[8] C.- S. Lin, H. A. Khan, and C.- C. Huang, “Can the neuro fuzzy model predict stock indexes better than its rivals?,” in *Proc. CIRJE*, Aug. 2002, CIRJE-F-165.

[9] R. Engle, “Autoregressive conditional heteroskedasticity with estimates of the variance of united kingdom inflation,” *Econometrica*, vol. 50, no. 4, pp. 987–1007, Jul. 1982.

[10] W. Enders, *Applied Econometric Time Series*. Hoboken, NJ: Wiley, 2004.

[11] M. P. Clements and D. F. Hendry, *Forecasting Economic Time Series*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[12] ——, *Forecasting Non-Stationary Economic Time Series*. Cambridge, MA: MIT Press, 1999.

[13] Y. S. Abu-Mostafa, A. F. Atiya, M. Magdon-Ismail, and H. White, “Introduction to the special issue on neural networks in financial engineering,” *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 653–656, Jul. 2001.

[14] R. R. Trippi and E. Turban, *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance*. Chicago, IL: Probus, 1993.

[15] A.- P. Refenes, *Neural Networks in the Capital Markets*. New York: Wiley, 1995.

[16] M. Adya and F. Collopy, “How effective are neural networks at forecasting and prediction? A review and evaluation,” *J. Forecasting*, vol. 17, no. 5–6, pp. 481–495, Sep. 1998.

[17] R. Gencay, “The predictability of security returns with simple technical trading rules,” *J. Empir. Finance*, vol. 5, no. 4, pp. 347–359, Oct. 1998.

[18] R. J. Kuo, C. H. Chen, and Y. C. Hwang, “An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network,” *Fuzzy Sets Syst.*, vol. 118, no. 1, pp. 21–45, Feb. 2001.

[19] T. Van Gestel, J. A. K. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle, “Financial time series prediction using least squares support vector machines within the evidence framework,” *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 809–821, Jul. 2001.

[20] J. E. Fieldsend and S. Singh, “Pareto evolutionary neural networks,” *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 338–354, Mar. 2005.

[21] E. W. Saad, D. V. Prokhorov, and D. C. Wunsch II, “Comparative study of stock trend prediction using time delay, recurrent, and probabilistic neural networks,” *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1456–1470, Nov. 1998.

[22] W. Leigh, R. Purvis, and J. M. Ragusa, “Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: A case study in romantic decision support,” *Decision Support Syst.*, vol. 32, no. 4, pp. 361–377, Mar. 2002.

[23] J. Moody and M. Saffell, “Learning to trade via direct reinforcement,” *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 875–889, Jul. 2001.

[24] A.-S. Chen, M. T. Leung, and H. Daouk, “Application of neural networks to an emerging financial market: Forecasting and trading the Taiwan Stock Index,” *Comput. Oper. Res.*, vol. 30, no. 6, pp. 901–923, May 2003.

[25] J. Moody, L. Wu, Y. Liao, and M. Saffell, “Performance functions and reinforcement learning for trading systems and portfolios,” *J. Forecasting*, vol. 17, no. 5–6, pp. 441–470, Sep. 1998.

[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[27] S. Mitra and Y. Hayashi, “Neuro-fuzzy rule generation: Survey in soft computing framework,” *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 748–768, May 2000.

[28] J. L. Castro, “Fuzzy logic controllers are universal approximators,” *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 4, pp. 629–635, Apr. 1995.

[29] H. Ying, Y. Ding, S. Li, and S. Shao, “Comparison of necessary conditions for typical Takagi-Sugeno and Mamdani fuzzy systems as universal approximators,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 5, pp. 508–514, Oct. 1999.

[30] D. Tikk, L. T. Kóczy, and T. D. Gedeon, “A survey on universal approximation and its limits in soft computing techniques,” *Int. J. Approx. Reason.*, vol. 33, no. 2, pp. 185–202, Jun. 2003.

[31] S. Guillaume, “Designing fuzzy inference systems from data: An interpretability-oriented review,” *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 3, pp. 426–443, Jun. 2001.

[32] R. R. Trippi and J. K. Lee, *Artificial Intelligence in Finance & Investing: State-of-the-Art Technologies for Securities Selection and Portfolio Management*. Chicago, IL: Irwin Professional, 1996.

[33] K. N. Pantazopoulos, L. H. Tsoukalas, N. G. Bourbakis, M. J. Brun, and E. N. Houstis, "Financial prediction and trading strategies using neurofuzzy approaches," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 28, no. 4, pp. 520–531, Aug. 1998.

[34] T. Nishina and M. Hagiwara, "Fuzzy inference neural network," *Neurocomput.*, vol. 14, no. 3, pp. 223–239, Feb. 1997.

[35] H. Nakanishi, I. B. Turksen, and M. Sugeno, "A review and comparison of six reasoning methods," *Fuzzy Sets Syst.*, vol. 57, no. 3, pp. 257–294, Aug. 1993.

[36] N. K. Kasabov and Q. Song, "DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 2, pp. 144–154, Apr. 2002.

[37] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *Int. J. Man-Machine Stud.*, vol. 7, no. 1, pp. 1–13, 1975.

[38] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985.

[39] J. Casillas, O. Cordón, F. Herrera, and L. Magdalena, *Interpretability Issues in Fuzzy Modeling*. Berlin, Germany: Springer-Verlag, 2003.

[40] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets Syst.*, vol. 28, no. 1, pp. 15–33, Oct. 1988.

[41] K. K. Ang and C. Quek, "RSPOP: Rough set-based pseudo outer-product fuzzy rule identification algorithm," *Neural Comput.*, vol. 17, no. 1, pp. 205–243, Jan. 2005.

[42] T. A. Johansen and R. Babuška, "Multiobjective identification of Takagi-Sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 847–860, Dec. 2003.

[43] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," *IEEE Trans. Fuzzy Syst.*, vol. 6, no. 4, pp. 530–537, Nov. 1998.

[44] Y. Jin, "Fuzzy modeling of high-dimensional systems: Complexity reduction," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 212–221, Apr. 2000.

[45] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Boston, MA: Kluwer, 1991.

[46] R. W. Swiniarski and A. Skowron, "Rough set methods in feature selection and recognition," *Pattern Recogn. Lett.*, vol. 24, no. 6, pp. 833–849, Mar. 2003.

[47] D. Kugiumtzis, B. Lillekjendlie, and N. Christophersen, "Chaotic time series part I: Estimation of some invariant properties in state space," *Model., Ident. Control*, vol. 15, no. 4, pp. 205–224, 1994.

[48] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Phys. Rev. Lett.*, vol. 45, no. 9, pp. 712–715, Sep. 1980.

[49] F. Takens, , D. A. Rand and L.- S. Young, Eds., "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence*. New York: Springer-Verlag, 1981, pp. 366–381.

[50] L. P. Maguire, B. Roche, T. M. McGinnity, and L. J. McDaid, "Predicting a chaotic time series using a fuzzy neural network," *Inform. Sci.*, vol. 112, no. 1–4, pp. 125–136, Dec. 1998.

[51] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks," in *Proc. IJCNN Int. Joint Conf. Neural Networks*, 1990, vol. 1, pp. 1–6.

[52] L.-X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 6, pp. 1414–1427, Nov. 1992.

[53] N. R. Sanders and L. P. Ritzman, "The need for contextual and technical knowledge in judgmental forecasting," *J. Behav. Decision Making*, vol. 5, no. 1, pp. 39–52, Jan. 1992.

[54] M. O'Connor, W. Remus, and K. Griggs, "Going up-going down: How good are people at forecasting trends and changes in trends?," *J. Forecasting*, vol. 16, no. 3, pp. 165–176, May 1997.

[55] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, , D. E. Rumelhart and J. L. McClelland, Eds., "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986, vol. 1, pp. 318–362.

[56] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis," *IEEE Trans. Neural Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 1991.

[57] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665–685, May 1993.

[58] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 267–278, 1994.

[59] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 6, pp. 902–918, Dec. 2001.

[60] K. K. Ang, C. Quek, and M. Pasquier, "POPFNN-CRI(S): Pseudo outer product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 838–849, Dec. 2003.

[61] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[62] D. Dubois and H. Prade, *Fundamentals of Fuzzy Sets*. Boston, MA: Kluwer, 2000.

[63] S. Taylor, Basics of moving averages 2001 [Online]. Available: http://www.investopedia.com/articles/technical/052201.asp

[64] Centre for computational intelligence 2005 [Online]. Available: http://www.c2i.ntu.edu.sg

[65] K. K. Ang and C. Quek, "Improved MCMAC with momentum, neighborhood, and averaged trapezoidal output," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 3, pp. 491–500, Jun. 2000.

[66] W. L. Tung and C. Quek, "GenSoFNN: A generic self-organizing fuzzy neural network," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1075–1086, Sep. 2002.

[67] R. W. Zhou and C. Quek, "POPFNN: A pseudo outer-product based fuzzy neural network," *Neural Networks*, vol. 9, no. 9, pp. 1569–1581, Dec. 1996.

Kai Keng Ang (S'05) received the B.A.Sc. (with first-class honors) and M.Phil. degrees in computer engineering from Nanyang Technological University, Singapore, in 1997 and 1999, respectively. He is currently working toward the Ph.D. degree at the Centre for Computational Intelligence, School of Computer Engineering, Nanyang Technological University.

He was a Senior Software Engineer with Delphi Automotive Systems, Singapore, Pte, Ltd., working on embedded software for automotive engine controllers from 1999 to 2003.

Mr. Ang was awarded the Singapore Millennium Foundation Ph.D. Scholarship in 2005.

Chai Quek (M'83) received the B.Sc. degree in electrical and electronics engineering and the Ph.D. degree in intelligent control, both from Heriot Watt University, Edinburgh, U.K., in 1986 and 1990, respectively.

He is an Associate Professor and a member of the Centre for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include intelligent control, intelligent architectures, artificial intelligence in education, neural networks, fuzzy systems, fuzzy rule-based systems, and genetic algorithms.