Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 615152, 20 pages
doi:10.1155/2012/615152

Research Article

A Trend-Based Segmentation Method and the Support Vector Regression for Financial Time Series Forecasting

Jheng-Long Wu and Pei-Chann Chang

Department of Information Management, Yuan Ze University, Taoyuan 32026, Taiwan

Correspondence should be addressed to Pei-Chann Chang, iepchang@saturn.yzu.edu.tw

Received 10 February 2012; Accepted 28 March 2012

Academic Editor: Ming Li

This paper presents a novel trend-based segmentation method (TBSM) and the support vector regression (SVR) for financial time series forecasting. The model is named as TBSM-SVR. Over the last decade, SVR has been a popular forecasting model for nonlinear time series problem. The general segmentation method, that is, the piecewise linear representation (PLR), has been applied to locate a set of trading points within a financial time series data. However, owing to the dynamics in stock trading, PLR cannot reflect the trend changes within a specific time period. Therefore, a trend based segmentation method is developed in this research to overcome this issue. The model is tested using various stocks from America stock market with different trend tendencies. The experimental results show that the proposed model can generate more profits than other models. The model is very practical for real-world application, and it can be implemented in a real-time environment.

1. Introduction

Support vector machines (SVMs) have outperformed other forecasting models of machine learning or soft computing (SC) tools such as decision tree, neural network (NN), bayes classifier, fuzzy systems (FSs), evolutionary computation (EC), and chaos theory by many researchers from historical nonlinear time series data applications in the last decade [1–5]. In these techniques, many researchers presented different forecasting models in dealing with characteristics such as imprecision, uncertainty, partial truth, and approximation to achieve practicability, robustness, and low solution cost in real applications [6–8]. However, the most important issue in resolving the nonlinear time series problem is error revision. ANNs use the empirical risk minimization principle to minimize the generalization errors but SVRs use the structural risk minimization principle because SVR is able to analyze with small

samples and to overcome the local optimal solution problem, which surpasses to ANNs [9–11]. Therefore, the SVRs forecasting model is applied to accomplish the forecasting task in this research. Presently, support vector regression (SVR), which was evolved from support vector machine (SVM) based on the statistical learning theory, is a powerful forecasting and machine learning approach for numerical prediction [12–15]. Also, SVR has high toleration error rate and high accuracy for learning solution knowledge in complex problems [16]. Although SVR can be applied well in time series data, the input vector is a key successful factor. Despite the volatile nature of the stock markets, researchers still can find certain correlations between these factors and stock prices. An investor's primary goal is to make profits. In order to help investors achieve their financial objectives, researchers have studied the relationship between financial markets and price variations over time from [17–20].

In the last few years, several representations of time series data have been proposed; the most often used representation is piecewise linear representation (PLR) [21–23]. It can decompose a time series data into a series of bottom and peak points [24, 25] in financial market. But the traditional PLR does not consider the multiple trending characteristics in time series. Moreover, the price movements of stocks are affected by many factors such as government policies, economic environments, interest rates, and inflation rates. The share prices of most listed companies also move up and down with other changing factors like market capitalization, earnings per share (EPS), price- to -earnings ratio, demand and supply, and market news. Moreover, there are more fractal properties of financial data, such as self-similarity, heavy-tailed distributions, long memory, as well as power laws [26–29]. One of fractal properties is long memory which is a common characteristic in financial data or other fields [30–32]. The daily stock trading is a short-term return so in this paper these fractal properties were not considered in our framework, just focusing on the real stock price's trends.

Therefore, there is a need to develop a new segmentation method which takes the price moving trends into consideration. As a result, this research will consider the multiple trends of stock price's movements in TBSM segmentation approach to capture the embedded knowledge of nonlinear time series. This research intends to improve the SVR forecasting performance using a trend based decomposition method. The TBSM approach has captured the tendency of stock price's movement which can be inputted into SVR in learning the historical knowledge of the time series data. Moreover, a more accurate forecasting result can be achieved when applied in real-time stock trading decision.

The rest of this paper is organized as follows. In Section 2, we describe TBSM segmentation principle. Forecasting model is discussed in Section 3. Section 4 explains modeling for trading decisions including using historical data to make trading decisions by the TBSM approach, selecting highly correlated technical indices by stepwise regression analysis (SRA), forecasting trading signals by SVR, and evaluating trading strategies. Section 5 explains how the TBSM with SVR for stock trading decisions and compares the profits obtained from various forecasting approaches. Finally, conclusions and directions for further research are discussed in Section 6.

2. A Trend Based Segmentation Method (TBSM)

In the time series database there are many approaches such as Fourier transform, wavelets, and piecewise linear representation which can be applied to find the turning point on time series data. According to the characteristics of sequential data, a piecewise linear representation of the data is more appropriate. A variety of algorithms to obtain a proper linear

```
Define: Threshold // cutting threshold
        X_Thld // horizontal area
        Y_Thld // vertical area
        X // a time series
        Y // stock price
1:  Procedure TBSM(T)
2:  Let T be represented as X[1, 2,..., n], Y[1, 2,..., n]
3:  n = 0
4:  Draw a line between (X1, Y1) and (Xn, Yn)
5:  Max d = maximum distance of (Xi, Yi) to the line
6:  If (Max d > Threshold)
7:     Let (Xi, Yi) be the point with maximum distance
8:     For j = X1 : Xn
9:        If (|Xj − Xi| < X_Thld) and (|Yj − Yi| < Y_Thld)
10:          Then Point[n] = [Xj, Yj], n = n + 1
11:       End If
12:    End For
13:    Select from Point[n ] : Xt1 = Min(X0), Xt2 = Max(Xn)
14:    Return:  S1 = T [X1, Xt1]
15:             S2 = T [Xt2, Xn]
16: End If
```

Algorithm 1: A pseudocode for TBSM in time series data.

representation of segment data have been presented. As reported in [33–36], PLR is used to support more tasks and provides an efficient and effective solution. In this paper we intend to enhance the segmentation accuracy based on different trends in stock price's movements. The basic idea of TBSM is to modify the PLR segmentation using the trend tendency in a specific time period. Three different trends such as uptrend, downtrend, and hold trend will be considered when making the segmentation. Detailed procedures of TBSM include the following. (1) PLR is applied to locate the turning points from the time series including up or downtrends. (2) The points around each turning point will be double-checked if the variations of the points are within the threshold. If yes, these points will have the same buy/sell trading in this period. (3) These points are set to be in the same trend. The pseudocode of the TBSM is shown in Algorithm 1.

For example, a time series $T = \{t_1, t_2, \ldots, t_{191}\}$ with 191 data is given to explain the basic idea of the TBSM procedure. As shown in Figure 1(a), several trading points are represented as buy (four red points) or sell (six green points) in this case. According to the TBSM procedure, we can draw a line S_1 form the first point to the last point as shown in Figure 1(b) and find the max distance to line S_1 which is point t_{26}. Then line S_1 is decomposed into two segments including line S_2 from t_1 to t_{26} and line S_3 from t_{26} to t_{191}. Based on point t_{26}, we can locate point t_{16} to t_{56} which are varied within the threshold. These points are set as hold trend and with the same state of point t_{26}. Therefore line S_2 and line S_3 will be changed to three different lines including line S_4 from point t_1 to point t_{16}, line S_5 from point t_{16} to point t_{56}, and line S_6 is from point t_{56} to point t_{191} as shown in Figure 1(c). Next step is repeating the same process for the rest of segments as t_{56} to t_{191}. The final results are shown in Figure 1(d) including two hold trend segments (dotted line), one uptrend segment, and two downtrend segments (solid line) in this time series.

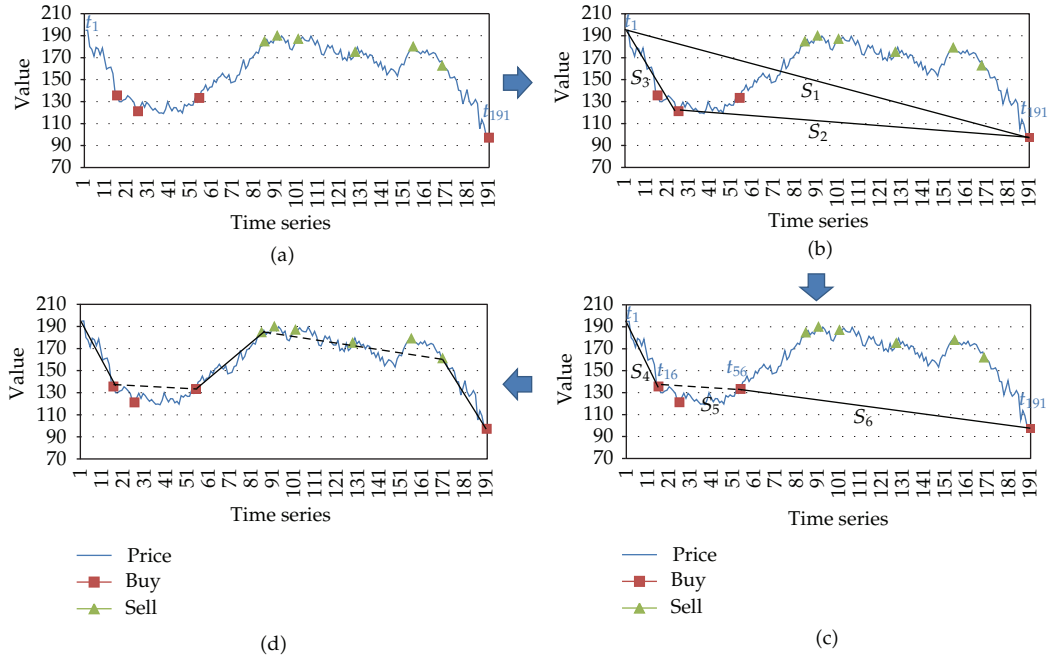

Figure 1: An example for TBSM in time series data.

3. Support Vector Regressions (SVRs)

Support vector regression is a modification of machine-learning-theory-based classification called support vector machine. Machine learning techniques have been applied for assigning trading signal. Many studies used support vector machine for determining whether a case contains particular class [37, 38]. But the shortcoming only deal with discrete class labels, whereas trading signal continuum data type because a weight of signal can take a buy or sell power. Grounded in statistical learning theory [1, 2], support vector regression is capable to predict the continuous trading signal while still benefiting from the robustness of SVM. SVM has been successfully employed to solve forecasting problems in many fields, such as financial time series forecasting [39] and emotion computation [40]. For explaining the concept of SVR, we have considered a standard regression problem. Let $S = \{X_i, Y_i\}_{i=1\ldots n}$ be the set of data where X_i is input vector (selected technical index in this research), Y_i (trading signal ts) is an output vector, and n is the number of data points. In regression analysis, we find a function $f(X_i)$ such that $Y_i = f(X_i)$. This function can be used to find the output value Y of any X. The standard regression function is as follows:

$$q_i = f(x_i) + \delta, \tag{3.1}$$

where δ denotes the random error and q_i denotes the estimated output. There are two types of regression problems, namely, linear and nonlinear. SVR is developed to tackle the nonlinear regression problems because the nonlinear regression problems have high complexity as well

as stock market trade. In SVR, at first the input vectors are nonlinearly mapped into a high-dimensional feature space (F), where they are linearly correlated with the respective output values.

SVR uses the following linear estimation function:

$$f(x) = (\omega \cdot \phi(x)) + b, \tag{3.2}$$

where ω denotes the weight vector, b denotes a constant, $\phi(x)$ denotes the mapping function in the feature space, and $(\omega \cdot \phi(x))$ denotes the dot product in the feature space F. SVR transfers the nonlinear regression problem of the lower dimension input space (x) into a linear regression problem of a high-dimension feature space. In other words, the optimization problem involving a nonlinear regression is converted into finding the flattest function in the feature space instead of input space.

Various cost functions like Laplacian, Huber's Gaussian, and ε-insensitive can be used in the formulation of SVR. The cost function should be suitable for the problem and should not be very complicated because a complicated cost function could lead to difficult optimization problems. Thus, we have used robust ε-sensitive cost function which is shown below:

$$L_\varepsilon(f(x), q) = \begin{cases} |f(x) - q| - \varepsilon, & \text{if } |f(x) - q| \geq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{3.3}$$

where ε denotes a precision parameter which represents the radius of the tube located around the regression function $f(x)$.

The $\{+\varepsilon, -\varepsilon\}$ region is called ε-insensitive zone. ε is determined by the user. If the actual output value lies in this region, the forecasting error is considered to be zero.

The weight vector, ω, and constant, b, in (3.2) are calculated by minimizing regularized risk function which is shown in (3.4):

$$R(C) = \frac{C}{n}\sum_{i=1}^{n} L_\varepsilon(f(x_i), q_i) + \frac{1}{2}|\omega|^2, \tag{3.4}$$

where $L_\varepsilon(f(x_i), q_i)$ denotes the ε-insensitive loss function, $|\omega|^2/2$ denotes the regularization term, and C denotes the regularization constant. ω decides the complexity and approximate accuracy of the regression model. Value of C is selected by the user to ensure appropriate value of w and low empirical risk.

The two positive slack variables ξ_i and ξ_i^* are used to replace the ε-insensitive loss function of (3.3). ξ_i is defined as the distance between the q_i and higher boundary of the ε-insensitive zone, and ξ_i^* is defined as the distance between the q_i and lower boundary of the ε-insensitive zone. Equation (3.4) is transformed into (3.5) by using the slack variables:

$$\text{Minimize} : R_{\text{reg}}(f) = \frac{1}{2}|\omega|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \tag{3.5}$$

$$\text{Subject to} \begin{cases} q_i - (\omega \cdot \phi(x_i)) - b \leq \varepsilon + \xi_i \\ (\omega \cdot \phi(x_i)) + b - q_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \quad \text{for } i = 1, \ldots, n. \end{cases} \tag{3.6}$$

Lagrange function method is used to find the solution which minimizes the regression risk of (3.4) with the cost function in (3.3) which results in the following quadratic programming problem (QP):

$$\begin{aligned}\text{Minimize}: \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i-\alpha_i^*)\left(\alpha_j-\alpha_j^*\right)(\phi(x_i)\cdot\phi(x_j))\\ +\sum_{i=1}^{N}\left(\varepsilon_i^{\text{up}}-y_i\right)\alpha_i+\sum_{i=1}^{N}\left(\varepsilon_i^{\text{down}}-y_i\right)\alpha_i^*,\end{aligned} \tag{3.7}$$

$$\text{Subject to}: \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0, \quad \text{where } \alpha_i,\alpha_i^*\in[0,C], \tag{3.8}$$

where α_i and α_i^* denote Lagrange multipliers. $\varepsilon_i^{\text{up}}$ and $\varepsilon_i^{\text{down}}$ represent the ith up- and downmargin, respectively. The value of $\varepsilon_i^{\text{up}}$ and $\varepsilon_i^{\text{down}}$ is equal to ε. The QP problem of (3.7) is solved under the constraints of (3.8). After solving the QP problem, we obtained Lagrange multiplier from (3.9), and (3.2) is transformed into the following equation (3.10):

$$\omega=\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)\cdot\phi(x_i), \tag{3.9}$$

$$f(x)=(\alpha_i-\alpha_i^*)(\phi(x_i)\cdot\phi(x))+b. \tag{3.10}$$

The Karush-Kuhn-Tucker (KKT) conditions are used to find the value of b. KKT conditions state that at the optimal solution, the product between the Lagrange multipliers and the constraints is equal to zero. The value of b can be calculated as follows:

$$b=\begin{cases}y_i-(\omega\cdot\phi(x_i))-\varepsilon_i^{\text{up}}, & \text{for } \alpha_i\in(0,C),\\ y_i-(\omega\cdot\phi(x_i))+\varepsilon_i^{\text{down}}, & \text{for } \alpha_i^*\in(0,C).\end{cases} \tag{3.11}$$

Using the trick of the kernel function, (3.10) can be written as (3.12):

$$f(x)=\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)K(x,x_i)+b, \tag{3.12}$$

where $K(x,x_i)=(\phi(x)\cdot\phi(x_i))$ denotes the kernel function which is symmetric and satisfies the Mercer's condition. SVR was able to predict the nonlinear relationship between technical indices and trading signal *ts* better than other soft computing (SC) techniques.

4. Application in Financial Time Series Data

This paper proposes a forecasting framework using a TBSM combined with SVR model which is called TBSM-SVR trading model for stock trading. The framework of TBSM-SVR trading model has five stages: the first is generating nonlinear trading segments by TBSM approach

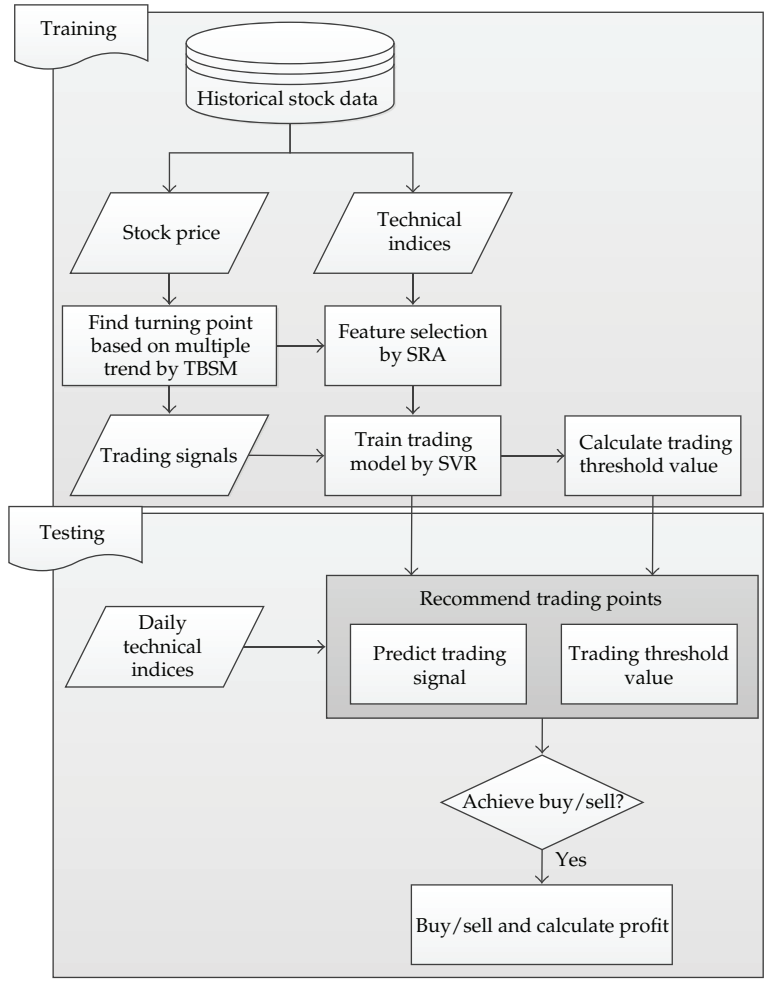

Figure 2: The framework of TBSM-SVR model for stock trading.

from historical stock price; the second is trading signal transformation from trading segments; the third is feature selection from technical indices by SRA approach; the fourth is learning the trading forecasting model by SVRs approach. The framework of TBSM-SVR model is shown in Figure 2. The five stages of TBSM-SVR model are explained as follows.

4.1. Find Turning Points Based on Multiple Trend by TBSM

According to TBSM procedure to find turning point based on trend of stock price, we selected a time series of historical stock price in a period to segment into several segments based on three trends including uptrend, downtrend, and hold trend. For example, a time series is given to segment trend segments from the date 2008/1/2 to 2008/12/30. Figure 3 shows the segmentation result by our proposed TBSM approach. The blue line is original historical stock price. The dashed lines are up/down trends which if the segment trend goes up is belonging to uptrend and if the segment trend goes down is belonging to downtrend. The dot line is

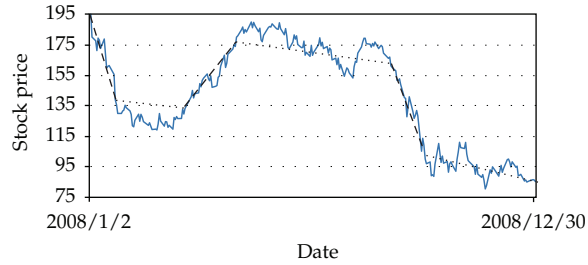

Figure 3: An example of segmentation result by TBSM.

belonging to hold trend. In our experiment, each stock price can split to multiple trend segments for trading signal transformation.

4.2. Trading Signal Transformation

In this stage, the aim is calculating the trading signal for a nonlinear time series of segmentation result which are a lot of segments based on trends. We suppose a segment S_k is uptrend; then we assume the real value into the vector S'_k like to $S_k = [0, 0.1, \ldots, 1]$; if S_k is hold trend but locates in buy point, then the vector like to $S'_k = [0.5, 0, 0.5]$; if S_k is hold trend but locates in sell point; then the vector like to $S'_k = [0.5, 1, 0.5]$; if S_k is downtrend, then the vector S'_k like to $[1, 0.9, \ldots, 0]$. Finally we combine these S'_k to a full time series of trading signal ts. If the segment belongs to uptrend or downtrend, then the formula equation (4.1) is used to calculate trading signal value:

$$S'_{k,i} = \begin{cases} \dfrac{i}{L} & \text{if } S_k \text{ is uptrend segment,} \\ \dfrac{(L-i)}{L} & \text{if } S_k \text{ is downtrend segment,} \end{cases} \tag{4.1}$$

where L denotes the length of segment S_k, whereas segment belonging to hold trend is using (4.2) to calculation:

$$S'_{k,i} = \begin{cases} 1 & \text{if } i\text{th is higherpoint in time series,} \\ 0 & \text{if } i\text{th is lower point in time series,} \\ 0.5 & \text{otherwise.} \end{cases} \tag{4.2}$$

For example, the S_1, and S_3 are hold trend; the S_1 is down-trend; the S_4 is up-trend. The result of trading signal ts is shown in Figure 4. The red dotted line is the hold trend which is a special signal for increasing reflects on the original turning points, so the hold trend is not a horizontal line. The purple dotted line is downtrend signal, and the orange dotted line is uptrend signal. For example, in the time series T the T_1 to T_5 and T_{10} to T_{14} are hold trend signal representation, T_6 to T_9 is downtrend signal representation, and finally T_{15} to T_{18} is uptrend signal representation. Finally the trading signal ts which is like to $ts = \{S_1, S_2, S_3, S_4\} = \{\langle 0.5, 0.5, 1, 0.5, 0.5\rangle, \langle 1, 0.66, 0.333, 0\rangle, \langle 0.5, 0.5, 0, 0.5, 0.5\rangle, \langle 0, 0.33, 0.66, 1\rangle\}$. For the detail process see the pseudocode in Algorithm 2.

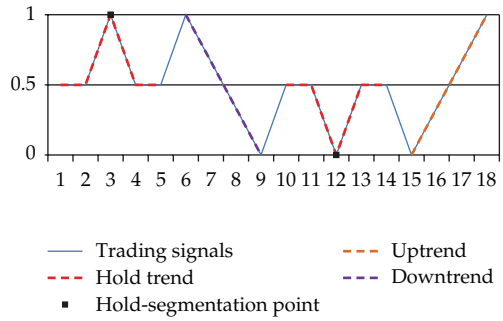

Figure 4: A sample of trading signal.

Input: *length*, *oldTs* // input data length and vector.
Output: *newTs* // a new time series vector of trading signal.
Method:

```
1:  Start = oldTs [1]
2:  End = oldTs[length]
3:  If Start = = −1 and End = = 1
4:        newTs [1] = 0
5:        For i = 1: length−1
6:              newTs[i+1] = 1/(lenghth−1)*i
7:        End For
8:  Else If Start = = 1 and End = = −1
9:        newTs[length] = 0
10:       For i = 1 : length−1
11:             newTs[i+1] = 1/(lenghth−1)*(length−i)
12:       End For
13:  Else
14:       For i = 2 : length−1
15:           newTs[i] = 0.5
16:       End For
17:  End If
```

Algorithm 2: A pseudocode for trend segments by TBSM in time series.

4.3. Feature Selection for Technical Indices by SRA

In this paper, we have considered 28 variables (technical indices) as listed in Table 1. These variables are correlated with variations in stock prices to some degree. The quantity of correlation varies for different variables. Rather than using all the 28 variables, we select the variables with a greater correlation than a user-defined threshold. The variable selection is done by stepwise regression analysis. We apply the SRA approach to determine which technical indices affecting the stock price. This is accomplished by selecting the variables repeatedly.

In the feature selection part input factors will be further selected using stepwise regression analysis (SRA). The SRA has been applied to determine the set of independent variables which is most closely affecting the dependent variable. The SRA is step by step to

Table 1: Technical indices used as input variables.

Technical	Technical index	Explanation
Moving average (Ma)	5 MA, 6 MA, 10 MA, 20 MA	Moving averages are used to emphasize the direction of a trend and smooth out price and volume fluctuations that can confuse interpretation.
Bias (BIAS)	5 BIAS, 10 BIAS	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)	6 RSI, 12 RSI	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)	9 K, 9 D	The stochastic line K and line D are used to determine the signals of overpurchasing, overselling, or deviation.
Moving average convergence and divergence (MACD)	9 MACD	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")	12 W%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.
Moving average convergence and divergence (MACD)	9 MACD	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")	12 W%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 (TV)	5 TV, 10 TV, 15 TV	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 (Δ)	Δ5 MA, Δ6 MA, Δ10 MA, Δ5 BIAS, Δ10 BIAS, Δ6 RSI, Δ12 RSI, Δ12 W%R, Δ9 K, Δ9 D, Δ9 MACD	Differences of technical index between the day and next day.

select factor into regression model which if factor has the significance level, then it is selected. We can follow (4.4) to calculate the F value of SRA:

$$\begin{aligned} \mathrm{SSR} &= \sum \left(\widehat{Y} - \overline{Y}\right)^2, \\ \mathrm{SSE} &= \sum \left(\widehat{Y}_i - Y_i\right)^2, \end{aligned} \tag{4.3}$$

$$F_j^* = \frac{\mathrm{MSR}(x_j \mid x_i)}{\mathrm{MSE}(x_j \mid x_i)} = \frac{\mathrm{SSR}(x_j \mid x_i)}{\mathrm{SSE}/(n-2)\ (x_j \mid x_i)} \quad i \in I, \tag{4.4}$$

where SSR denotes a regression sum of square. SSE denotes residual sum of squares. x is the value of technical index. y is the value of stock price. n is the total number of training data. $\widehat{Y}$ is the forecasting value of regression. $\overline{Y}$ is the average stock price of training data. After the feature selection by SRA, we can provide a set of features to form an input vector for the next step to learning the forecasting model.

The steps of the SRA approach are described as follows.

Step 1. Find the correlation coefficient $r$for each technical index $v_1, v_2, \ldots, v_n$ with the stock price y in a stock. These correlation coefficients are stored in a matrix called correlation matrix.

Step 2. The technical index with largest R^2 value is selected from the correlation matrix. Let the technical index be v_i. Derive a regression model between the stock price and technical index, that is, $\widehat{y} = f(v_i)$.

Step 3. Calculate the partial F value of other technical indices. Compare the R^2 value of the remaining technical indices and select the technical index with the highest correlation coefficient. Let the technical index be v_j. Derive another regression model, that is, $\widehat{y} = f(v_i, v_j)$.

Step 4. Calculate the partial F value of the original data for the technical index v_j. If the F-value is smaller than the user-defined threshold, v_j is removed from the regression model since it does not affect the stock price significantly.

Step 5. Repeat Step 3 to Step 4. If the F-value of variable is more than the user-defined threshold, the variable should be added to the model, otherwise it should be removed.

In addition, the range of the input variables of SVR model should be between 0 and 1. Hence, the selected technical indices are normalized as follows:

$$\mathrm{Normal}(x_{ij}) = \frac{x_{ij} - \mathrm{Min}(x_i)}{\mathrm{Max}(x_i) - \mathrm{Min}(x_i)} \quad i = 1, \ldots, n;\ j = 1, \ldots, m;\ n,\ \ m \in \Re, \tag{4.5}$$

where $\mathrm{Normal}(x_{ij})$ denotes the normalized value of jth data point of ith technical index. $\mathrm{Max}(x_i)$ denotes the maximum value of ith technical index. $\mathrm{Min}(x_i)$ denotes the minimum value of ith technical index. x_{ij} denotes original value of jth data point of ith technical index. n and m denote the total number of technical indices and data points, respectively.

4.4. Learning the Trading Forecasting Model by SVR

Support vector regression will be applied as a machine learning model to extract the hidden knowledge in the historic stock database. The single output is the trading signal ts from TSBM process, and the multiple input features are technical indices from SRA selection. SVR learning model transforms multiple features into high multidimensional feature space, and the transformed feature space can be mapped into a hyperplane space to determine correct signals based on those support vector points. On the kernel function selection, we try to use linear, RBF, polynomial, and sigmoid functions to generate better performance for the SVR model because the stock market is a very complicated nonlinear environment. Since the SVR approach possesses high learning capability and accuracy in predicting continuous signals for building hidden knowledge among trading signals and technical indices, it is a widely used tool for predicting the trading signals.

4.5. Trading Points Decision from Forecasted Trading Signal

In the daily forecasting, if the forecasted trading signals by SVR satisfied buy threshold, then this means it is needed to buy stock quickly because it is very close to turning point; otherwise if the state satisfied a sell threshold, then there is need to sell stock. These satisfied points are recommended to transaction in stock market. Before determining the trading point, we will calculate the buy/sell threshold values for two trading types. The trading thresholds of two types are as follows:

$$\begin{aligned} \text{Buy}_{\text{threshold}} &= \mu + \sigma, \\ \text{Sell}_{\text{threshold}} &= 1 - \mu + \sigma, \\ \mu &= \frac{1}{N}\sum_{i=1}^{N} x_i', \\ \sigma &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i' - \mu)}, \end{aligned} \tag{4.6}$$

where μ denotes the average of trading signal in training data. σ denotes the standard deviation of trading signal in training data. Buy_threshold denotes the buy trading threshold. Sell_threshold denotes the sell trading threshold. If forecasted trading signals form SVR model in testing data are more than buy_threshold, then this suggests trading point for buy stocks else if forecasting signal in testing data is smaller than sell_threshold, then this suggests trading for sell stock.

In the trading decision step, the TBSM-SVR model is employed to calculate daily trading signals. The detailed principles for making trading decisions include the following.

(1) If the time series prediction of trading signals by TBSM-SVR model is going up and intersects with buy trading threshold Buy_threshold, then it is a "buy" trading decision.

(2) If the time series prediction of trading signals by TBSM-SVR model is going down and intersects with sell trading threshold sell_threshold, then it is a "sell" trading decision.

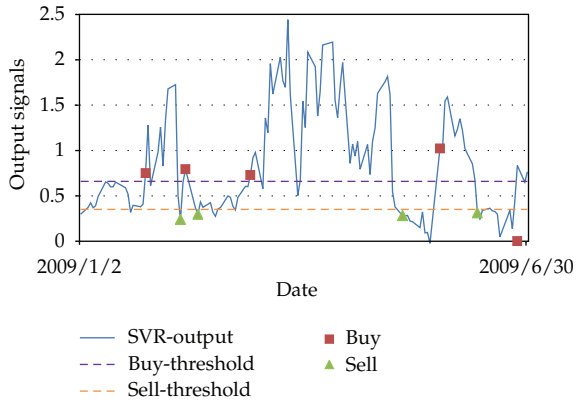

Figure 5: An example of result for detecting trading points of Apple.

(3) A "hold" trading decision is made (or do not make any trading decision) when the forecasting trading signal does not intersect with buy and sell thresholds.

For example, Figure 5 shows trading points decision for Apple stock. How to suggest the buy/sell points for stock in a time series in which the red square points are buy points and green triangle points are the sell points? Both are satisfied two thresholds in which the orange dotted line is sell threshold and the purple dotted line is buy threshold, so we can forecast the trading points daily by an automatically trading system.

5. Experimental Results

5.1. Profit Evaluation and Parameters Setting

In this research, the trading point (buy and sell timing) is decided by the TBSM-SVR model based on the forecasting trading signal of SVR and TBSM segmentation. In the experimental section, we also use various forecasting models to the generated profiting trading points and compare their performances. The profits in each different forecasting model are calculated as follows:

$$\text{profits} = C\prod_{i=1}^{k}\left\{\frac{(1-a-b)\times p_{S_i} - (1+a)\times p_{B_i}}{(1+a)\times p_{B_i}}\right\}, \tag{5.1}$$

where C is the total amount of money to be invested at the beginning as well as the capital of money, a refers to the tax rate of ith transaction, b refers to the handling charge of ith transaction, k is the total number of transaction, p_{S_i} is the selling price of the ith transaction and p_{B_i} is the buying price of ith transaction.

This study uses minimal root mean square error (RMSE) to measure the model performance in SVR train stage. In the model selection strategy that the dataset uses the last one trading period of training data contains (buy/sell and sell/buy states). The RMSE of an

Table 2: The parameter setup for TBSM and SVR by DOEs (design of experiments).

Approach	Parameter	Value	Explanation
TBSM	*Threshold*	0.1σ to 1σ	The difference of price at uptrend or downtrend
TBSM	*X_Thld*	0.1σ to 1σ	The difference of days at hold trend
TBSM	*Y_Thld*	0.1σ to 1σ	The difference of price at hold trend
SVR	C	10^{-3} to 10^{3}	Cost
SVR	ε	10^{-4} to 10^{-1}	Epsilon
SVR	d	2^{-9} to 2^{-1}	Degree
SVR	g	2^{1} to 2^{4}	Gamma

estimator $\widehat{ts}$ with respect to the estimated parameter ts is defined as the square root of the mean square error:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} ts_i - \widehat{ts_i}}{N}}. \tag{5.2}$$

ts denotes the trading signal by trading signal transformation from TBSM segmentation in Section 4.2. $\widehat{ts}$ denotes the estimated trading signal by SVR forecasting model. N denotes total number in each training data (Table 2).

In parameter section we use design of experiments (DOEs) approach to set each parameter for capture optimal parameter combination for trading system in financial data. The parameters of the TBSM are based on standard deviation σ from stock price in each stock which is the range from 0.1σ to 1σ for testing in each parameters. In SVR model, the kernels chosen for testing are "radial basis function (RBF)" and "polynomial" function. The common combination includes cost C; epsilon ε and γ are selected by the grid search with exponentially growing sequences. C ranges from 10^{-3} to 10^{3}. ε from 10^{-4} to 10^{-1} and γ is fixed as 0. In "polynomial" function, the degree d ranges from 2^{-9} to 2^{-1}. The gamma g ranges from 2^{1} to 2^{4} in RBF kernel.

5.2. Profit Comparison in the US Stock Market

In this research, we have selected 7 stocks from the US stock market to compare the profit achieved by various trading models, including Apple, BOENING CO. (BA), Caterpillar Inc. (CAT), Johnson and Johnson (JNJ), Exxon Mobil Corp. (XOM), Verizon Communication Inc. (VZ), and S&P 500. Among all the stocks, 253 data points were collected for the training period from 1/2/2008 (mm/dd/yy) to 12/31/2008 while 124 data points were used for the testing period from 1/2/2009 to 6/30/2009. In this research, we have compared our forecasting model of TBMS-SVR approach with two other identification models developed in the past. The PLR-BPN model proposed by Chang et al. [26] used neural networks in combination with PLR and exponential smoothing to determine the trading points. Kwon and Kish [41] used statistical model such as moving average, rate of change and trading volumes to determine the buy-sell points and generated profit.

Table 3: Feature selection result in each stock for technical indices by SRA.

Stock	Technical index
Apple	5 MA, 6 MA, 9 K, 9 MACD, 12 W%R
BA	5 MA, 6 MA, 9 K, 10 TV, 12 W%R
CAT	5 MA, 6 MA, 9 K, 10 TV, Δ5 MA
JNJ	5 MA, 6 MA, 6 RSI, 9 MACD, Δ5 MA
S&P 500	5 MA, 5 BIAS, 10 TV, 26 BR, TAPI
VZ	5 MA, 6 MA, Δ5 MA, 10 TV, 26 VR
XOM	5 MA,6 MA, Δ5 MA

Table 4: Model selection results from TSBM-SVR model for each stock.

Stock	Kernel									
	Radial basis function (RBF)					Polynomial				
	g	C	ε	SVs	RMSE	d	C	ε	SVs	RMSE
Apple	2^{-1}	10^3	10^{-4}	253	0.0819	2	[0.001 : 1000]	[0.0001 : 0.1]	71	0.266
BA	2^{-1}	10^3	10^{-1}	107	0.0955	2	[0.001 : 1000]	[0.0001 : 0.1]	76	0.269
CAT	2^{-1}	10^3	10^{-3}	254	0.0898	2	[0.001 : 1000]	[0.0001 : 0.1]	156	0.233
JNJ	2^{-1}	10^2	10^{-1}	137	0.2617	1	[0.001 : 1000]	[0.0001 : 0.1]	116	0.426
S&P 500	2^{-1}	10^3	10^{-4}	254	0.0004	1	[0.001 : 1000]	[0.0001 : 0.1]	112	0.379
VZ	2^{-1}	10^3	10^{-3}	251	0.0031	1	[0.001 : 1000]	[0.0001 : 0.1]	125	0.269
XOM	2^{-1}	10^3	10^{-4}	253	0.0001	2	[0.001 : 1000]	[0.0001 : 0.1]	182	0.18

Table 5: Comparison of profit obtained by various forecasting models.

Stock no.	Stock name	TBSN-SVR model (RBF)	PLR-SVR model (RBF)	PLR-BPN model	Statistical model
1	Apple	92.35%	35.84%	12.97%	20.50%
2	BA	59.49%	35.69%	17.50%	20.03%
3	CAT	43.39%	36.09%	9.36%	24.83%
4	JNJ	13.95%	9.47%	16.88%	0%
5	S&P 500	22.78%	4.19%	3.77%	9.81%
6	VZ	28.60%	2.60%	27.72%	0%
7	XOM	22.40%	12.34%	–1.99%	–7.65%
Average		40.42%	19.46%	12.32%	9.65%

The technical indices selected result by SRA as shown in Table 3. Apple, Ba, CAT, JNJ, S&P 500, and VZ used 5 features (technical indices) for training forecasting model; XOM used 3 features for training forecasting model. From this result we can know that a few features can capture more trading knowledge.

From model selection results the RBF kernel has better low error in each stock by RMSE. Moreover, the gamma, degree, cost, epsilon, support vectors, and RMSE as shown in Table 4 are necessary parameters and measures. The models of TBSM-SVR in each stock are selecting optimal parameter combination by RMSE consideration.

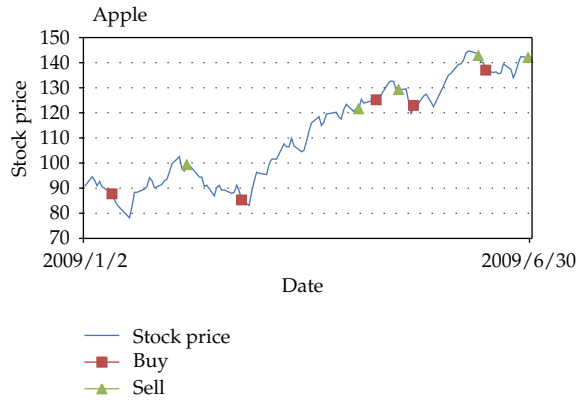

Figure 6: The forecasted trading points of Apple (an uptrend stock).

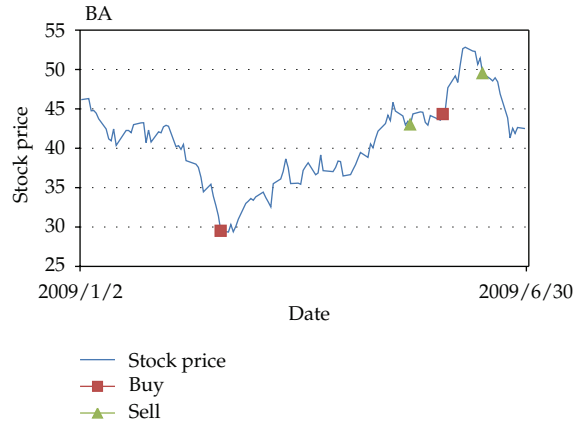

Figure 7: The forecasted trading points of BA (a steady-trend stock).

Each forecasting model provides trading points for each stock, so the best profits of the 3 forecasting models are shown in Table 5. The results turn out that our proposed TBSM with SVR model generates the greatest returns for the seven stocks, that is, number 1, 2, 3, 4, 5, 6, and 7 outperform other models. The average profit rate of these seven stocks is 40.42% using the TBSM-SVR model whereas the average profit rate generated by other models like PLR-SVR, PLR-BPN, and Statistical is 19.46%, 12.32%, and 9.65%, respectively. Therefore, our TBSM approach is better than PLR approach which is only considered linear representation.

The buy and sell points obtained from the TBSM forecasting model in each stock are shown in Figures 6, 7, 8, 9, 10, 11, and 12. The red square represents the buy point, and the black triangle represents the sell point using a trading strategy to determine turning points. Furthermore, our proposed approach TBSM is better than PLR segmentation which denotes that TBSM approach captures better trading knowledge for SVR forecasting model. Due to PLR only the linear representation is considering, so it loses important trend. Therefore, TBSM is an effective segmentation method for nonlinear time series data in stock market.

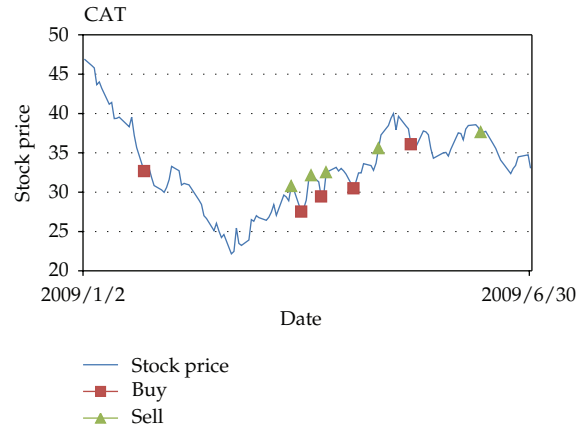

Figure 8: The forecasted trading points of CAT (a downtrend stock).

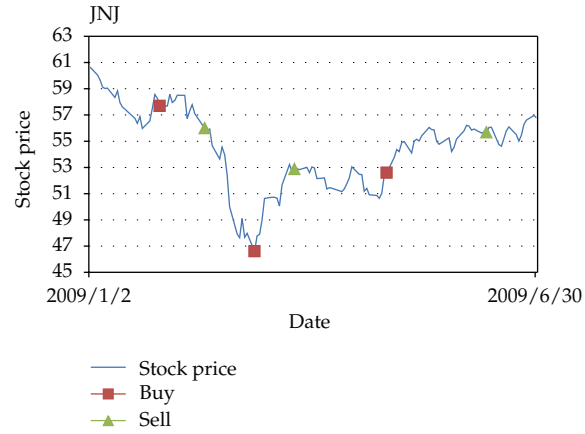

Figure 9: The forecasted trading points of JNJ (a steady-trend stock).

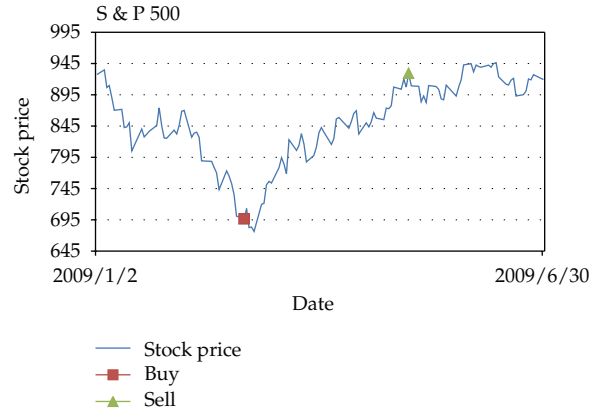

Figure 10: The forecasted trading points of S&P 500 (a steady-trend stock).

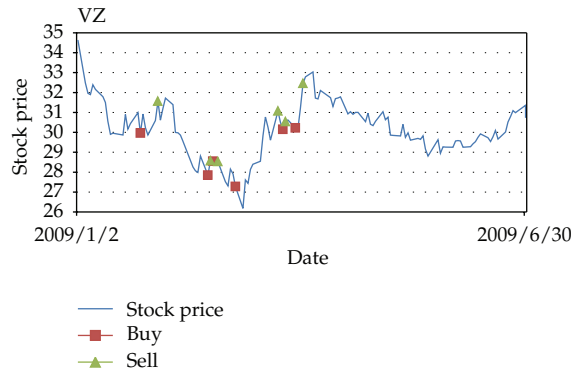

Figure 11: The forecasted trading points of VZ (a downtrend stock).

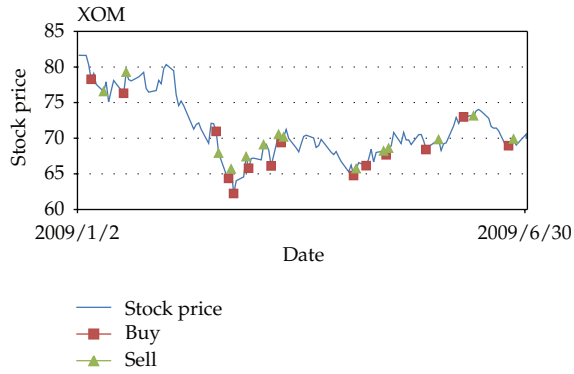

Figure 12: The forecasted trading points of XOM (a downtrend stock).

6. Conclusions

In this paper we proposed a trading system combining TBSM with SVR, and it is called TBSM-SVR-based stock trading system. This new trading system has been very effective in earning high profit while with the greatest ability. Experimental results showed that the TSBM can segment the stock price's variation into different trading trends. The trading signal in each trading trend will be assumed to be the same. The nonlinear time series can be better represented using these trading trends. Additionally, SVR is applied to capture the trading knowledge using the trading signals derived from these trading trends. The captured knowledge is more effective using TBSM-SVR when compared to PLR segmentation method. As a result, the primary goal of the investor could be easily achieved by providing him with simple trading decisions. However, the limitation of the TBSM-SVR trading system is the machine learning tool; that is, SVR is still not that mature yet. There are still rooms for the improvement of a better machine learning mechanism to be developed. Therefore, the trading system may make a wrong trading and lose money. In the future works, we can extend the segmentation method by considering a more detailed trend by investigating different buy-hold strategy or better trading strategy. In addition, the trend based segmentation method can further consider the fractal properties such as long memory, which can be accommodated to improve the segmentation performances.

References

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[2] V. N. Vapnik, *Statistical Learning Theory*, Adaptive and Learning Systems for Signal Processing, Communications, and Control, John Wiley & Sons, New York, NY, USA, 1998.

[3] Z. Liu, "Chaotic time series analysis," *Mathematical Problems in Engineering*, vol. 2010, Article ID 720190, 31 pages, 2010.

[4] B. J. Chen, M. W. Chang, and C. J. Lin, "Load forecasting using support vector machines: a study on EUNITE Competition 2001," *IEEE Transactions on Power Systems*, vol. 19, no. 4, pp. 1821–1830, 2004.

[5] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, pp. 219–269, 1995.

[6] X. H. Yang, D. X. She, Z. F. Yang, Q. H. Tang, and J. Q. Li, "Chaotic bayesian method based on multiple criteria decision making (MCDM) for forecasting nonlinear hydrological time series," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 10, no. 11-12, pp. 1595–1610, 2009.

[7] D. She and X. Yang, "A new adaptive local linear prediction method and its application in hydrological time series," *Mathematical Problems in Engineering*, vol. 2010, Article ID 205438, 15 pages, 2010.

[8] N. Muttil and K. W. Chau, "Neural network and genetic programming for modelling coastal algal blooms," *International Journal of Environment and Pollution*, vol. 28, no. 3-4, pp. 223–238, 2006.

[9] D. Niu, Y. Wang, and D. D. Wu, "Power load forecasting using support vector machine and ant colony optimization," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2531–2539, 2010.

[10] P. F. Pai and W. C. Hong, "Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms," *Electric Power Systems Research*, vol. 74, no. 3, pp. 417–425, 2005.

[11] W. C. Hong, "Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model," *Energy Conversion and Management*, vol. 50, no. 1, pp. 105–117, 2009.

[12] T. Farooq, A. Guergachi, and S. Krishnan, "Knowledge-based Green's Kernel for support vector regression," *Mathematical Problems in Engineering*, vol. 2010, Article ID 378652, 16 pages, 2010.

[13] S. O. Lozza, E. Angelelli, and A. Bianchi, "Financial applications of bivariate Markov processes," *Mathematical Problems in Engineering*, vol. 2011, Article ID 347604, 15 pages, 2011.

[14] A. Swishchuk and R. Manca, "Modeling and pricing of variance and volatility swaps for local semi-markov volatilities in financial engineering," *Mathematical Problems in Engineering*, vol. 2010, Article ID 537571, 17 pages, 2010.

[15] M. S. Abd-Elouahab, N. E. Hamri, and J. Wang, "Chaos control of a fractional-order financial system," *Mathematical Problems in Engineering*, vol. 2010, Article ID 270646, 18 pages, 2010.

[16] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[17] P. C. Chang and C. H. Liu, "A TSK type fuzzy rule based system for stock price prediction," *Expert Systems with Applications*, vol. 34, no. 1, pp. 135–144, 2008.

[18] P. F. Pai and C. S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, 2005.

[19] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," *Journal of Business and Economic Statistics*, vol. 20, no. 1, pp. 134–144, 2002.

[20] H. Liu and J. Wang, "Integrating independent component analysis and principal component analysis with neural network to predict Chinese stock market," *Mathematical Problems in Engineering*, vol. 2011, Article ID 382659, 15 pages, 2011.

[21] X. P. Ge, "Pattern matching in financial time series data," *Computer Communications*, vol. 27, pp. 935–945, 1998.

[22] E. Keogh and M. Pazzani, "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback," in *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD '98)*, pp. 239–241, August 1998.

[23] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan, "Mining of concurrent text and time series," in *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD '00)*, pp. 37–44, August 2000.

[24] S. Ghosh, P. Manimaran, and P. K. Panigrahi, "Characterizing multi-scale self-similar behavior and non-statistical properties of fluctuations in financial time series," *Physica A*, vol. 390, no. 23-24, pp. 4304–4316, 2011.

[25] P. C. Chang, C. Y. Tsai, C. H. Huang, and C. Y. Fan, "Application of a case base reasoning based support vector machine for financial time series data forecasting," in *Proceedings of the International Conference on Intelligent Computing (ICIC '09)*, vol. 5755, pp. 294–304, September 2009.

[26] P. C. Chang, C. Y. Fan, and 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*, vol. 39, no. 1, pp. 80–92, 2009.

[27] L. Todorova and B. Vogt, "Power law distribution in high frequency financial data? An econometric analysis," *Physica A*, vol. 390, no. 23-24, pp. 4433–4444, 2011.

[28] M. K. P. So, C. W. S. Chen, J. Y. Lee, and Y. P. Chang, "An empirical evaluation of fat-tailed distributions in modeling financial time series," *Mathematics and Computers in Simulation*, vol. 77, no. 1, pp. 96–108, 2008.

[29] M. Li and W. Zhao, "Visiting power laws in cyber-physical networking systems," *Mathematical Problems in Engineering*, vol. 2012, Article ID 302786, 13 pages, 2012.

[30] L. Muchnik, A. Bunde, and S. Havlin, "Long term memory in extreme returns of financial time series," *Physica A*, vol. 388, no. 19, pp. 4145–4150, 2009.

[31] M. Li, C. Cattani, and S. Y. Chen, "Viewing sea level by a one-dimensional random function with long memory," *Mathematical Problems in Engineering*, vol. 2011, Article ID 654284, 13 pages, 2011.

[32] M. Li, "Fractal time series—a tutorial review," *Mathematical Problems in Engineering*, vol. 2010, Article ID 157264, 26 pages, 2010.

[33] J. O. Lachaud, A. Vialard, and F. De Vieilleville, "Analysis and comparative evaluation of discrete tangent estimators," in *Proceedings of the 12th International Conference on Discrete Geometry for Computer Imagery (DGCI '05)*, E. Andres, G. Damiand, and P. Lienhardt, Eds., vol. 3429,, pp. 240–251, Springer, April 2005.

[34] Y. Zhu, D. Wu, and S. Li, "A piecewise linear representation method of time series based on feature pints," in *Proceedings of the11th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES '07), 17th Italian Workshop on Neural Networks (WIRN '07)*, pp. 1066–1072, January 2007.

[35] H. Wu, B. Salzberg, and D. Zhang, "Online event-driven subsequence matching over financial data streams," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '04)*, pp. 23–34, June 2004.

[36] Z. Zhang, J. Jiang, X. Liu et al., "Pattern recognition in stock data based on a new segmentation algorithm," in *Proceedings of the 2nd International Conference on Knowledge Science, Engineering and Management (KSEM '07)*, vol. 4798 of *Lecture Notes in Computer Science*, pp. 520–525, 2007.

[37] Y. W. Wang, P. C. Chang, C. Y. Fan, and C. H. Huang, "Database classification by integrating a case-based reasoning and support vector machine for induction," *Journal of Circuits, Systems and Computers*, vol. 19, no. 1, pp. 31–44, 2010.

[38] L. Zhang, W. D. Zhou, and P. C. Chang, "Generalized nonlinear discriminant analysis and its small sample size problems," *Neurocomputing*, vol. 74, no. 4, pp. 568–574, 2011.

[39] N. Sapankevych and R. Sankar, "Time series prediction using support vector machines: a survey," *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24–38, 2009.

[40] J. L. Wu, L. C. Yu, and P. C. Chang, "Emotion classification by removal of the overlap from incremental association language features," *Journal of the Chinese Institute of Engineers*, vol. 34, no. 7, pp. 947–955, 2011.

[41] K. Y. Kwon and R. J. Kish, "Technical trading strategies and return predictability: NYSE," *Applied Financial Economics*, vol. 12, no. 9, pp. 639–653, 2002.