Applied Soft Computing 11 (2011) 3998–4010

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A dynamic threshold decision system for stock trading signal detection

Pei-Chann Chang[a,*], T. Warren Liao[b], Jyun-Jie Lin[a], Chin-Yuan Fan[c]

[a] Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan, ROC
[b] Department of Construction Management and Industrial Engineering, Louisiana State University, PFT Hall, Baton Rouge, LA 70803, USA
[c] Science & Technology Policy Research and information Center, National Applied Research Laboratories, Taipei, Taiwan, ROC

ARTICLE INFO

Article history:
Received 24 July 2010
Received in revised form 21 February 2011
Accepted 22 February 2011
Available online 8 March 2011

Keywords:
Dynamic threshold
Artificial Neural Network (ANN)
Piecewise Linear Representation method (PLR)
Stock turning points detection
Stock trading signals

ABSTRACT

Trading signal detection has become a very popular research topic in financial investment area. This paper develops a model using the Piecewise Linear Representations (PLR) and Artificial Neural Networks (ANNs) to analyze the nonlinear relationships between the stock closed price and various technical indexes, and uncovering the knowledge of trading signals hidden in historical data. Piecewise Linear Representation tools are applied to find the best stock turning points (trading signals) based on the historical data. These turning points represent short-term trading signals for selling or buying stocks from the market. This study further applies an Artificial Neural Network model to learn the connection weights from these historical turning points, and afterwards an exponential smoothing based dynamic threshold model is used to forecast the future trading signals. The stock trading signal is predicted using the neural network on a daily basis. The dynamic threshold bounds generated provide a guide for triggering a buy or sell decision when the ANN-predicted trading signal goes above or under the threshold bounds. Through a series of experiments, this research shows superior results than our previous research (Chang et al., 2009 [1]) and other benchmark researches.

1. Introduction

Stock trading signals have become a very popular research topic in financial engineering. Many researchers and investors have tried to determine the best time to buy or sell stocks, but inevitably make wrong decisions. The reason why trading signals are so difficult to detect or even observe is that price variations are subject to the problems of high dimensionality and non-stationary, because economic environments and political situations both affect stock price variations which make it even more difficult for researchers and investors to forecast trading signals.

Financial analysis often uses the fundamental analytical technique of trend analysis, which predicts or confirms trend reversals and trades at the peak or valley of the stock's price variation range. A financial time series consists of a sequence of local maximal and minimal points. These points contain useful information for financial analysis. Technical analysis is a major analytical technique based solely on price data. Evaluating financial time series data involves looking for peaks, valleys, trends, patterns, and other factors affecting a stock's price movement. Therefore, making buy/sell decisions based on these factors is a challenging task.

Stocks trading signals normally use technical indicators to monitor the stock prices and assist investors in setting up trading rules for buy–sell–hold decisions. Technical indicators are produced based on historical stock data. Historical stock data is in turn affected by the overall economic environment and political situations over time. This implies that technical indicators hide important information previously ignored. In previous studies, researchers have exerted significant effort in forecasting stock price variation and have proposed numerous sophisticated techniques to predict stock prices movements. Many researchers have used data mining and artificial intelligence tools to analyze technical indicators in an attempt to find the best trading signals. However, most past studies focus on the precise price prediction only, but in real world applications, determining the opportunity of trading points is more important than getting the price prediction. To overcome the shortcoming, this research develops a novel trading signals detection system to detect the buy/sell points more effectively and technical indicators are used as the input features for trading signals prediction.

The contributions of this research include two parts; firstly, we develop a model using Piecewise Linear Representations (PLR) and Artificial Neural Networks (ANNs) to analyze the nonlinear relationships between the stock closed price and various technical indexes, and to capture the knowledge of trading signals that are hidden in historical data. Then the learned ANN model is used to predict the future trading signals on a daily basis. However, the stock price varies in daily life, the trading signals' output from ANN is within the range of 0–1. It is still not clear to the user to define a good buy or sell decision. Therefore, secondly, a dynamic threshold

* Corresponding author. Tel.: +886 936101320; fax: +886 34638884.
E-mail address: iepchang@saturn.yzu.edu.tw (P.-C. Chang).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.02.029

decision system is further developed to trigger a trading decision. The dynamic threshold decision system is applied to generate a buy or sell decision when the predicted stock trading signal goes above or under the threshold. Therefore, the buy or sell decision can be predicted on a daily basis. Based on this strategy, the investors can buy or sell the selected stock and make a profit.

The remainder of this paper is divided into five sections. Section 2 reviews related literature in the area of stock trading signals. Section 3 describes the development of a hybrid system for supporting stock trading point decision making. Section 4 presents the experimental tests conducted in this study to determine the effectiveness of trading points generated using the proposed hybrid system. The final section provides conclusions and directions for future research.

2. Literature survey

The stock market is a complex system where decision-making can be very difficult; the market is affected by numerous factors and there is a high level of noisiness [2], stock trading signal detection is becoming the most popular research area in financial engineering. Many researchers have used different methods to predict stock trading signal. Recent studies in this area have increasingly used pattern recognition, data mining and computational intelligence. This section briefly introduces stock trading signal detection and also discusses the research methods used in stock trading signal prediction.

2.1. Traditional financial time series forecasting

Many researchers try to use formulas and statistical tools to find stock trading signals. The most popular methods include the regression model, the GARCH (generalized autoregressive conditional heteroskedasticity) model, the ARIMA (autoregressive moving average) model, and the probabilistic model. The regression model is a kind of simple linear forecasting model, where the output value y depends on the input signals $x_1 \ldots x_n$. However, this type of model does not seem to do very well in forecasting financial stock or exchange data. To address this weakness, Busuttil and Kalnishkan [3] developed two new regression models, the weight controlled kernel aggregating algorithm for regression (WeCKAAR) and the kernel aggregating algorithm for regression with changing dependencies (KAARCh). The WeCKAAR adds decaying weights to existing regression techniques, while the KAARCh adds an aggregating algorithm to trading signals. These methods can merge experts' opinion selected from a large pool of opinion rules to obtain optimal strategies. With the help of these two algorithms, their study showed excellent results in forecasting results of the Russian Trading System Stock Exchange (RTSSE).

The GARCH model is a very popular model used in traditional trading signal prediction. The GARCH model assigns exponentially decreasing weights to the original regression models. Diebold and Mariano [4] used this model to test exchange rate currencies. Their paper shows that the results of this model are similar to the traditional regression model and random walk strategy. Gencay [5] attempted to test linear and non-linear models using the GARCH model, but their results failed to generate significant trading signal data.

Another popular model which has been used in forecasting financial time series data is the Box–Jenkins ARIMA model. Unlike traditional regression models, these models combine stock data's moving average value and also find similar patterns. Timmermann and Granger [6] used this model to solve financial time series data, while Valenzuela et al. [7] and Pai and Lin [8] combined ARIMA with intelligent techniques, e.g. support vector machines for time series prediction. The later study uses the ARIMA model as a linear predictor to forecast stock price data, and then uses support vector machines to deal with the residual parts obtained from ARIMA model. Their research produces forecasting patterns that are very similar to the actual results. Their approach seems more successful than other methods. However, ARIMA model still has the limitation for system reliability [9]. The researcher presents that SVM model with the intelligence approach, e.g. genetic algorithm, is a better predictor than traditional approaches in forecasting system reliability.

Although the models described above show very interesting results in financial time series forecasting, they do not produce significant results in trading signal information. Unlike time series forecasting, trade signal prediction must be very precise. A strong trading signal detection method must consider not only stock price variations, but also other information that can help investors "buy low, sell high". Previous research shows that traditional linear models like the regression model, the GARCH model, and the ARIMA model, do not perform well in this task.

The probabilistic model offers an alternative to solve this problem. Bao and Yang [10] proposed a new learning strategy in stock trading signal detection that using the probabilistic model. Their proposed approach first attempts to find stock data critical point, and then adopts Markovian network to find the best trading signal probability. They were able to produce excellent results for S&P500 index data, and according to the authors' description, successfully predicted 416 of 454 profitable stocks. These results are quite successful in financial trading signal area.

As this section describes, probabilistic models can help solve the trading signal problem. However, probabilistic models require complex mathematical formulas which are not that easy to be understood by the investors. Therefore, it seems there should be another way to solve this problem more efficiently.

2.2. Pattern recognition

Pattern recognition is another hot topic been investigated for predicting stock trading signals. Many researchers have used historical data patterns to find future trading signals. Recently, Leigh et al. [11–13] proposed a chart heuristic for stock trading signal forecasting. A chart heuristic is a special kind of pattern recognition method. This method is based on the recognition of graphical stocks' price patterns and stocks' transaction volume. Using a series weight calculation, this method can find the best timing for trading that will earn the most profit. Leigh et al. [14] and Wang and Chan [15] used this method to forecast prices variation for the NYSE, the DOW-JONES Industrial Average, and the S&P500 index, and all achieved very significant results.

A chart heuristic seems to be the most popular pattern recognition tool used in financial time series data forecasting and trading signal detection. However, this method only considers graphical patterns, and ignores important messages that appear in different indices. Technical indices offer another solution in financial trading signal detection. A technical index shows much more information than price and transaction volume. Technical indices are another kind of method frequently used in financial trading signal prediction.

2.3. Technical index analysis

Technical index analysis (TA) is the process of analyzing historical stock prices to determine the possible future trends of a stock price. TA explores internal market information and assumes that all the necessary factors are in the stock exchange information. TA methodology is based on historically formed regularities in the stock exchange, and assumes that the same result will repeat in the future. Technical index analysis (TA) is one of the most popular methods in use by stock traders [16]. Traditionally, TA proposes a set of investment rules for the investor. However, the use of individual trading rules is not very effective from the investor's point of view [17,18]. Since the application of numerous rules is complicated and it is hard to combine trading rules, the overall trading profit results seems not good enough.

Many researchers have tried to use the combination of technical analysis and transaction amount to determine the exact moment for stock trading. To determine these time points, analysts develop a group of technical rules (TR) based on technical indicators (TIs). The aim of each rule is to generate either a buy signal when a bull market is anticipated or a sell signal when a bear market is expected. The better rule is one earning higher profit. Every TI is a function of time-series values, whereas each TR is a function of TI values and other independent parameters. A TR function can produce buy, hold, or sell signals. For more detailed information on TA, refer to [19,20].

Technical analysis can help extract financial information from stock price pattern. However, this method must be combined with other methods to help investors make accurate decisions. Recently, many researchers have used computational intelligence tools in combination with technical analysis to better determine trading signals.

2.4. Computational intelligence

Computational intelligence methods are also popular in financial trading signal forecasting [21]. Artificial Neural Networks have been applied to this area more frequently than other methods. However Refs. [22–25] show that these models have limitations when subjected to the tremendous noise and complex dimensionality of stock price data. The quantity of data and input variables also interfere with each other. Therefore, the results may not be convincing.

To improve the strength of Artificial Neural Networks, researchers have applied other computational intelligence (CI) methods to predict stock trading signal. A stock trading method based on hidden Markov processes can be adopted to model the dynamics of stock price trends. In Ref. [26], volatility from price data was used to characterize the amplitude of price return fluctuations, and the authors assumed that volatility could be described by a hidden Markov process. Westerhoff [27] developed a novel stock market model based not only on past prices, but also on past trading volume. This model has the potential to replicate some important stylized facts of stock markets, especially bubbles and crashes, excess volatility, fat tails, uncorrelated price increments, and volatility clustering. Thawornwong et al. [28] developed an intelligent stock trading decision support system that could forecast buying and selling signals based on the predictions of short-term and long-term trends using rule-based neural networks. The neural network component is composed of two rule-based neural networks used to predict stock price trends. A hybrid neuro-genetic approach [29,30] was also developed for stock trading, in which a recurrent neural network (NN) with one hidden layer was used for the prediction model. Recurrent neural network's input features are generated from a number of technical indicators used by financial experts. The genetic algorithm (GA) optimizes the NN's weights under a 2-D encoding and crossover strategy. This model was tested with data from 36 companies in the NYSE and NASDAQ for a period of 13 years, from 1992 to 2004. This neuro-genetic hybrid model was shown to have notable improvement over the buy-and-hold strategy, and the context-based ensemble further improved the results.

Recently, evolutionary computations and Artificial Neural Networks (ANNs) have gained wide popularity, especially in problems whose solution spaces are so complex and large that it is impossible to use traditional optimization methods. The application of ANNs for stock market prediction has its advantages and disadvantages. For example, Ref. [31] showed that ANNs could be used successfully for short-term series prediction. The application of an ANN is especially effective when the link between the independent and dependent variables from stock price variation is nonlinear and very noisy, which is typical of a stock market. The main drawback of the ANN approach is that ANNs do not provide any insight into the underlying processes, and prevent us from obtaining a specific collection of rules. Therefore, decision-making that relies solely on ANN results is not advisable. On the other hand, the application of evolutionary algorithms eliminates these problems and makes it possible to create more complex rules out of simple ones at the expense of greater computational effort. This suggests that hybrid systems can help avoid the weaknesses of signaling method while integrating their individual strengths. The following sections briefly introduce and survey related techniques applied in this study.

2.4.1. Technical analysis in our system

As mentioned before, technical analysis (TA) is the process of analyzing historical stock prices to determine the possible future trends of a stock price. TA explores internal market information and assumes that all the necessary factors are in the stock exchange information as a result of price dynamics. The TA methodology is based on historically formed regularities in the stock exchange, and assumes that the same results which appear in stock price variation will repeat in the future.

Thus, technical analysts affirm that the combination of the past share transaction rate and the transaction amount allows the investigator to determine the exact moment when the shares are either overvalued or undervalued. To determine these time points, analysts form a group of technical rules (TR) based on technical indicators (TIs). The aim of each rule is to generate a buy signal when a bull market is anticipated or a sell signal when a bear market is expected. The better the rules, the higher the earnings from trading stocks are. Every TI is a function of time-series values, whereas each TR is a function of TI values and other independent parameters. A TR function can produce buy, hold, or sell signals. For more detailed information on TA, refer to [19].

2.4.2. Exponential smoothing, ES

Exponential smoothing is a technique that can be applied to time series data, either to produce smoothed data for presentation, or to make prediction. The time series data themselves are a sequence of observations. Unlike the simple moving average that weights the past observations equally; exponential smoothing assigns exponentially decreasing weights over time.

The trading signal sometimes has the same direction (up or down) with stock; this makes it unreasonable to choose the threshold as 0.5 in the training phase. For overcoming this situation and developing a dynamic threshold, this research applies the adaptive exponential smoothing method [32] to solve this problem. Exponential smoothing had been used in many applications [33], especially in forecasting model [34,35].

2.4.3. Piecewise Linear Representation, PLR

Our previous study [1] shows that stock trading signals can be observed from variations in technical indexes or pattern matching. However, these variations are not easy to observe for all investors, and each trading time point appears and disappears quickly. Therefore, this study attempts to develop an intelligent trading point prediction system that investors can use to make a good trading strategy. Some recent studies applied PLR to pattern matching [36,37]. This study takes a different approach using PLR to identify the trough and peak of historical data. Based on these trading points, an Evolving Neural Network model is built to forecast the

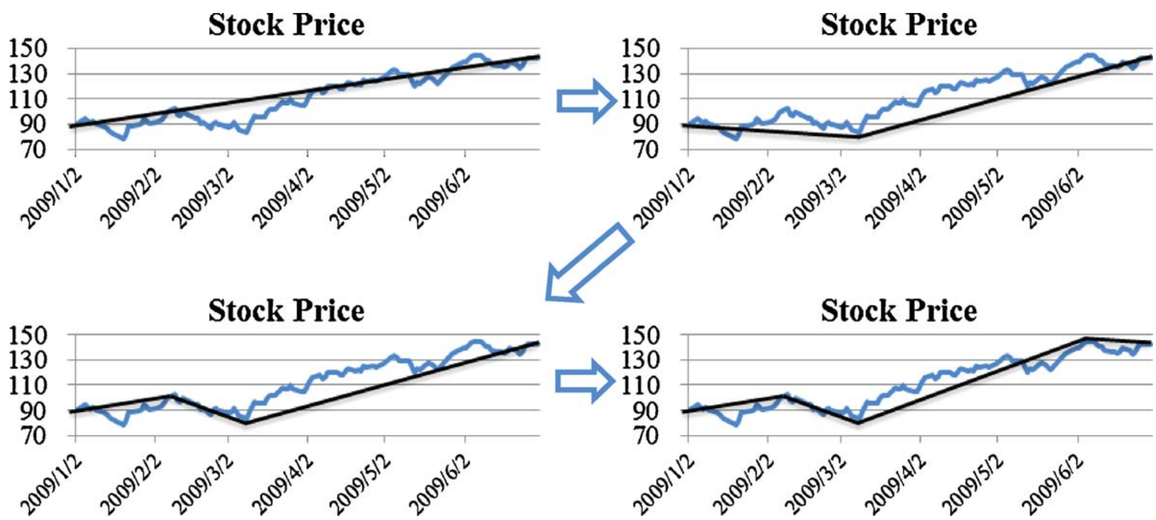

Fig. 1. Using PLR to generate possible trading points (Apple).

future trading point of a specific stock. The main procedures of PLR in predicting the trading point are described as follows:

PLR can be used as a sliding window to find the turning points of financial time series data. This method determines the size of the sliding window, and labels the relative maximum or minimum data points within the time window as the turning points. However, this may not be a very good approach, since the turning points depend upon the size of the sliding window. If the window size is not properly set, the sub-segments generated by PLR may lead to the wrong decision for future trading points. Therefore, this study selects historical data for piecewise representation. The turning points decided in this approach are therefore more representative of PLR and can be used to predict future trading points. Another key factor to be decided is the threshold value of a piecewise representation, which will be explained in greater detail in the following section.

2.4.3.1. Setup the segmentation threshold value (δ) of PLR. Larger segmentation threshold value will create long trend patterns; on the contrary, the patterns are very sensitive when the threshold value is very small. This research adopts genetic algorithms to fine-tune the threshold value (δ). Depending on the variation of each stock in the historical data, the threshold value is set up within the range of [0.01,5.0].

2.4.3.2. Trading point decision. PLR were developed for pattern matching, the advantage of this approach is to find the local extrema. The contribution of this study is that it uses PLR outputs to generate possible trading points. Take the Apple stock as an example. PLR will generate a segmentation diagram according to the threshold value applied. The graphs in Fig. 1 show the original time series and the segmentation result in steps. These segments are transformed later and regarded as trading signals to be used as input into the ANN for supervised learning. It is worthy to note that different threshold values for PLR will provide different segmentation results. A smaller threshold value represents a more detailed segmentation thus it is more sensitive with stock variation. While a larger threshold value represents a coarse segmentation that is a longer trend of the stock; therefore, this study uses different threshold values in PLR to study the variation of the proposed model and selects the one that makes the most profit within these values. In addition, Fig. 2 lists the pseudo codes of PLR used to generate trading points.

The quality of the trading points generated by PLR is determined by the threshold value (δ). In the traditional application, a trading signal is usually determined based on financial experts' opinions. The financial experts use technical analysis and economic situations to make a trading decision. If the trading signal is not properly determined, the investor can make a mistake and lose money. However, this study links these trading decisions to the financial expert of the threshold value (δ). In summary, this study applies GA to optimize the threshold value of PLR, with the expectation of generating better sub-segments and higher profits.

As discussed above, a considerable amount of research has been conducted to study the behavior of stock price movements. However, investor are more interested in making profit by following simple trading decision such as buy/hold/sell from the system rather than predicting the stock price itself. Therefore, this study develops an intelligent stock trading decision support system by providing trading signals to investors based on stock price variations.

2.4.4. Neural networks

Traditional Artificial Neural Networks (ANNs) simulate the biological neural networks in human brains. ANNs area kind of "learning algorithm"; that is, they can be trained to improve their performance by either supervised or unsupervised learning. A back-propagation network (BPN) is the most useful technique in the supervised learning of Artificial Neural Networks (ANNs), and therefore, this study chooses supervised learning, i.e., learned by samples. After learning (or training), the trained weight can be used to predict future occurrence. The BPN method has been widely applied to many scientific and commercial fields for nonlinear analysis and prediction. However, some researchers, like Yao [38] doubt the efficiency of traditional neural networks. The most important reason for this doubt is that the learning algorithm is strictly dependent on the shape of the error surface and the initial connection weights, so it means that Artificial Neural Network cannot converge to the global optimum. These limitations make Artificial Neural Networks inconsistent and unpredictable in different applications [39–42]. In this research, because the issues of neural networks described above, we design a transform function of trading signals and make it easier to be learned by neural networks. Furthermore, a dynamic threshold strategy is adopted to discover good trading points from the output of neural networks.

Prediction of a financial market is rather challenging due to chaos and uncertainty of the system. This study makes two

```
Input: T: time series, threshold
Output: P: best segments
Program IPLR (S: starting point, E: ending point)
    while not convergence
        Encode(threshold)
        Crossover()
        Mutation()
        profit = PLR(S, E, threshold)
        Elite strategy by ordering the profit
    End while
    return the best segments
End
```

```
Program PLR (S: starting point, E: ending point, threshold)
    Build the straight line L connect S and E
    For each point X in T
        Calculate Euclidean Distance(X, L)
    End For
    |x̂| = {x_i | max dist(x_i, L)}
    If |x̂| > threshold   //|x̂| is the distance of x̂ and L
        PLR(S, x̂)
        PLR(x̂, E)
    Else return L
    End if
    Calculate the profit by current segments
End
```

Fig. 2. The pseudo code of the PLR.

attempts to forecast the turning points of a stock movement: (1) A PLR approach is applied to generate possible historical turning points from historical stock price data for neural networks training and learning applications. (2) A modified neural network technique is developed to train the system and to predict the future turning points of a stock price movement. In addition, in order to adapt to the unexpected changes in the future trend of the stock, an exponential smoothing based dynamic threshold model is developed to produce the bounds to be compared with ANN-predicted trading signals for triggering a buy or sell decision.

3. Methodology

This research combines an Intelligent Piecewise Linear Representation and Artificial Neural Network techniques to detect financial time series data trading signals. Piecewise Linear Representation first selects turning points from the historical stock price database. Next, the PLR result is transformed to trading signals used in this research. Thirdly, the neural networks are adopted to "learn" the relationship between technical indices and trading signals. Finally, a dynamic threshold approach is applied to determine the buy–sell signals in test period. The flow chart of this model is shown in Fig. 3.

3.1. Stocks screening

A set of candidate stocks was selected based on the following criteria: (1) capital size; (2) monthly sales; (3) earnings per shares (EPS); (4) transaction volume per day; and (5) marginal accounts. Based on these factors, this study selected six famous stocks for testing, including three American stocks, Apple, Boeing Aerospace (BA) and Verizon Communications (VZ) and three Taiwan stock including AUO, EPISTAR and UMC.

3.2. Input variable selection

a. Determine the best segmentation threshold of PLR by GA
   An initial threshold is generated randomly for PLR in time series data segmentation and then evolved by genetic algorithm to find the best segmentation threshold value of PLR.

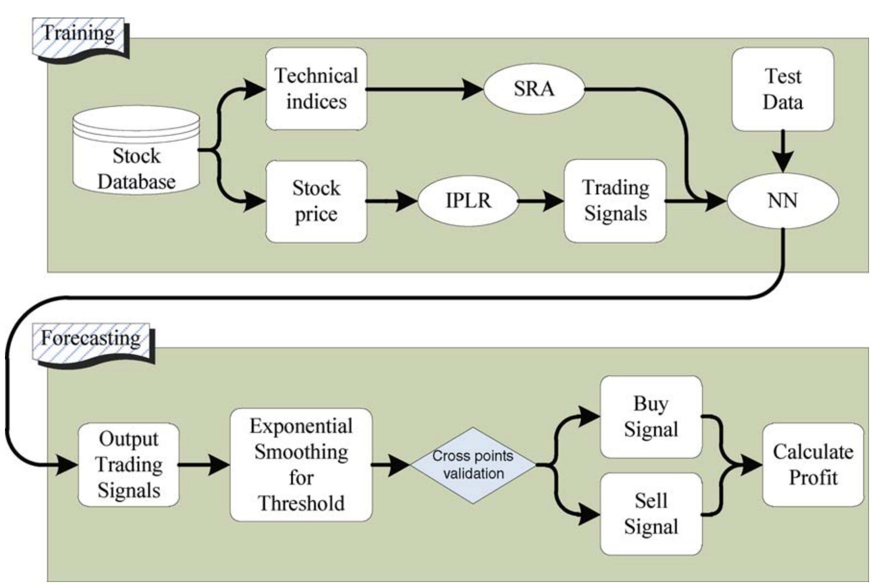

Fig. 3. The flow chart of the proposed methodology.

Fig. 4. The PLR segments of stock price (solid line: stock price, dark dot line: down-trend; dot line: up-trend).

b. Use PLR to segment the stock data

PLR is used to segment the selected stock data using the threshold generated in Step a. The stock data segmented will be transformed into trading signals. The segmentation algorithm uses a sliding window with a varying size. The sliding window contains at most *m* points, beginning after the last identified end point and ending right before the current point. If there are more than *m* points between the last end point and the current point, the sliding window contains only the last *m* points. The segmentation tries to find a possible upper or lower point in the current sliding window. Of course, different thresholds of PLR will produce different types of segmentations. As shown in Fig. 4, we use the segmentation threshold of two to divide the data. After the segmentation, the upper and lower extrema have to be converted into trading signals.

c. Transform PLR segments to trading signals

Traditionally, neural networks are adopted to classify positive and negative data, as a binary classifier. However, for predicting the turning points in time series, it is difficult to detect the higher or lower extrema since the stock price's movement varies up and down randomly. Fig. 4 shows the raw data of stock price's movement. In addition, the up trend and down trend of the stock price are shown using these dash lines. In this research, PLR is applied to segment the data of stock price and the temporary extrema are identified. Hence, the study adopts a transform function to transform the PLR segments to the trading signals and the pseudo code is provided in Fig. 5. Fig. 6 shows the transformed trading signals from Fig. 4 and this study uses these trading signals to train the neural networks. The basic assumption is that the technical indexes are closely related to the trading signals and the neural network is trained to capture this hidden knowledge among them. In Fig. 6, the *x*-axis, i.e., length, means the number of days of the stock in the up-trend or down-trend period and the *y*-axis is the trading signal which ranges from 0.0 to 1.0. And the decision of a "buy" or "sell" is based on the output of the neural network. When the output (trading signal) goes beyond the dynamic threshold bounds determined by exponential smoothing, the system will make a trading decision. Since the outputs from the neural network are expected to fluctuate within a small range. Thus, there are chances to make a false alarm, i.e., a bad "buy" or "sell" decision if a fixed threshold value is employed. To improve the performance, dynamic threshold bounds are needed to react to this dynamic situation. Therefore, a dynamic trigger threshold model is developed in this research which can dynamically react to the outputs and provide a more accurate "buy" or "sell" decision.

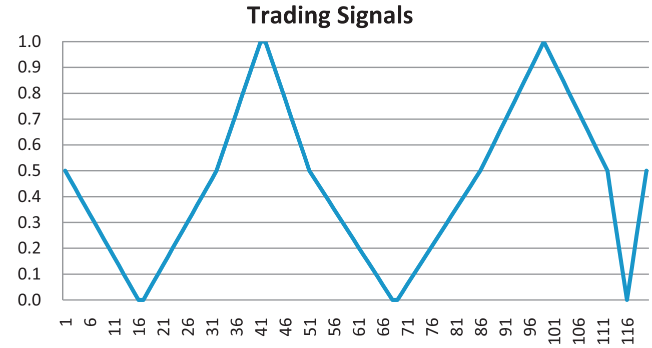

Fig. 6. The trading signals generated from the output of PLR.

d. Input variables selection using stepwise regression analysis (SRA)

Table 1 summarizes a set of technical indices affecting the stock price movement, as identified in previous studies by Chang et al. [1,43,44]. These input factors will be further selected using stepwise regression analysis (SRA). The stepwise regression analysis has been applied to determine the set of independent variables that most closely affect the dependent variable. This is accomplished by repeating the variable selection process. The following section explains the step-by-step procedure of the SRA approach:

Step 1. Calculate the correlation coefficient (r) of every input variable ($X_1, X_2, \ldots, X_n$), i.e., technical indices, and output data (Y), i.e., trading signal. Store all numbers in a correlation matrix.

Input : length //number of days in up-trend series	Input : length //number of days in down-trend series
Output : Ti , Array	**Output :** Ti
Procedure Up-Trend	**Procedure Down-Trend**
Ti[0] = 0.5	Ti[0] = 0.5
for i = 1 to length **do**	**for** i = 1 to length **do**
if (i/length) < 0.5 **then**	**if** (i/length) < 0.5 **then**
Ti[i] = Ti[i-1] - Δt ,// Δt = 1/length	Ti[i] = Ti[i-1] + Δt ,// Δt = 1/length
else	**else**
Ti[i] = Ti[i-1] + Δt	Ti[i] = Ti[i-1] - Δt
Endif	**Endif**
Endfor	**Endfor**
End Proc.	**End Proc.**

Fig. 5. The pseudo code for trading signals generation from the PLR segment.

Table 1
Technical indices used as input variables.

Technical index	Technical index (input in our system)	Explanation
Moving average (MA)	5MA, 6MA, 10MA, 20MA	Moving averages are used to emphasize the direction of a trend and smooth out price and volume fluctuations that can confuse interpretation
Bias (BIAS)	5BIAS, 10BIAS	The difference between the closing value and moving average line, which uses the stock price nature of returning back to average price to analyze the stock market
Relative strength index (RSI)	6RSI, 12RSI	RSI compares the magnitude of recent gains to recent losses in an attempt to determine overbought and oversold conditions of an asset
Nine days stochastic line (K, D)	KD	The stochastic line K and line D are used to determine the signals of over-purchasing, over-selling, or deviation
Moving average convergence and divergence (MACD)	9MACD	MACD shows the difference between a fast and slow exponential moving average (EMA) of closing prices. Fast means a short-period average, and slow means a long period one
Williams %R (pronounced "percent R")	12W%R	Williams %R is usually plotted using negative values. For the purpose of analysis and discussion, simply ignore the negative symbols. It is best to wait for the security's price to change direction before placing your trades
Transaction volume	Transaction volume	Transaction volume is a basic yet very important element of market timing strategy. Volume provides clues as to the intensity of a given price move
Differences of technical index (△)	△5MA, △6MA, △10MA, △5BIAS, △10BIAS, △6RSI, △12RSI, △12W%R, △9K, △9D, △9MACD	Differences of technical index between t day and $t+1$ day

Step 2. Choose the largest number of square (r^2) from the correlation matrix (suppose that X_i is the largest one in the current stage), and derive a regression model that is $\hat{Y} = f(X_i)$; then consider the correlation of Y with other input data. Assuming X_j has statistical significance; α value is applied to consider the significance of each input variable.
Step 3. Calculate the partial F value from other input data, as in Eq. (4), and choose the largest correlation coefficient among these input variables (assume it is X_j). Then, derive another regression model $\hat{Y} = f(X_i, X_j)$ again.

$$\mathrm{SSR} = \sum (\hat{Y}_i - \bar{Y})^2 \tag{2}$$

$$\mathrm{SSE} = \sum (\hat{Y}_i - Y_i)^2 \tag{3}$$

$$F_j^* = \frac{\mathrm{MSR}(X_j|X_i)}{\mathrm{MSE}(X_j, X_i)} = \frac{\mathrm{SSR}/1(X_j|X_i)}{\mathrm{SSE}/(n-2)(X_j, X_i)}, \quad i \in I \tag{4}$$

Step 4. Calculate the partial F value of the original data for input variable X_j. If the value is smaller than the critical value associated with a significance level, alpha, e.g. 0.1, remove it from the model since X_j is not statistically significant for the output.
Step 5. Repeat step 3 to step 4. If every input variable's partial F value is greater than the critical value, then stop. It means that every input value should have significant influences on the output value. According to Refs. [1,43,44], if F value of a specific variable is greater than the critical value, it is added to the model as a significant factor. When F value of a specific variable is smaller than the critical value, it is removed from the model. The statistical software SPSS for Windows 10.0 is applied for stepwise regression analysis in this research.

e. Normalize the selected variables from SRA
Normalize the selected variables from SRA as follows:

$$X_i{}' = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \tag{5}$$

f. Normalized variables input to Evolving Neural Network for trading point prediction
The set of normalized variables and the transformed trading signals (from c) are inputted into *Evolving Neural Network* for training the connection weights. Once the model is trained, it can be used to predict future trading points.

3.3. Optimization of segmentation threshold

During the optimization process, the segmentation threshold of PLR is a parameter to be optimized since the value can be from 0.1 to 10.0 or higher. Genetic algorithms (GA) are applied to optimize this threshold which is named as Intelligent PLR (IPLR). The fitness value of GA is the maximum profit among the training period. The IPLR is applied to find the optimal segmentation threshold for PLR. The profit derived from this threshold will be the best and then these trading points can be inputted into the neural network for training.

3.4. Artificial Neural Network, ANN

The parameters of ANN, especially the hidden layer's neural number, training rate, and initial random number seed, are always problem-dependent. In experimental results, detail parameter settings for the neural networks are provided and gradient method is applied to calculate the connection weights of the neurons.

In general, a neural network is defined as follows:

Assume that each input factor in the input layer is denoted by x_i, y_j and z_k represent the output in the hidden layer and the output layer, respectively. y_j and z_k can be expressed as follows:

$$y_j = f(X_j) = f\left(w_{0j} + \sum_{i=1}^{I} w_{ij} x_i\right) \tag{6}$$

and

$$z_k = f(Y_k) = f\left(w_{0k} + \sum_{j=1}^{J} w_{jk} y_j\right) \tag{7}$$

where w_{0j} and w_{0k} are the bias weights, f is the activation function used in both hidden and output layers, and X_j and Y_k are the temporarily computed results before applying the activation function. This study selects a sigmoid function (or logistic function) as the activation function. Therefore, the actual outputs y_j and z_k in the hidden and output layers, respectively, can also be written as:

$$y_j = f(X_j) = \frac{1}{1 + e^{-X_j}} \tag{8}$$

Table 2
An example of exponential smoothing.

Week	Real value	Predicted value with different α-values		
		0.1	0.2	0.3
8	102	85.00	85.00	85.00
9	110	86.70	88.40	90.10
10	90	89.03	95.72	96.07
11	105	89.13	92.18	94.25
12	95	90.72	94.74	97.48

and

$$z_k = f(Y_k) = \frac{1}{1+e^{-Y_k}} \tag{9}$$

The activation function f introduces the non-linear effect to the network, and maps the computational results to a domain [0,1]. The derivative of the sigmoid function in Eqs. (10) and (11) can be easily derived as:

$$f' = f(1-f) \tag{10}$$

The error function is defined as:

$$E = \frac{1}{2}\sum_{k=1}^{K} e_k^2 = \frac{1}{2}\sum_{k=1}^{K}(t_k - z_k)^2 \tag{11}$$

where t_k is a known network output (or desired output or target value) and e_k is the error in each output node. The goal is to minimize E so that the weight in each link is accordingly adjusted and the final output can match the desired output.

3.5. Dynamic threshold bounds

In our previous research [1], the triggering threshold was fixed and calculated by the test time period. Our previous research verified that trading signals indeed could be found in the model (neural networks) but the model could not be used in practical problems. Therefore, this study develops a dynamic triggering threshold approach and tries to decide the turning points in response to daily changing situations. In this research, the exponential smoothing (ES) method is adopted to calculate the threshold for triggering trading time points.

Exponential smoothing is one of the commonly used approaches in time series analysis and forecasting. The study adopts exponential smoothing to forecast the threshold for triggering trading time points in each day. As shown in Table 2, ES used the historical data and a smoothing factor, α, to compute the predicted value.

The general formula of exponential smoothing is as follow:

$$F_t = F_{t-1} + \alpha(A_{t-1} - F_{t-1}) = \alpha A_{t-1} + (1-\alpha)F_{t-1} \tag{12}$$

In Eq. (12), F_t is the predicted value at time t, while A_{t-1} and F_{t-1} represent the actual and the predicted value, respectively, at time $t-1$. α is the parameter to adjust the weights between real value and predicted value and make the forecasting more reliable.

Taking one of the data as an example, the predicted value of 10th week can be calculated as below:

$$F_{10} = F_9 + \alpha(A_9 - F_9)$$

$$\alpha = 0.1 \quad F_{10} = 86.7 + 0.1(110 - 86.7) = 89.03$$

$$\alpha = 0.2 \quad F_{10} = 88.4 + 0.2(110 - 88.4) = 92.72$$

$$\alpha = 0.3 \quad F_{10} = 90.1 + 0.3(110 - 90.1) = 96.07$$

In this research, the higher α value does not mean the higher prediction accuracy, the α-value should be determined by experiments.

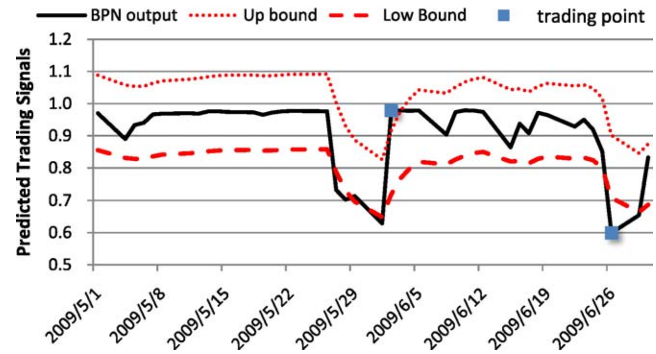

Fig. 7. The trading signals with upper–lower bound dynamic threshold.

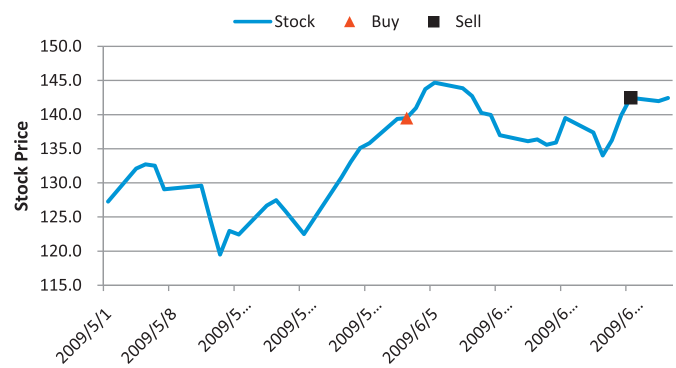

Fig. 8. The buy (triangle) and sell (square) points in stock price.

The idea of ES in this research is using historical data to dynamically predict the threshold. In general, we define the formula of α by Eq. (13).

$$\alpha = \frac{2}{\mathrm{AP}+1} \tag{13}$$

AP means Alpha's parameter, which represents the number of days that ES is used. In this study, the value of AP is five trading days (a week), therefore α-value is set as

$$\alpha = \frac{2}{5+1} = 0.33\overline{33}$$

As shown in Figs. 7 and 8, the results from ES alone can decide the trading points during the training or testing period. However, in consideration of the variation of neural networks' output, this study uses upper and lower bounds to avoid picking up excessive turning points and to derive more accurate trading points. In this research, the bounds are 10–30% of the original threshold predicted by ES, and the exact bounds are adjusted depending upon the variations of stock. In addition, during the test period, the first trading decision may be "buy" or "sell". The basic assumption in this research is that we have to "buy" then "sell" a stock. It is not allowed to "sell" first then "buy" a stock as an option. Therefore, if the first signal is "sell", it will be ignored and the system makes no decision at that time period. The formula for calculating upper bound and lower bound is shown as Eq. (14).

$$\begin{cases} \text{upper bound} = es_i \times (1+\text{bound}) \\ \text{lower bound} = es_i \times (1-\text{bound}) \end{cases}, \quad \text{for } es_i \text{ is } i\text{th exponential smoothing output} \tag{14}$$

the value of bound is from 10% to 30% of the dynamic threshold predicted by ES.

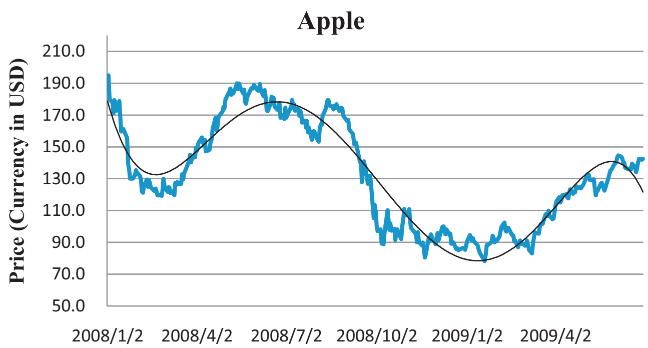

Fig. 9. Apple stock price and trend line.

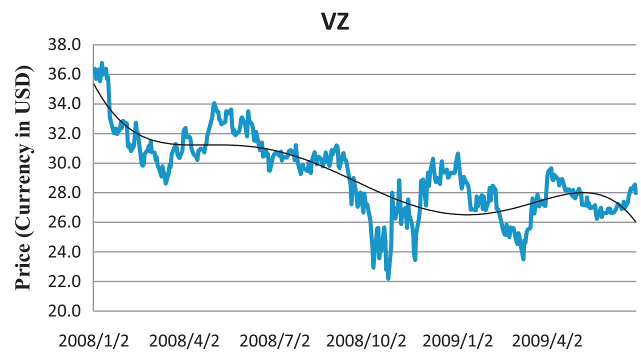

Fig. 11. Verizon Communications (VZ) stock price and trend line.

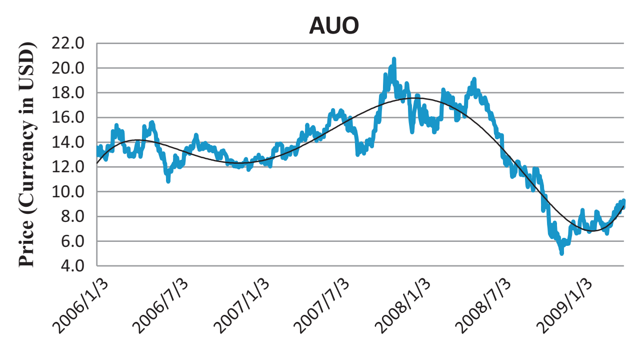

Fig. 12. AUO stock price and trend line.

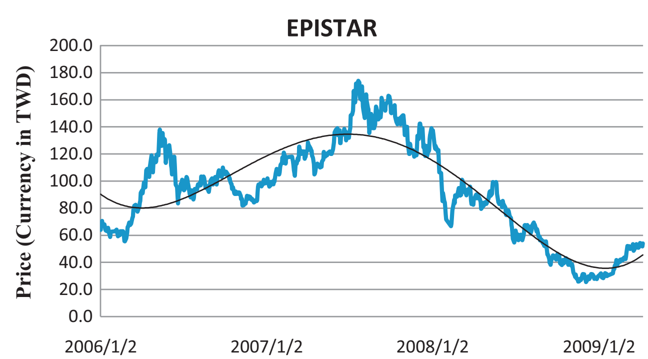

Fig. 13. EPISTAR stock price and trend line.

3.6. Estimate the earning

The earning of each stock traded within these testing periods is calculated based on an initial investment of US 100,000 dollars and typical transaction costs in the current stock exchange market. Finally, the net profit is computed.

4. Numerical examples

In this section, six different stocks are selected for testing the performance of the proposed dynamic threshold model. The six stocks include three stocks from the US and three stocks from Taiwan. In American stocks, the training data will be based on the data from 2008/01/02 to 2008/12/30 while the test data starts from 2009/1/2 to 2009/6/30. In those stocks one has an up-trend in the test period (Apple), one is flat (Boeing Aerospace) and another has a down-trend (Verizon Communications). For each Taiwan stock, the training period starts from 2006/1/2 to 2008/10/14 and the test period is from 2008/10/15 to 2009/4/9. The stocks include AU Optronics Corp. (AUO), EPISTAR and United Microelectronics Corporation (UMC). The length of America stocks is one year, which is shorter than the Taiwan stocks, because of stability and the transaction volume is large enough. Using a longer period to train a powerful supervised learning tool may cause the model to become unstable and "remember" some older information in the past period. On the other hand for Taiwan stocks, based on our experiments, three years data are necessary to properly train the model. For fair comparison, the test period is limited to half year in all stocks. In the following, these stocks with different fluctuation patterns are shown in Figs. 9–14. All datasets except for EPISTAR could be downloaded from Yahoo!Finance (http://finance.yahoo.com) and EPISTAR can be found in the Taiwan Economic Journal (http://www.tej.com.tw/twsite/). The experimental results show that regardless the trend of stock, our proposed model can determine the suitable trading points (buy/sell) and make a profit even when the stock price in the test period is quite different from those in the training period.

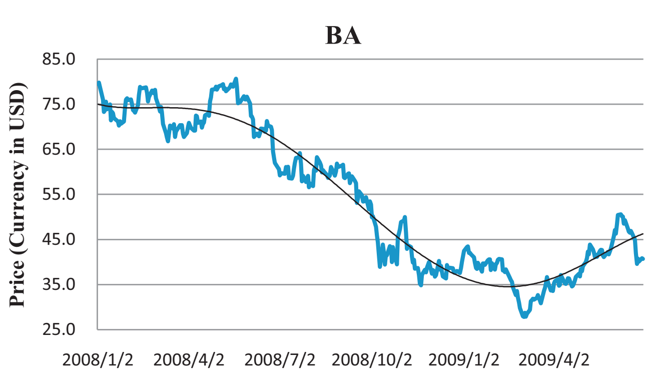

Fig. 10. Boeing Aerospace (BA) stock price and trend line.

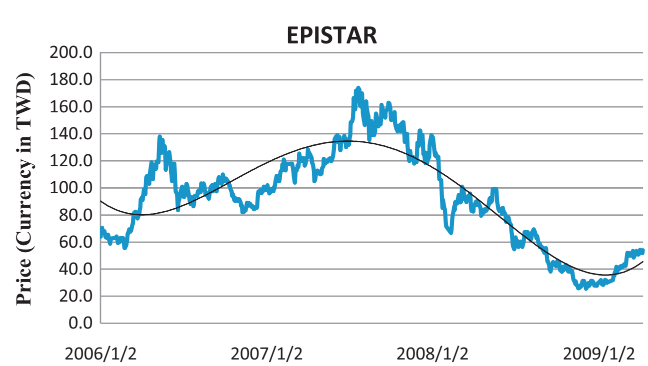

Fig. 14. UMC stock price and trend line.

Table 3
The parameter setup for ENN by DOE (Design of Experiments).

Parameter	Best
# of neuron in hidden layer	7
Transfer function	Sigmoid
Learning rule	Least Mean Square (LMS)
# of learning epoches	1000

Table 4
Input variables for different segmentation threshold values by SRA for PLR.

Initial threshold (δ)	Selected factors	Profits
2	5BIAS	53,432.1
3	9MACD, Δ5BIAS	55,653.1
4	Δ5BIAS, Δ9K, Δ9MACD	61,280.3
5	12W%R, Δ5BIAS, Δ6MA	56,276.2
6	5BIAS	53,720.7

4.1. Input variables and parameter settings of ANN

The parameter setting of ANN including the number of neurons in the hidden layer, transfer function, learning rule, etc., are determined based on a DOE (Design of Experiments). These parameters are very important since they will affect the system performance if they are not properly adjusted. The final set up of the parameters for ANN is listed in Table 3 after the DOE experimental tests.

For the Apple stock, the best segmentation threshold value is 4.0 with the maximum profit. The formula of profit calculation is listed in Eq. (15) and the input variables selected are Δ5BIAS, Δ9K, Δ9MACD, as shown in Table 4. These input variables are applied to the IPLR procedure and they will not be changed at all throughout the whole forecasting process.

$$\text{profit} = \prod_{i=1}^{n} \left\{ \left[\frac{(1-a-b)\cdot S_i - (1+a)\cdot B_i}{(1+a)\cdot B_i} \right] + 1 \right\} \cdot C \qquad (15)$$

where a is the tax, b is the transition fee and n means the number of buy/sell decisions. S_i is the sell price; B_i is the buy price while C is the invested capital.

4.2. The comparisons of different prediction models

After setting up the parameters of the experiments, we compared the proposed dynamic threshold based neural networks method with three existing methods. These three different algorithms to be compared are the Random walk strategy [45], Mallick et al. [46], and the PLR-BPN approach in our previous research [1]. Through a series of experimental tests, we can find that the proposed model consistently generates the highest profit among others. The overall comparisons of these four different models on these six different stocks in terms of rate of return are shown in Table 5. To compare with other methods, Table 5 shows the rate of return of each method instead of profit instead. The formulation of rate of return is listed in Eq. (16) and C is defined as in Eq. (15). For verifying the performance of this study visually, trading points generated by the proposed model for these six stocks are shown in Figs. 15–20. We also take the transaction cost into consideration, i.e., 40 US dollars per transaction.

$$\text{rate of return} = \left[\frac{\text{profit} - C}{C} \right] \times 100\% \qquad (16)$$

4.3. Discussions

In our previous research [1], a fixed training pattern was applied. Though significant profits were obtained, generally a fixed training pattern is not so convincing because historical data may or may not be able to reflect the current financial market tendencies. Especially, we are living in a highly dynamic and integrated international financial environment and any event suddenly occurs in any country will affect the financial market abruptly owing to the closely related global economic situation. Therefore, a dynamic threshold based strategy will improve our system, making it more accurate

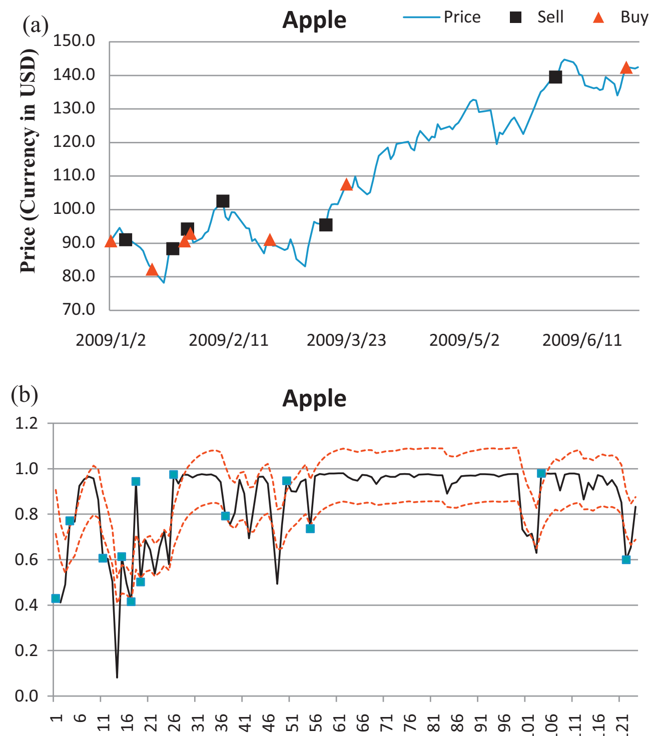

Fig. 15. (a) The forecast trading points of Apple (an up-trend stock) and (b) the dynamic threshold model that generate the decisions.

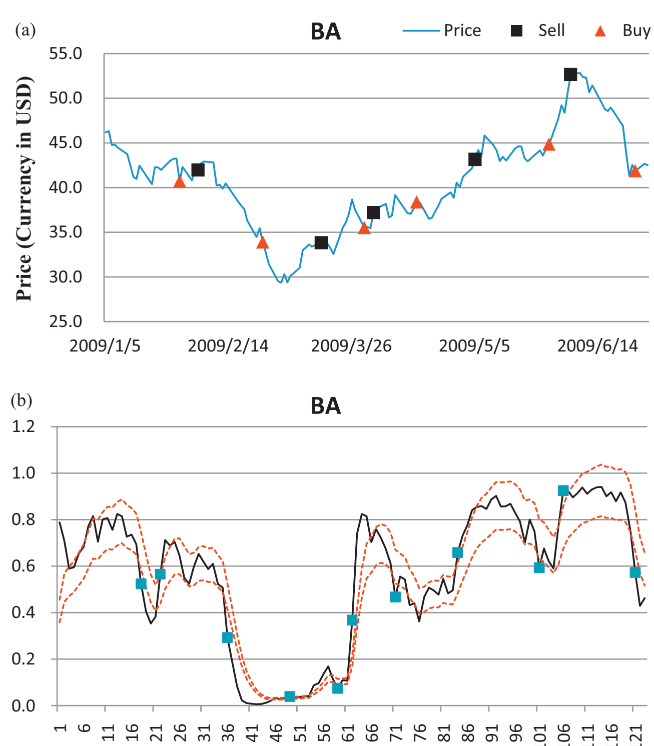

Fig. 16. (a) The forecast trading points of BA (a steady-trend stock) and (b) the dynamic threshold model that generates the decisions.

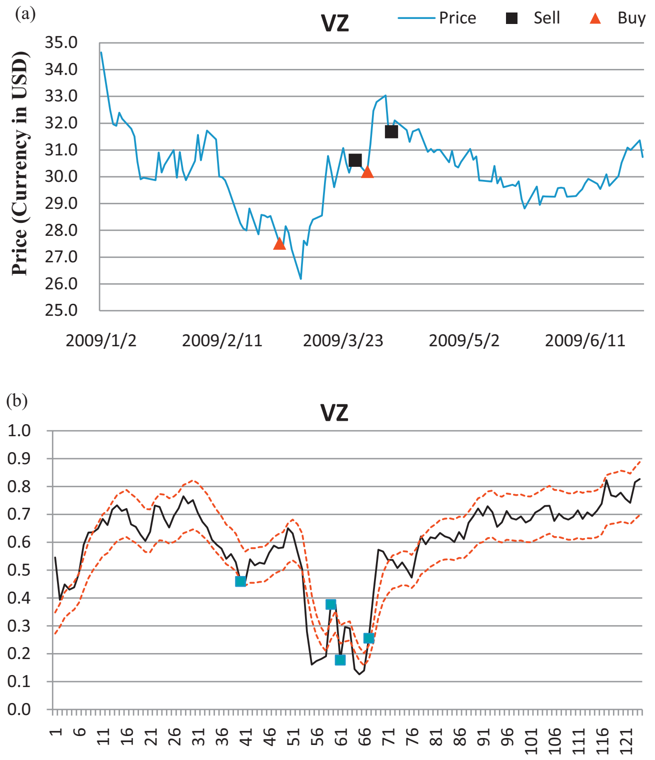

Fig. 17. (a) The forecast trading points of VZ (a down-trend stock) and (b) the dynamic threshold model that generates the decisions.

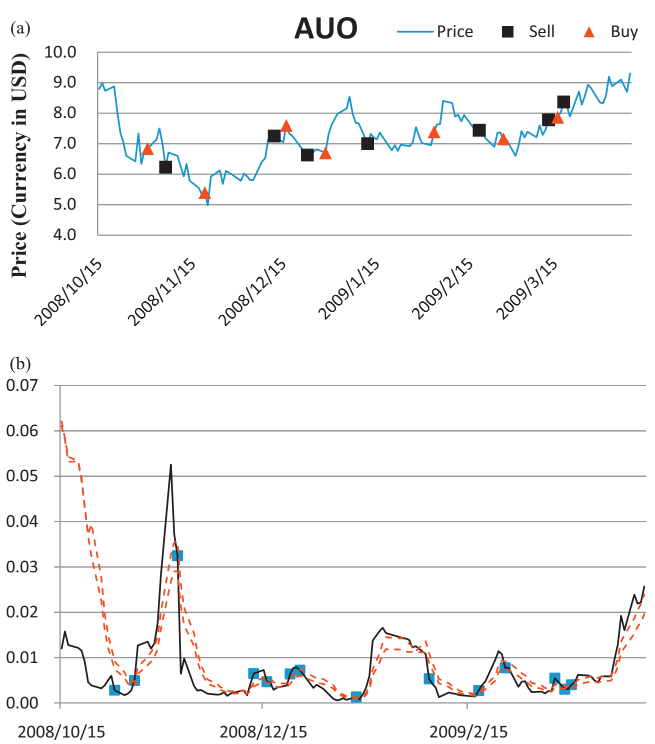

Fig. 18. (a) The forecast trading points of AUO and (b) the dynamic threshold model that generates the decisions.

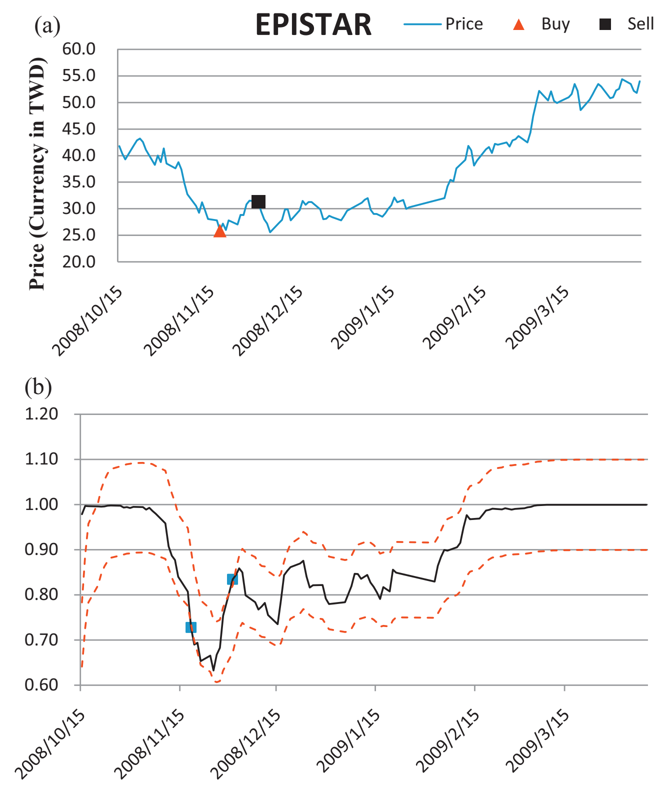

Fig. 19. (a) The forecast trading points of EPISTAR and (b) the dynamic threshold model that generates the decisions.

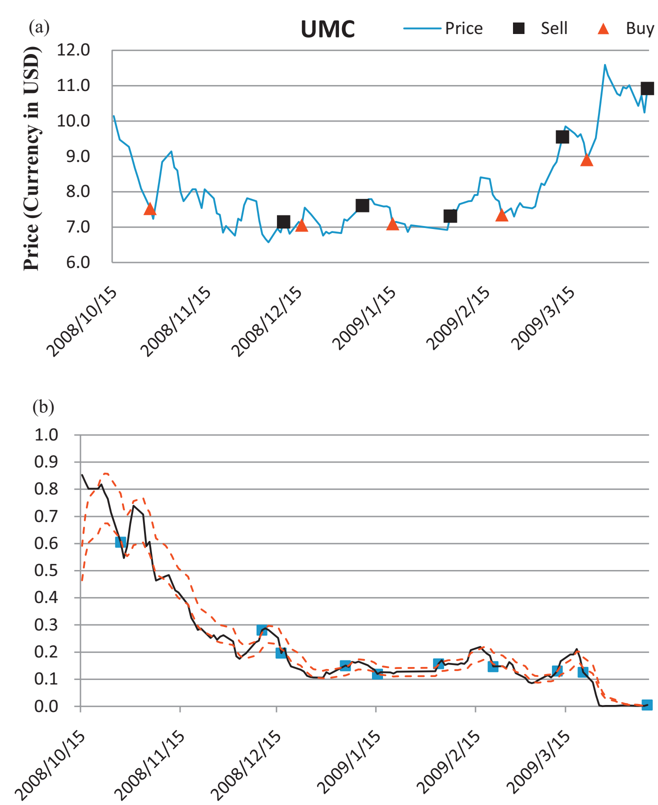

Fig. 20. (a) The forecast trading points of UMC and (b) the dynamic threshold model that generates the decisions.

Table 5
The overall comparisons of this research and others researches in rate of return.

Stocks	Random walk strategy [45]		Mallick et al. [46]		PLR [1]		This research	
	Rate of return (%)	Trading times	Rate of return (%)	Trading times	Rate of return (%)	Trading times	Rate of return (%)	Trading times
Apple	8.02	2	10.20	10	12.97	23	61.28	13
BA	−20.42	2	15.38	14	17.50	20	38.03	11
VZ	13.42	4	12.94	8	27.72	11	15.36	4
AUO	15.72	2	13.02	6	45.70	16	10.42	14
EPISTAR	9.67	4	15.44	12	35.10	12	20.34	2
UMC	3.37	2	5.41	6	3.87	6	63.13	10

and flexible for adapting to the volatile stock environment. That is why the proposed system shows better profit-making capability than the IPLR alone.

5. Conclusions

A considerable amount of research has been conducted to study the behavior of stock price movement. However, the investor is more interesting in making profit. Hence, a useful decision support system shall provide simple trading decisions such as buy/hold/sell other than predicting the stock price itself. Therefore, we take a different approach by applying a GA-tuned PLR to decompose the historical data into different segments. As a result, turning points (trough or peak) of the historical stock data can be detected and then serve as input into the neural network to train the connection weights of the model. Then, a new set of input data can be entered the model to make a prediction of the trading signal by the NN. A dynamic threshold method that is based on exponential smoothing is then applied to determine whether the trading signals outputted from the NN shall trigger a sell or buy decision or not with the objective to increase the profitability of the model. The experimental results show that the dynamic threshold based approach can make significant amount of profit consistently. In summary, the proposed system is very effective in predicting the future trading points of a specific stock. In the future, the proposed system can be further investigated by incorporating other soft computing techniques or using a better forecasting model other than NN. These future potential topics are listed as follows:

1. Clustering of financial time series data: data preprocessing is one of the features that can be applied in financial time series data processing. Effective clustering of time series data can further improve the forecasting accuracy of the forecasting system. But how these data to be clustered and what input factors are needed are interesting issues to be further investigated? The review by Liao [47] could be useful here.
2. Employing a different forecasting model: there are numerous forecasting models other than BPN model existing in the academic area. It is important to study the behavior of these models when applied in the prediction of the stock's trading points. Different input factors and different forecasting models such as SVM (support vector machine), FNN (fuzzy neural network) and CBR (case-based reasoning) are possible candidates for improving the accuracy of the proposed model.

References

[1] P.C. Chang, C.Y. Fan, C.H. Liu, Integrating a piecewise linear representation method and a neural network model for stock trading points prediction, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 39 (1) (2009) 80–92.
[2] Y.F. Wang, On-demand forecasting of stock prices using a real-time predictor, IEEE Transactions on Knowledge and Data Engineering 15 (4) (2003) 1033–1037.
[3] S. Busuttil, Y. Kalnishkan, Weighted kernel regression for predicting changing dependencies, Lecture Notes in Computer Science 4701 LNAI (2007) 535–542 (including sub series Lecture Notes in Artificial Intelligence and Lecture Notes in Bio informatics).
[4] F.X. Diebold, R.S. Mariano, Comparing predictive accuracy, Journal of Business and Economic Statistics 20 (1) (2002) 134–144.
[5] R. Gencay, Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules, Journal of International Economics 47 (1) (1999) 91–107.
[6] A Timmermann, C.W.J. Granger, Efficient market hypothesis and forecasting, International Journal of Forecasting 20 (1) (2004) 15–27.
[7] O. Valenzuela, I. Rojas, F. Rojas, H. Pomares, L.J. Herrera, A. Guillen, L. Marquez, M. Pasadas, Hybridization of intelligent techniques and ARIMA models for time series prediction, Fuzzy Sets and Systems 159 (7) (2008) 821–845.
[8] P.F. Pai, C.S. Lin, A hybrid ARIMA and support vector machines model in stock price forecasting, Omega 33 (6) (2005) 497–505.
[9] P.F. Pai, System reliability forecasting by support vector machines with genetic algorithms, Mathematical and Computer Modelling 43 (3–4) (2006) 262–274.
[10] D. Bao, Z. Yang, Intelligent stock trading system by turning point confirming and probabilistic reasoning, Expert Systems with Applications 34 (1) (2008) 620–627.
[11] W. Leigh, C.J. Frohlich, S. Hornik, R.L. Purvis, T.L. Roberts, Trading with a stock chart heuristic, IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans 38 (1) (2008) 93–104.
[12] W. Leigh, R. Hightower, N. Modani, Forecasting the New York stock exchange composite index with past price and interest rate on condition of volume spike, Expert Systems with Applications 28 (1) (2005) 1–8.
[13] W. Leigh, N. Modani, R. Purvis, T. Roberts, Stock market trading rule discovery using technical charting heuristics, Expert Systems with Applications 23 (2) (2002) 155–159.
[14] W. Leigh, R. Purvis, 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 Systems 32 (4) (2002) 361–377.
[15] J.L Wang, S.H. Chan, Trading rule discovery in the US stock market: an empirical study, Expert Systems with Applications 36 (3 PART 1) (2009) 5450–5455.
[16] G.S. Atsalakis, K.P. Valavanis, Surveying stock market forecasting techniques – part II: soft computing methods, Expert Systems with Applications 36 (3 PART 2) (2009) 5932–5941.
[17] M.A.H. Dempster, T.W. Payne, Y. Romahi, G.W.P. Thompson, Computational learning techniques for intraday FX trading using popular technical indicators, IEEE Transactions on Neural Networks 12 (4) (2001) 744–754.
[18] A.W. Lo, H. Mamaysky, J. Wang, Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation, Journal of Finance 55 (4) (2000) 1705–1765.
[19] S.B Achelis, Technical Analysis from A to Z, McGraw-Hill, 2000.
[20] K. Kurumatani, H. Kawamura, A. Ohuchi, Market micro-structure analysis by multiagent simulation in X-economy – comparison among technical indices, Information Sciences 170 (1) (2005) 65–74.
[21] Z. Zhu, H. He, J.A. Starzyk, C. Tseng, Self-organizing learning array and its application to economic and financial problems, Information Sciences 177 (5) (2007) 1180–1192.
[22] P.C. Chang, C.H. Liu, C.Y. Fan, Data clustering and fuzzy neural network for sales forecasting: a case study in printed circuit board industry, Knowledge-Based Systems 22 (5) (2009) 344–355.
[23] C.H. Cheng, T.L. Chen, L.Y. Wei, A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting, Information Sciences 180 (9) (2010) 1610–1629.
[24] L. Yu, H. Chen, S. Wang, K.K. Lai, Evolving least squares support vector machines for stock market trend mining, IEEE Transactions on Evolutionary Computation 13 (1) (2009) 87–102.
[25] L.J. Cao, F.E.H. Tay, Support vector machine with adaptive parameters in financial time series forecasting, IEEE Transactions on Neural Networks 14 (6) (2003) 1506–1518.
[26] Z. Eisler, J. Perelló, J. Masoliver, Volatility: a hidden Markov process in financial time series, Physical Review E – Statistical, Nonlinear, and Soft Matter Physics 76 (5) (2007).
[27] F.H. Westerhoff, Technical analysis based on price-volume signals and the power of trading breaks, International Journal of Theoretical and Applied Finance 9 (2) (2006) 227–244.

[28] S. Thawornwong, D. Enke, C. Dagli, Genetic algorithms and neural networks for stock trading prediction and technical signal optimization, in: Proceedings of the Annual Meeting of the Decision Sciences Institute, 2002.
[29] Y.K. Kwon, B.R. Moon, A hybrid neurogenetic approach for stock forecasting, IEEE Transactions on Neural Networks 18 (3) (2007) 851–864.
[30] G. Armano, M. Marchesi, A. Murru, A hybrid genetic-neural architecture for stock indexes forecasting, Information Sciences 170 (1) (2005) 3–33.
[31] F.D. Freitas, A.F. De Souza, A.R. de Almeida, Prediction-based portfolio optimization model using neural networks, Neurocomputing 72 (10–12) (2009) 2155–2170.
[32] D.W. Trigg, A.G. Leach, Exponential smoothing with an adaptive response rate, Operational Research Quarterly 18 (1) (1967) 53–59.
[33] E.S. Gardner Jr., Exponential smoothing: the state of the art – part II, International Journal of Forecasting 22 (4) (2006) 637–666.
[34] E.L. de Faria, M.P. Albuquerque, J.L. Gonzalez, J.T.P. Cavalcante, Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods, Expert Systems with Applications 36 (10) (2009) 12506–12509.
[35] B. Billah, M.L. King, R.D. Snyder, A.B. Koehler, Exponential smoothing model selection for forecasting, International Journal of Forecasting 22 (2) (2006) 239–247.
[36] A. Kimura, K. Kashino, T. Kurozumi, H. Murase, A quick search method for audio signals based on a piecewise linear representation of feature trajectories, IEEE Transactions on Audio, Speech and Language Processing 16 (2) (2008) 396–407.
[37] Y. Ding, X. Yang, A.J. Kavs, J. Li, A novel piecewise linear segmentation for time series, in: Proceedings of the 2010, the 2nd International Conference on Computer and Automation Engineering, 2010.
[38] X. Yao, Evolving artificial neural networks, Proceedings of the IEEE 87 (9) (1999) 1423–1447.
[39] Y. Ikuno, H. Kawabata, Y. Shirao, M. Hirata, T. Nagahara, Y. Iagaki, Application of an improved genetic algorithm to the learning of neural networks, Solid State Communications 91 (3) (1994) 731–735.
[40] G. Leng, T.M. McGinnity, G. Prasad, Design for self-organizing fuzzy neural networks based on genetic algorithms, IEEE Transactions on Fuzzy Systems 14 (6) (2006) 755–765.
[41] C.K. Goh, E.J. Teoh, K.C. Tan, Hybrid multiobjective evolutionary design for artificial neural networks, IEEE Transactions on Neural Networks 19 (9) (2008) 1531–1548.
[42] P.A. Castillo, J.J. Merelo, M.G. Arenas, G. Romero, Comparing evolutionary hybrid systems for design and optimization of multilayer perceptron structure along training parameters, Information Sciences 177 (14) (2007) 2884–2905.
[43] P.C. Chang, C.H. Liu, A TSK type fuzzy rule based system for stock price prediction, Expert Systems with Applications 34 (1) (2008) 135–144.
[44] P.C. Chang, C.Y. Fan, A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews 38 (6) (2008) 802–815.
[45] C.L. Giles, S. Lawrence, A.C. Tsoi, Noisy time series prediction using a recurrent neural network and grammatical inference, Machine Learning 44 (1–2) (2001) 161–183.
[46] D. Mallick, V.C.S. Lee, Y.S. Ong, An empirical study of genetic programming generated trading rules in computerized stock trading service system, in: 5th International Conference Service Systems and Service Management – Exploring Service Dynamics with Science and Innovative Technology, ICSSSM 2008, 2008, pp. 1–6.
[47] T.W. Liao, Clustering of time series data – a survey, Pattern Recognition 38 (2005) 1857–1874.