International Journal of Innovative Computing, Information and Control
Volume **9**, Number **10**, October **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 3965–3980

A TURNING POINTS METHOD FOR STREAM TIME SERIES PREDICTION

Van Vo[1,2], Jiawei Luo[1,*] and Bay Vo[3]

[1]School of Information Science and Engineering
Hunan University
Yuelu District, Changsha 410082, P. R. China
[*]Corresponding author: luojiawei@hnu.edu.cn

[2]Faculty of Information Technology
Ho Chi Minh City University of Industry
Ho Chi Minh City, Vietnam
vothithanhvan@hui.edu.vn

[3]Information Technology College
Ho Chi Minh City, Vietnam
vdbay@itc.edu.vn

Received October 2012; revised February 2013

Abstract. *All of dimensionality reduction techniques are very meaningful to preprocess the large dataset and then use it to analyze and discover knowledge. In this paper, we propose an approach established on turning points to reduce the dimensions of stream time series data, and this task supports the prediction process faster in stream data environment. The turning points that are extracted from the maximum or minimum points of the time series data proved more efficient and effective in the process of preprocessing data for time series predictive analysis. To execute the proposed framework, we apply the stock time series obtained from Yahoo Finance, the predictive analysis based on a Sequential Minimal Optimization algorithm and the experimental results validate the effectiveness of our approach.*

Keywords: Stream mining, Time series dimensionality reduction, Turning points, Time series prediction

1. **Introduction.** In data mining studies, one of the significant tasks is the process of finding and extracting potential information to find knowledge hidden in the large dataset. Lately, time series analysis has related studies [4,10] which have received serious attention by many scientists for wide scientific and other applications. A time series is different from the traditionally achieved data as they have their own special characteristics, such as high dimensionality with large data size, and necessity to update continuously.

In the surroundings, the fundamental problem of time series data mining is how to represent the time series data [8,17,19]. The transforming the time series to another domain for dimensionality reduction [6,10,26,29] followed by an indexing mechanism is another approach.

Recent studies [17] make an analysis of the movement of a stock based on a selected number of points from the time series. These problems include Important Points (IPs), Perceptually Important Points (PIPs) and Turning Points (TPs). In a time series data, the important points are local minimum and maximum points. The perceptually important points are applied to the identification of frequently appearing technical (analysis) patterns. The PIPs usually contain a few noticeable points, and they are referred for technical pattern matching in many stock applications [1,2].

The main purpose of prediction methodologies is to use historical and other available data to predict a future event. Time series predictive analysis is one of interesting and challenging tasks in the field of data mining researches. In general, time series predictive analysis has two main approaches which are statistical and computational intelligence. With statistical approaches, it includes moving average, autoregressive moving average, autoregressive integrated moving average, linear regression and multiple regression models [10]. Yang et al. [9] give out the approach of support vector regression (SVR) [3] based on the Sequential Minimal Optimization (SMO) algorithm to create a model and predict future value of single time series by mining computation.

With the idea of dimensionality reduction, we are interested in the approach of reducing the number of data before applying the prediction techniques that are suitable for the stream data environment, which is in support of low prediction, training cost and high accuracy of the future values. In this paper, we propose a framework with the implementation of the TPs technique as preprocessing data apply to the SMO prediction algorithm. The experiment has three parts: first we remove the less important points, and then we evaluate the accuracy of future values depending on the number of histories used for prediction, and after that we make the performance comparison. To attest the effectiveness of the proposed framework, we use the stock price data set obtained from Yahoo Finance.

This paper is organized as follows. The dimensionality reduction and predictive analysis related works are listed in Section 2. We define the problem with several definitions and propose our framework in Section 3. Section 4 presents our research on time series dimensionality reduction by the turning points approach and time series prediction by SMO. Section 5 demonstrates the evaluation of our experiments. Finally, in Section 6, we conclude our work and propose our future works.

2. **Related Work.**

Time series analysis becomes an interesting and important research area due to its frequent appearance in many distinct applications [22,27,28]. In recent times, the increasing use of time series data has launched various researches in the field of data and knowledge management. Time series data are described as large, with high dimensionality and that needs continuous update. Moreover, the time series data are usually considered as a whole instead of individual numerical fields.

Mostly, there are many kinds of time series data related research, such as finding similar time series, subsequence searching in time series [2,20,21,29], dimensionality reduction and segmentation [12,17,26,29]. These analyses have been studied in considerable detail by both database and pattern recognition communities for different domains of time series data.

In the context of time series data mining, the fundamental problem is how to represent the time series data [1]. One of the common approaches is transforming the time series to another domain for dimensionality reduction followed by an indexing mechanism [6]. Moreover, the similarity measure between time series or time series subsequences and segmentation are two core tasks for various time series mining tasks [2]. Based on the time series representation, different mining tasks can be found in the literature, and they can be roughly classified into four areas: pattern discovery and clustering, classification, rule discovery and summarization. Some research concentrates on one of these areas, while the others may focus on more than one of the above processes.

Dimensionality reduction [17,26,29] is one of the most important preprocessing procedures for analyzing a stream time series environment. Dimensionality reduction is the process of reducing the number of variables or points under specific consideration. It can be divided into two main problems as an example of feature selection and feature extraction [18]. The technique is called feature selection implementation for selecting a subset of related features for building strong and useful learning models. The feature selection technique helps improving the performance of an analyzing model, such as moderating the effect of the course of dimensionality, enhancing generalization capability, speeding up the learning process, and improving model interpretability.

In many cases, the original representation [1,20,29] of the time series data might be redundant because of some reasons. For example, first, many of the variables will have a variation smaller than the measurement noise and thus will be irrelevant and second, many of the variables will be correlated with each other and thus a new set of in-correlated variables will be found.

There are some typical methods for time series dimensionality reduction in order to represent time series in lower dimensional spaces including the Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT) [14], Piecewise Linear Approximation [16], Piecewise Aggregate Approximation (PAA), Singular Value Decomposition (SVD) and Adaptive Piecewise Constant Approximation (APCA) [17]. Time series are highly correlated data, so that, the representation techniques use a scheme that aims at reducing the dimensionality of time series by projecting the original data onto lower dimensional spaces and processing the query in those reduced spaces. This scheme is widely used in time series data mining literature.

A sequence of data point is known as a time series and the major change of the data point has different extents of influence on the shape of the time series. That is why each data point of the time series has its own importance to the data stream. Some data points may contribute to the overall shape of the time series while others may only have little influence on the time series or they may even be discarded. These points are therefore, more important than other data points in the time series. Several approaches are based on important points such as Landmark points, Extreme points and Perceptually Important Points (PIP) [17,22,30].

One of the common statistical prediction approaches is the Autoregressive Integrated Moving Average (ARIMA) [27,28]. In general, the *ARIMA (p, d, q)* class consists of Autoregressive *AR (p)*, Moving Average *MA(q)*, and Autoregressive Moving Average *ARMA (p, q)* classes. Box and Jenkins proposed a general ARIMA model to cope with the modeling of non-stationary time series. An autoregressive model of order p views the present value of the series as the linear regression of the previous p values, whereas a moving average model of order q is conceptually a line regression of the current value of the series against previous white noise error terms. The *ARMA (p, q)* model is obtained by combining AR and MA, if a time series is not stationary, and this time series needs to be differentiated before applying *ARMA (p, q)*.

Related works on predicting values involve techniques that apply fuzzy rules and the problem of predicting unknown values is defined as follows: Assume we know H consecutive values, x_1, x_2, $\ldots$, and x_H, and want to predict the next F future values x_{H+1}, x_{H+2}, $\ldots$, and x_{H+F}. The traditional fuzzy predictor predicts values directly on the raw time series, where the fuzzy rules are in the form "*if* $x_1 = value_1$, $x_2 = value_2$, $\ldots$, $x_H = value_H$, *then* $x_k = value_k$, $k \in [H+1, H+F]$". However, this approach can guarantee the prediction accuracy only if the time series statistics are stationary. Otherwise, with a dynamic series, it might fail to give accurate results. Recently, Wong et al. [25] suggested an adaptive time variant prediction model based on window size of fuzzy time series. Joshi and Kumar [16] presented a fuzzy time series model based on the non-determinacy index by incorporating intuitionistic fuzzy sets.

For data mining techniques, Men and Liu apply the Least Squares Support Vector Machines (LS-SVMs) [19] method based on time series to actual load forecasting. This methodology LS-SVMs algorithm based on time sequence can assure higher accuracy and faster convergence speed as compared with the other traditional time series method and it also can discover the global optimal solution. Guraksin and Uguz [24] use LS-SVMs in a biomedical system in order to search the classification of normal, mitral stenosis and pulmonary stenosis heart sounds.

Xian and Lei [26,27] provide DFT and probability predictions. The DFT takes a discrete time series of n equally spaced samples and transforms or converts this time series through a mathematical operation into a set of n complex numbers defined in what is called the frequency domain.

3. **Problem Statement and Framework.** A time series is a set of observations x_t, each one being recorded at a specific time t. A discrete time series is one where the set of times at which observations are made is a discrete set. Continuous time series are obtained by recording observations continuously over some time interval.

In another way, a time series as a sequence of random variables x_1, x_2, x_3, $\ldots$, x_N, where the random variable x_1 marks the value taken by the series at the first time point, the variable x_2 marks the value for the second time period, x_3 marks the value for the third time period, and so on.

When the total number of data points in the time series is known in advance, this time series is static, and that time series has a length N. When data points are arrived at continuously, the value of N represents the number of data points seen in the time series so far, which is, the so-called time series streaming.

Analyzing time series data led to the decomposition of time series into components. Each component is defined to be a major factor or force that can affect any time series. Three major components of time series have been identified. A trend refers to the long-term tendency of a time series to rise or fall.

Stream time series is a set of time series data, T_1, T_2, $\ldots$, and T_m $(m > 1)$. The example of a stream time series environment is displayed in Figure 1.

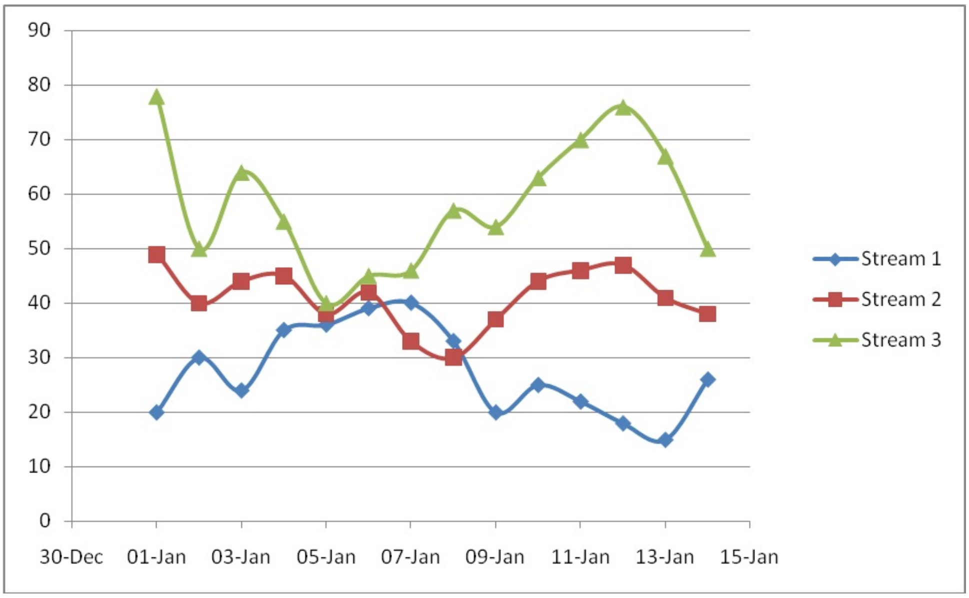


FIGURE 1. Example of stream time series environment

TABLE 1. Symbols and their descriptions

Symbol	Description
T_i	the i^{th} time series ($T_i = \{T_1, \ldots, T_n\}$ with $i = 1, \ldots, n$)
T'_i	the i^{th} time series after dimensionality reduction techniques
n	the number of time series $\{T_1, \ldots, T_n\}$
m	the number of data points in the stream $(T_{i0}, T_{i1}, \ldots, T_{i(m-1)})$
p	the number of data points before dimensionality reduction
H	the number of historical data used for prediction
F	the number of predicted values

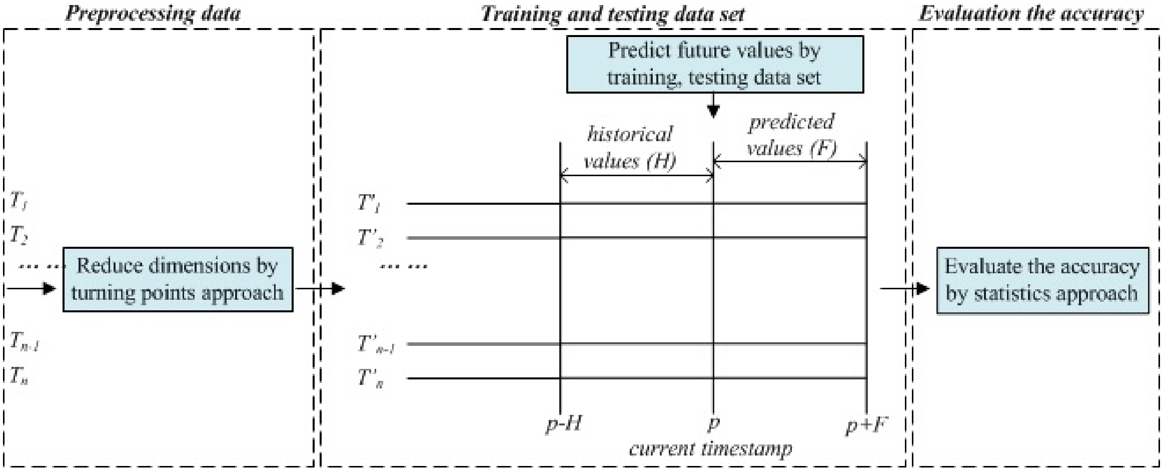

Preprocessing data
Training and testing data set
Evaluation the accuracy
Predict future values by training, testing data set
historical values (H)
predicted values (F)
T_1
T_2
……
T_{n-1}
T_n
Reduce dimensions by turning points approach
T'_1
T'_2
……
T'_{n-1}
T'_n
Evaluate the accuracy by statistics approach
p-H
p
p+F
current timestamp

FIGURE 2. Our proposed framework

With our approach, three principal processes in our framework are shown in Figure 2 and the notations in Table 1. In general, the approach has three main processes: preprocessing data, future values prediction period and evaluation process. The first process is data preprocessing and aims to reduce the dimensionality of time series. In this process, we apply the turning point with the defined patterns to reduce the dimensions. The next process is the training and testing of the historical data to predict future values. In this process, we apply SMO techniques in order to reduce the memory storage of dynamic programming. In the last one, we evaluate the accuracy based on indicators of statistical methods.

Definition 3.1. *Let us assume that we have n time series T_1, T_2, …, and T_n in the stream data environment, each T_i containing m ordered values at the current timestamp $(m-1)$, that is, $T_i = \{t_{i0}, t_{i1}, \ldots, t_{i(m-1)}\}$ where t_{ij} is the value at timestamp j in T_i.*

Definition 3.2. *Suppose that n stream time series only receive data after F timestamps, after applying perceptually important points and turning points' techniques, we have p time series. In other words, for each time series T'_p, the future values t_{ip}, $t_{i(p+1)}$, …, and $t_{i(p+F-1)}$ fitting to timestamps p, $(p+1)$, …, and $(p+F-1)$, respectively, arrive in a batch manner at the same timestamp $(p+F)$. At the period from time stamp p to $(p+F-1)$, the system does not know about F future values in each time series.*

Definition 3.3. *The objective of the prediction system is to efficiently predict the $n \times F$ values for n time series with the prediction error as low as possible and the accuracy as high as possible. The prediction error [23] is defined as the difference between the actual* *value and predicted value for the corresponding period.*

$$E_t = A_t - F_t \tag{1}$$

where E, A and F are the prediction error, actual value and predicted value at period t.

4. Our Approach.

4.1. Turning points.

A time series consists of a sequence of local maximal/minimal points and several of them mirror the information of data trend reversals. These local maximum and minimum points are called critical points; in another way, we can say that a time series is comprised of a sequence of critical points.

These critical points are often called *turning points* because they show the change in the trend of the time series data. For example, A, B, C, D and E points are turning points in Figure 3. Turning points are used over a wide area in data mining analysis since they contain more information than the other points. Turning points describe the trend of the time series data change and they can be also used to identify the beginning or end of a transaction period.

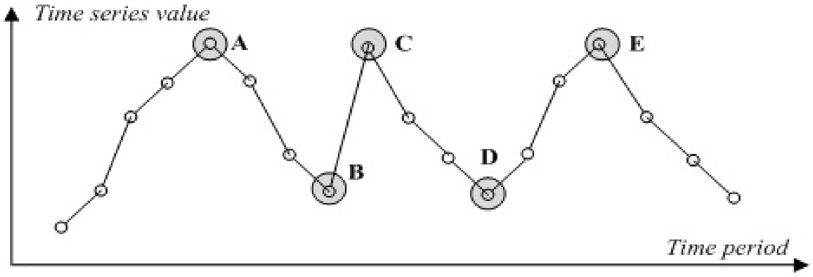


Figure 3. The definition of turning points

The turning point t_i in a time series T_i is a point with indication in two cases. The first case is if that point ends the increasing trend at t_i and starts a decreasing period, the next case is if that point ends the decreasing trend at t_i and starts an increasing period.

In our approach, we only consider the critical points of each time series in a certain period. The turning points in time series are defined as the points that separate two adjacent trends and have the shortest distance from the release time of announcements. Only some of the critical points are preserved (those critical points, which are considered as interference factors are removed).

All the local maximal/minimal points in original time series are extracted to form the initial critical point series. After constructing the initial critical point series T_i', a critical point selection criterion is applied to filter out the critical points corresponding to noise. The time series T_i and T_i' are called original and are arrived at after preprocessing the data.

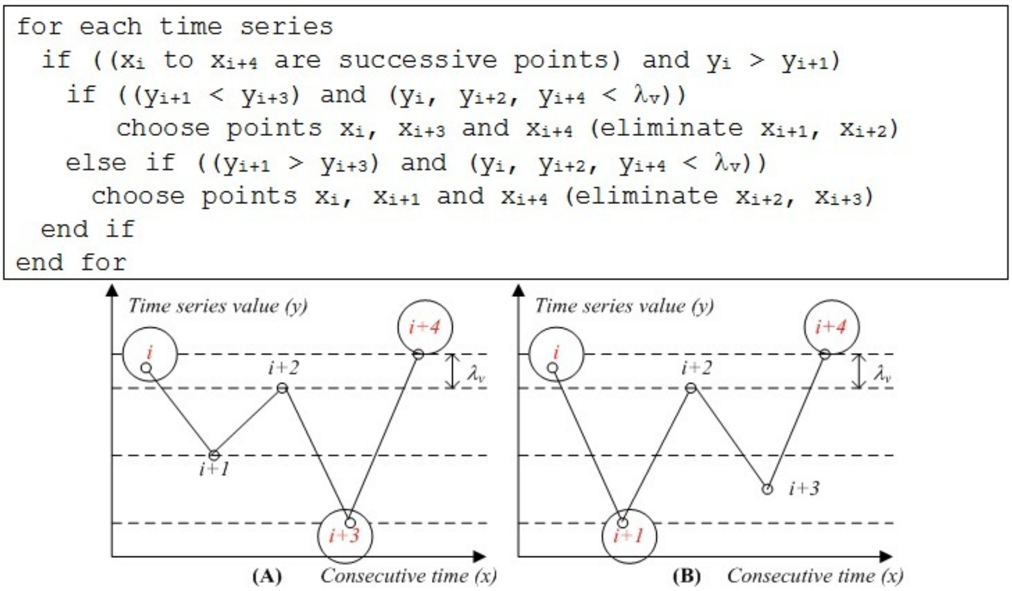
We suppose that the first and the last data points in the original time series T_i are preserved as the first and last points in T_i'. The direction for selection is also based on the fluctuation parameter λ_v and the time duration threshold λ_t. The λ_v is defined as Equation (2) below:

$$\lambda_v = \frac{1}{k} \times \sum |t_k| \tag{2}$$

where t_k is the k^{th} value of the time series data. Our strategies for eliminating the points that are not important are shown in detail below. The time duration threshold λ_t in our approach is 5 successive points.

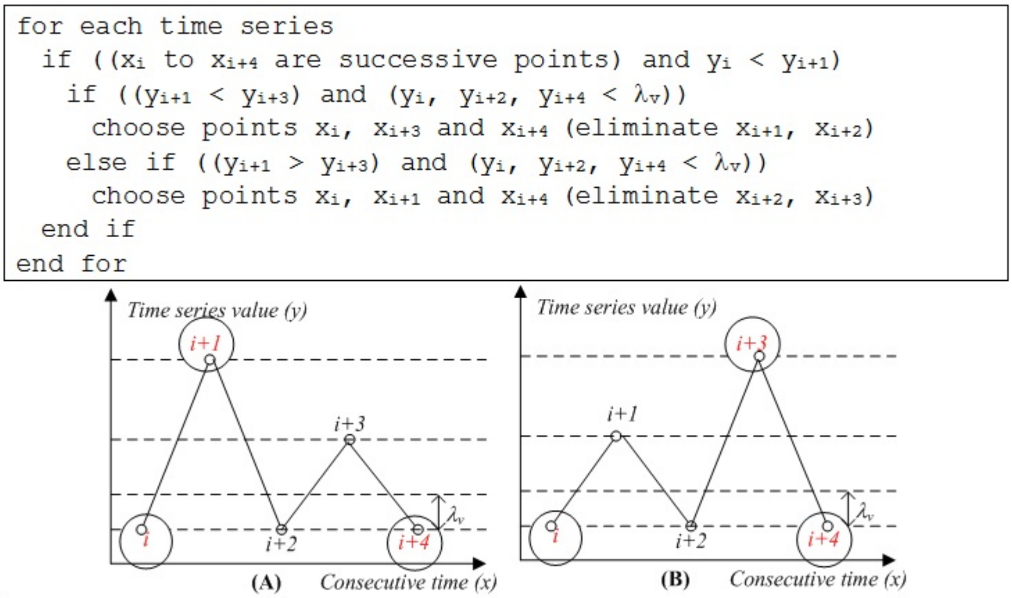
In a stream environment, for a given time series $T_{ij} = \{t_{i1}, t_{i2}, \ldots, t_{im}\}$, a turning point (peak or trough) in the time series T_{ij} is defined as any time period j of the i^{th} stream such that the change (decrease or increase) in the observations of the time series after the consideration of both the specific thresholds of fluctuation and time duration. It is not certain that the points are located on the upward or downward trend of each time series.

With the purpose of making less time and memory for implementing the framework, in this study, we propose three strategies with six cases for eliminating according to the turning point, they are shown in Figures 4, 5 and 6 (also including the pseudo code). In each strategy, the option for choosing or eliminating depends on the value parameter λ_v and time threshold λ_t. It means we consider both specific of data's fluctuation and time duration of each time series in a stream environment.



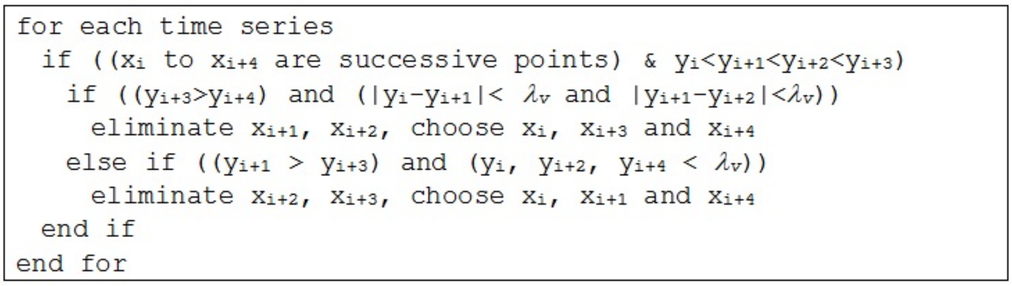
```
for each time series
  if (($x_i$ to $x_{i+4}$ are successive points) and $y_i > y_{i+1}$)
    if (($y_{i+1} < y_{i+3}$) and ($y_i$, $y_{i+2}$, $y_{i+4} < \lambda_v$))
        choose points $x_i$, $x_{i+3}$ and $x_{i+4}$ (eliminate $x_{i+1}$, $x_{i+2}$)
    else if (($y_{i+1} > y_{i+3}$) and ($y_i$, $y_{i+2}$, $y_{i+4} < \lambda_v$))
      choose points $x_i$, $x_{i+1}$ and $x_{i+4}$ (eliminate $x_{i+2}$, $x_{i+3}$)
  end if
end for
```

FIGURE 4. First strategy for eliminating points



```
for each time series
  if (($x_i$ to $x_{i+4}$ are successive points) and $y_i < y_{i+1}$)
    if (($y_{i+1} < y_{i+3}$) and ($y_i$, $y_{i+2}$, $y_{i+4} < \lambda_v$))
      choose points $x_i$, $x_{i+3}$ and $x_{i+4}$ (eliminate $x_{i+1}$, $x_{i+2}$)
    else if (($y_{i+1} > y_{i+3}$) and ($y_i$, $y_{i+2}$, $y_{i+4} < \lambda_v$))
      choose points $x_i$, $x_{i+1}$ and $x_{i+4}$ (eliminate $x_{i+2}$, $x_{i+3}$)
  end if
end for
```

FIGURE 5. Second strategy for eliminating points



```
for each time series
  if (($x_i$ to $x_{i+4}$ are successive points) & $y_i<y_{i+1}<y_{i+2}<y_{i+3}$)
    if (($y_{i+3}>y_{i+4}$) and ($|y_i-y_{i+1}|< \lambda_v$ and $|y_{i+1}-y_{i+2}|<\lambda_v$))
      eliminate $x_{i+1}$, $x_{i+2}$, choose $x_i$, $x_{i+3}$ and $x_{i+4}$
    else if (($y_{i+1} > y_{i+3}$) and ($y_i$, $y_{i+2}$, $y_{i+4} < \lambda_v$))
      eliminate $x_{i+2}$, $x_{i+3}$, choose $x_i$, $x_{i+1}$ and $x_{i+4}$
  end if
end for
```

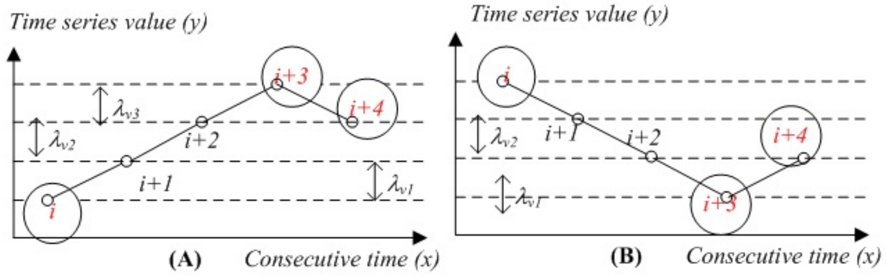


FIGURE 6. Third strategy for eliminating points

To ensure the change in the observations of the turning points in terms of both the time and value, we set a step range for eliminating the points that are not important. To avoid a false turning point, that locates on the upward or downward trend of the time series, in being identified as a true one, our strategy is to ensure that the type of the previously identified turning point is opposite (*downward or upward*) to that of the current point. It means that a peak must follow a valid trough closely, and no additional peak is located between them.

4.2. **Applying SMO algorithm for prediction.** The principal idea of support vector regression [3] is mapping the vector x *(input vector)* into high dimensional feature space by nonlinear mapping function Φ and then to perform linear regression in the feature space. This transformation is realized by Kernel function $K(x_i, y_i) = \Phi(x_i) \cdot \Phi(y_i)$. The Kernel function may be Gaussian, polynomial or neural network non-linearity.

Working with the large data set environment, the prediction using the Support Vector Machine algorithm [3] makes the operation speed slower. Especially, in the stream time series environment with many time series $T_1, T_2, \ldots, T_n$ with $T_i = \{t_{i0}, t_{i1}, t_{i2}, \ldots, t_{i(m-1)}\}$, if each one of the time series puts its own kernel matrix into the main memory, the primary memory will overflow.

Support vector machine has many ways to optimize the quadratic programming; SMO is the best way to resolve that problem. The reason we applied the prediction with SMO is the performance reducing satisfaction, SMO algorithm just calls the kernel matrix iteration; therefore, the performance is improved significantly. This implementation will reduce the main memory at run time. The reason we chose the SMO prediction algorithm [7,9,15,31] because it is based on the Support Vector Machine. In addition, the SMO algorithm just calls the kernel matrix iteration, and so the performance is improved significantly.

Examine the case about a binary classification issue with a dataset $\{(x_1, y_1),\ldots,(x_n, y_n)\}$, where x_i is an input vector and $y_i \in \{-1, +1\}$ is a binary label corresponding to it. A soft-margin support vector machine is trained in order to solve a quadratic programming problem, which is expressed in the dual form as Equation (3):

$$\max_{\xi} \sum_{i=1}^{n} \xi_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{c} y_i y_i K\left(x_i, x_j\right) \xi_i \xi_j \tag{3}$$

subject to

$$0 \leq \xi_i \leq C, \quad i = 1, 2, \ldots, n$$
$$\sum_{i=1}^{n} y_i \xi_i = 0$$

where C is an SVM hyperparameter and $K(x_i, x_j)$ is the Kernel function, both of them are supplied by the user; and the variables ξ_i are Lagrange multipliers.

The dot product in the feature space is equivalent to the Kernel function of the input space. Thus, instead of directly to the value of the scalar product, we made indirectly through Kernel function. In this research, we use the stock data as a nonlinear transformation, so that nonlinear Gaussian function (RBF-Radial Basis Function) can be chosen as the Kernel function.

$$K(x_i, y_i) = \exp\left(-\gamma \|x_i - y_i\|^2\right) \tag{4}$$

SMO is an iterative algorithm for solving the optimization problem of the SVM. SMO algorithm implements the dividing problem into a series of smallest possible sub-problems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers, the smallest possible problem involves two such multipliers. Then, for any two multipliers ξ and ξ^*, the constraints are reduced to:

$$\begin{aligned} &0 \leq \xi, \quad \xi^* \leq C \\ &y_1 \xi + y_2 \xi^* = k \end{aligned} \tag{5}$$

where C is an SVM hyper parameter and this reduced problem can be solved analytically.

Our algorithm proceeds as below for each time series:

1. *Looking for a Lagrange multiplier that violates the Karush-Kuhn-Tucker [25] conditions for the optimization problem.*
2. *Select the second multiplier ξ^*, optimize the pair (ξ, ξ^*).*
3. *Repeat two steps 1 and 2 until convergence.*

When all the Lagrange multipliers, ξ, ξ^* satisfy the KKT conditions (within a user-defined tolerance), it means the problem has been solved. Although the SMO algorithm is guaranteed to converge, heuristics are used to choose the pair of multipliers to accelerate the rate of convergence.

We select the first Lagrange multiplier by using the external loop of the SMO algorithm to enable the Lagrange multiplier to optimize. Choosing the second Lagrange multiplier is according to maximizing the step length of the learning during joint optimization. $|E1 - E2|$ is used to approximate the step size in SMO [7,9,15].

4.3. **Prediction accuracy.** To ensure the accuracy of prediction results, the proposed framework must be evaluated accurately, and its performance generalized before it can be used to predict. The prediction error is defined as the difference between the actual value and the predicted value for the corresponding period as in Equation (1).

In order to evaluate how accurately of the proposed framework, we need to choose a suitable way that describes the data. Therefore, we use statistics to evaluate the simulation effect and predictive validity of our framework. There are several types of evaluations

[11], such as mean absolute error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE).

Specifically, our approach used RMSE and MAE to evaluate the results of the framework. The MAE and RMSE are given by Equations (6) and (7). Mean absolute error is the measure of the deviation between the real values and predicted values. The smaller the values of MAE, the closer are the predicted time series values to the actual values (a smaller value suggests a better predictor). Root mean squared error is a frequently used measure of the differences between values predicted by an estimator and the values actually observed from the thing being estimated. In the statistical approach, RMSE is a good measure of accuracy.

$$MAE = \frac{1}{n}\sum_{t=1}^{N}|E_t| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{N}|E_t^2|} \tag{7}$$

Besides, in this paper, the accuracy of n time series T_1, T_2 and T_n will use the average of each time series. The average is computed with Equation (8):

$$\overline{E} = \frac{1}{n}\sum_{t=1}^{N}E_t \tag{8}$$

5. Experimental Evaluation.

5.1. Experimental environment and dataset.

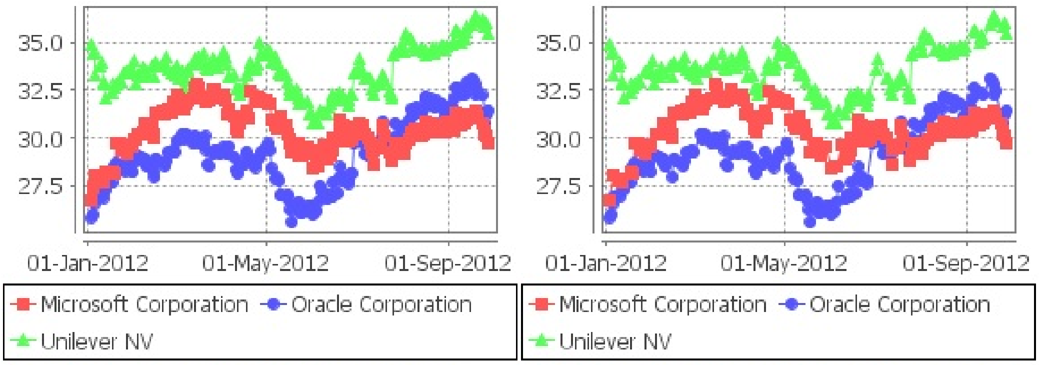
The experimental dataset used was the financial stock time series data. The stock data set contains daily, weekly and monthly open stock prices obtained from Yahoo Finance. The experiment was performed on a 2G AMD PC with 2 GB of main memory and running the Windows XP operating system.

We tested our approaches with four different stock companies' data (HBC – HSBC Holding plc, ORCL – Oracle Corporation, MSFT – Microsoft Corporation and UN – Unilever NV, HPQ – Hewlett-Packard Company). From 01 January 2012 to 30 September 2012, we collected a series of 188 daily stock quotes, which are composed of seven raw data elements' date, opening price, high price, low price, closing price, trading volume and adjust closing price. The period under consideration is the stock ticker's daily closing price. We used 80% of the dataset after the preprocessing process as the training set, and the other 20% was used as the testing set.

5.2. Experimental results and analysis.

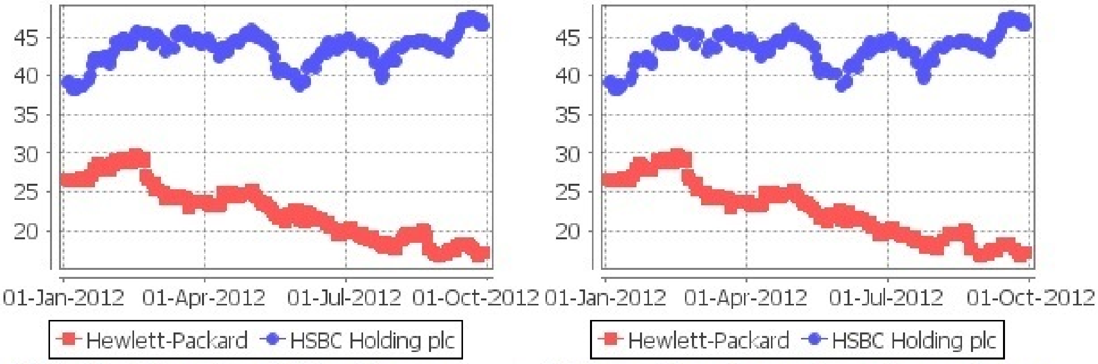
In this section, we talk about the experiments and the results from this study to assess our proposed method. Our discussion is constructed with the following three aspects, and the experimental process can be described as follows: First, we reduce the dimensions of the stream time series in the preprocessing step. Second, we consider the relationship between the number of history price stock-values (after filtering the turning points) and the predicted number of future values. Third, we perform the comparison of our approach with other techniques.

5.2.1. *Data preprocessing.* First of all, we apply the preprocessing step in order to reduce the dimensions of the historical data. With the aim to keep the shape of the original data, we implement the experimental in case of low and high fluctuations. In Figure 7(a), we show the original closing stock data of ORCL, MSFT and UN. As shown, the data does not have a large variability within the nearest 9 months. The characteristic of the daily stock data does not exist on the weekends and some special days. As observed in the figure, the variances of ORCL, MSFT and UN stock price streams are in low fluctuation. Figure 8(a) displays the historical data of HBC and HPQ; this data has a higher fluctuation.



(a) Original data from Yahoo Finance (b) Data after using turning points technique

Figure 7. Stock streams of MSFT, ORCL and UN (the case of low fluctuation data)



(a) Original data from Yahoo Finance (b) Data after using turning points technique

Figure 8. Stock streams of HBC and HPQ (the case of high fluctuation data)

With the research of statistics and probability theory, the standard deviation presents how much variation exists for the average. It shows that lower standard deviation indicates the data points tend to be very close to the mean; the high standard deviation indicates the data points are spread out over a large range of values. The standard deviation of the stock stream data is a description of the risk associated with price-fluctuations. Table 2 provides the common statistical parameters of the daily stock prices of the five different companies.

In this case, the original point of each stock time series data is about 188 observations. The standard deviations of the tested daily stock streams MSFT, ORCL and UN are low, so the predicted values are more accurate than in the other cases of HBC and HPQ. We also provide the number of data points after dimensionality reduction in Table 2. After using turning points as dimensionality reduction technique, the values of mean and standard deviation have a little change, so we can keep the shape of the original data. The number of points is different with each stock time series. All original data points must be checked through our process, we only keep the turning points which are satisfied both time threshold and value parameters. As a mention above, our strategies for eliminating points are shown in Figures 4, 5 and 6.

Table 2. Statistics of daily stock streams for prediction

	Mean value		Standard deviation		After reducing	
	Original	After preprocess	Original	After preprocess	No. of point	% reducing
HBC	43.511	43.451	2.088	2.11	146	22.34%
HPQ	22.512	21.787	3.692	3.875	152	19.15%
ORCL	29.235	29.061	1.779	1.771	150	20.21%
MSFT	30.433	30.513	1.205	1.172	148	21.28%
UN	33.597	33.557	1.2	1.197	154	18.08%

We present the results data after implementing the turning point proposed approach in Figures 7(b) and 8(b).

5.2.2. *Predictive analysis.* In this research, the value of training and testing data set depends on how long the observation is and how far ahead we want to predict. In time series predictive analysis, one step predicts may not be as relevant as multi step predicts. We implement as follows: First, we select the observation ith for the test data set, treat the remaining observations as the training data set, compute the error on the test observation. We repeat the first step N times where N is the total number of observations. Then we calculate the accuracy measures of the prediction process based on the errors obtained.

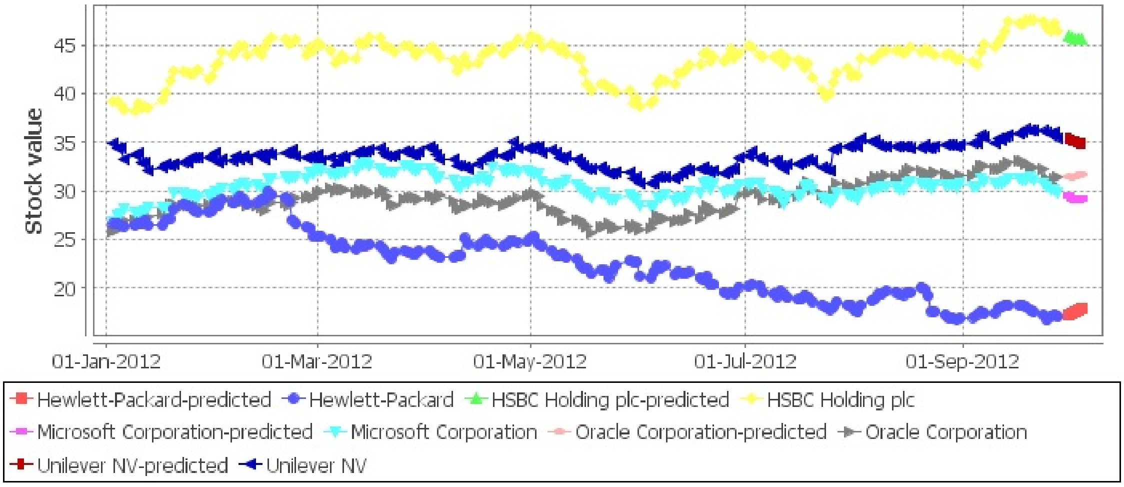


Figure 9. Future prediction values of stock stream with original data

Table 3. Evaluation of training data of the stock price stream (evaluated by RMSE)

Data	Original data			After preprocessing data		
Step(s)-ahead	1	2	3	1	2	3
HBC	0.3796	0.4597	0.4949	0.3375	0.3733	0.414
HPQ	0.2094	0.2901	0.3598	0.3159	0.3756	0.4438
ORCL	0.2795	0.3478	0.3842	0.3035	0.3498	0.3924
MSFT	0.2608	0.3128	0.345	0.2772	0.3076	0.3348
UN	0.1554	0.2119	0.2443	0.2643	0.3049	0.3272
$\overline{E}$	0.25694	0.32446	0.36564	0.29968	0.34224	0.38244

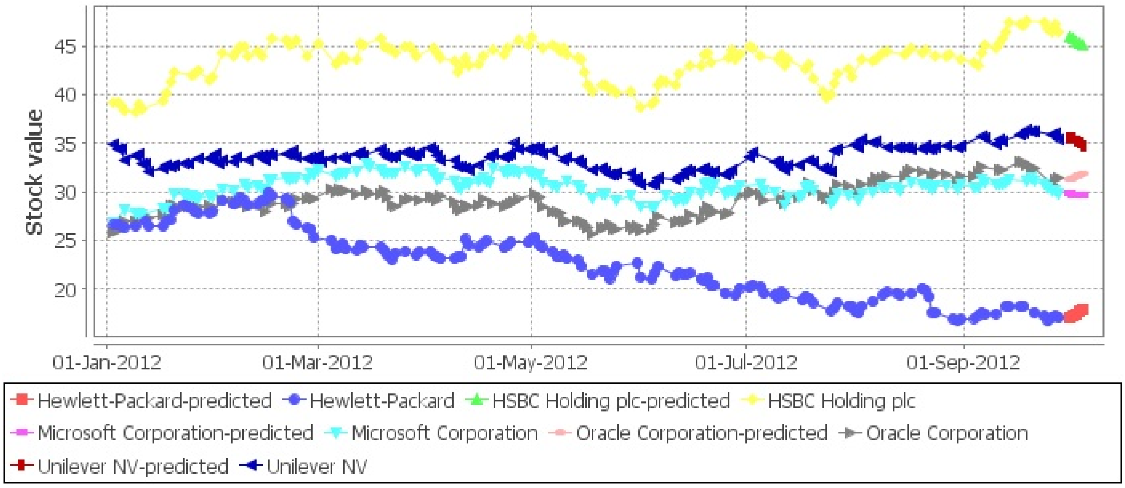
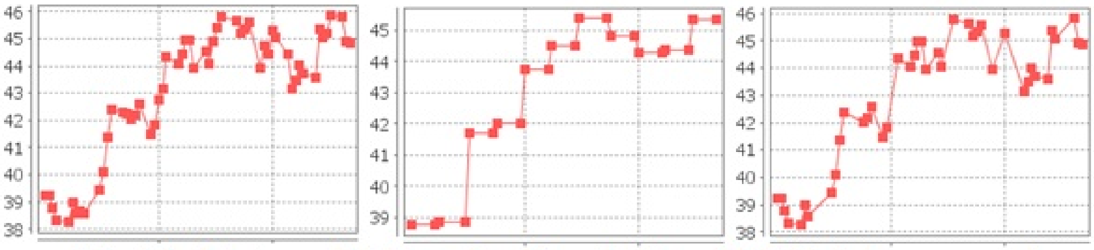


Figure 10. Future prediction values of stock stream with data after preprocessing



(a) original data, (b) dimensionality reduction with PAA, (c) our approach.

Figure 11. The comparison of our approach and PAA technique

In Figure 9, we show both history and future prediction values of 5 stock streams in the case of using the turning point technique. The results of training data are shown in Table 3, which are evaluated by RMSE of both original data and data after dimensionality reduction technique. For prediction of the next five days, we need five steps ahead in the training process (we only display three first steps). In order to ensure the accuracy of prediction results, the model must be evaluated accurately, and its performance generalized before it can be used to predict. The RMSE in Equation (7) is lower in the case of using dimensionality reduction data. Figure 10 shows the predictive analysis with preprocessing data.

We use the grid search method and cross through assessment (5-fold cross validation) to find the optimal values for input parameters C and γ parameters. The value is limited to the range: $C \in [2^{-5}, 2^{15}]$ and $\gamma \in [2^{-15}, 2^{3}]$.

5.2.3. *Performance comparison.* A comparison of our approach with other dimensionality reduction techniques is shown in Figure 11 (Figure 11(a) shows the original time series data; Figure 11(b) and Figure 11(c) display the result of PAA dimensionality reduction and our approach of the original data).

PAA implements the dimensionality reduction of a time series from N in the original space to N in the reduced space by segmenting the time series into equal-sized frames and representing each segment by the mean of the data points that lie within that frame. The PAA method is simple and straightforward but the result does not retain the shape of the

Table 4. Comparison of performance accuracy with the actual values

Date	Linear regression	SMO based on SVM	Actual value
01 October 2012	31.7694	31.6621	31.39
02 October 2012	31.8187	31.8083	31.9
03 October 2012	31.8713	31.8708	31.82
04 October 2012	32.0377	31.9592	31.65
04 October 2012	32.2507	31.9354	31.67

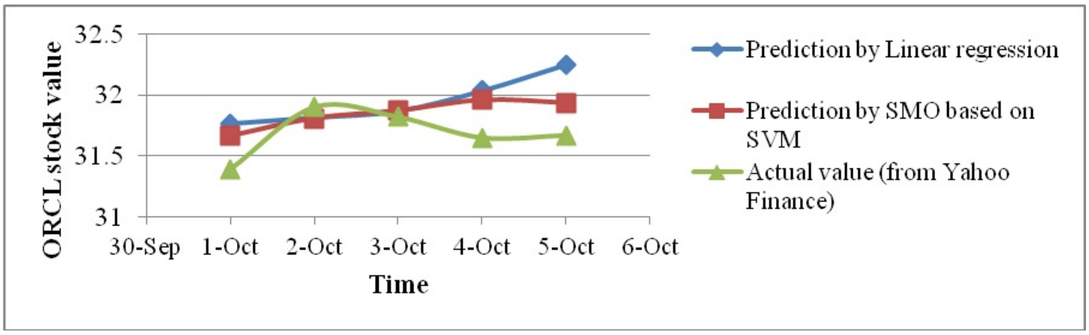


Figure 12. The comparison of actual values and predicted values

change in the time series stock data. In spite of this, the PAA method has been successfully used as a competitive method. However, our approach for dimensionality reduction still endures one deficiency, that is, it complicated to implement and its computational complexity is higher than that of the PAA technique.

In Table 4, we exhibit the performance of the predictive analysis of Linear Regression and SMO based on SVM in the stream data environment, and the actual value retrieved from Yahoo Finance with MSFT daily stock stream. In Figure 12, we show the predictive value of our approach and the actual value retrieved from Yahoo Finance with ORCL daily closing stock stream. This figure includes one line of actual value, two lines of predicted values by Linear Regression and SMO algorithm based on SVM. The predicted values of the SMO algorithm approximate the actual values.

6. **Conclusions and Future Work.**

We have studied the problem of time series prediction with dimensionality reduction of large historical data to a smaller data set and then using the data mining technique to compute future values.

In this paper, we have shown and theoretically explained the use of the turning point approach to reduce dimensions and used the SMO algorithm based on the SVM prediction, and provided the evaluation indicators of accuracy and generalization. After the dimensionality reduction method based on the turning point process, the time series generated by our approach still maintains the shape of the original trends. The approach is very meaningful in the environment where the data set is large. Therefore, we applied this work on the stock price data set obtained from Yahoo Finance.

In the future, we propose to use the segmentation technique as a process in the analysis of time series data that will improve the efficiency and accuracy of prediction values. In addition, we plan to extend the current work towards pattern discovery, such as in similarity search and finding motifs techniques in the stock time series data.

Acknowledgment. The work was supported by the National Natural Science Foundation of China (Grant no. 61240046) and the Science and Technology Planning Project of Hunan Province (Grant no. 2011FJ3048). We would like to thank the University of Waikato for providing machine learning open source [14]. And we also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] X. Bai and C. Zhao, Research on time series forecasting model based on support vector machines, *Proc. of the International Conference on Measuring Technology and Mechatronics Automation*, vol.3, pp.227-230, 2010.

[2] D. Bao and Z. Yang, Intelligent stock trading system by turning point confirming and probabilistic reasoning, *Expert Systems with Applications*, vol.34, no.1, pp.620-627, 2008.

[3] D. Bao, A generalized model for financial time series representation and prediction, *Applied Intelligence*, vol.29, no.1, pp.1-11, 2008.

[4] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, 3rd Edition, Prentice-Hall, New York, NY, 1994.

[5] K. P. Burnham and D. R. Anderson, Multimodel inference: Understanding AIC and BIC in model selection, *Sociological Methods and Research*, vol.30, no.2, pp.261-304, 2004.

[6] A. Camerra, T. Palpanas, J. Shieh and E. Keogh, iSAX 2.0: Indexing and mining one billion time series, *Proc. of the IEEE 10th International Conference on Data Mining*, pp.58-67, 2010.

[7] Z. Chen and Y. Yang, *Assessing Forecast Accuracy Measures*, http://www.stat.iastate.edu/preprint/articles/2004-10.pdf, 2004.

[8] T. Fu, A review on time series data mining, *Engineering Applications of Artificial Intelligence*, vol.24, no.1, pp.164-181, 2011.

[9] T.-C. Fu, F.-L. Chung, K.-Y. Kwok and C.-M. Ng, Stock time series visualization based on data point importance, *Engineering Applications of Artificial Intelligence*, vol.21, no.8, pp.1217-1232, 2008.

[10] T.-C. Fu, F.-L. Chung, R. Luk and C.-M. Ng, Representing financial time series based on data point importance, *Engineering Applications of Artificial Intelligence*, vol.21, no.2, pp.277-300, 2008.

[11] T.-C. Fu, F.-L. Chung, R. Luk and C.-M. Ng, Stock time series pattern matching: Template-based vs. rule-based approaches, *Engineering Applications of Artificial Intelligence*, vol.20, no.3, pp.347-364, 2007.

[12] E. Fuchs, T. Gruber, J. Nitschke and B. Sick, Online segmentation of time series based on polynomial least-squares approximations, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.32, no.12, pp.2232-2245, 2010.

[13] H. Li, C. Guo and W. Qiu, Similarity measure based on piecewise linear approximation and derivative dynamic time warping for time series mining, *Expert Systems with Applications*, vol.38, no.12, pp.14732-14743, 2011.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, The WEKA data mining software: An update, *SIGKDD Explorations*, vol.1, no.1, pp.10-18, 2009.

[15] J. C. Platt, Sequential minimal optimization: A fast algorithm for training support vector machines, *Microsoft Research*, 1998.

[16] B. P. Joshi and S. Kumar, Intuitionistic fuzzy sets based method for fuzzy time series forecasting, *Cybernetics and Systems: An International Journal*, vol.43, pp.34-47, 2012.

[17] X. Li and T. Tian, Time series recognition based on wavelet transform and fourier transform, *Proc. of IEEE Symposium on Industrial Electronics and Applications*, pp.722-726, 2010.

[18] K. Mehdi and B. Mehdi, A novel hybridization of artificial neural networks and ARIMA models for time series forecasting, *Applied Soft Computing*, vol.11, no.2, pp.2664-2675, 2011.

[19] D.-Y. Men and W.-Y. Liu, Application of least squares support vector machine (LS-SVM) based on time series in power system monthly load forecasting, *Proc. of Power and Energy Engineering Conference (APPEEC)*, pp.1-4, 2011.

[20] S.-H. Park, J.-H. Lee, S.-J. Chun and J.-W. Song, Representation and clustering of time series by means of segmentation based on PIPs detection, *Proc. of the 2nd International Conference on Computer and Automation Engineering*, vol.3, pp.17-21, 2010.

[21] J. C. Platt, Sequential minimal optimization: A fast algorithm for training support vector machines, *Advances in Kernel Methods*, pp.185-208, 1999.

[22] C.-C. Wang, A comparison study between fuzzy time series model and ARIMA model for forecasting Taiwan export, *Expert Systems with Applications*, vol.38, no.8, pp.9296-9304, 2011.

[23] J. R. Wang and X. L. Deng, Selecting training points of the sequential minimal optimization algorithm for support vector machine, *Proc. of the 2nd International Conference on Control, Instrumentation and Automation*, pp.456-458, 2011.

[24] G. E. Guraksin and H. Uguz, Classification of heart sounds based on the least squares support vector machine, *International Journal of Innovative Computing, Information and Control*, vol.7, no.12, pp.7131-7144, 2011.

[25] W. K. Wong, E. Bai and A. W. C. Chu, Adaptive time variant models for fuzzy time series forecasting, *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics*, vol.40, no.6, pp.1531-1542, 2010.

[26] L. Xian and C. Lei, Efficient methods on predictions for similarity search over stream time series, *Proc. of the 18th International Conference on Scientific and Statistical Database Management*, pp.241-250, 2006.

[27] L. Xian and C. Lei, Efficient similarity search over future stream time series, *IEEE Transactions on Knowledge and Data Engineering*, vol.20, no.1, pp.40-54, 2008.

[28] L. Xian, C. Lei, X. Y. Jeffrey, H. Jimsong and M. Jian, Multiscale representations for fast pattern matching in stream time series, *IEEE Transactions on Knowledge and Data Engineering*, vol.21, no.4, pp.568-581, 2009.

[29] J. F. Yang, Y. J. Zhai, D. P. Xu and P. Han, SMO algorithm applied in time series model building and forecast, *Proc. of the 6th International Conference on Machine Learning and Cybernetics*, vol.4, pp.2395-2400, 2007.

[30] Q. Wang and V. Megalooikonomou, A dimensionality reduction technique for efficient time series similarity analysis, *Information Systems*, vol.33, no.1, pp.115-132, 2008.

[31] C.-Y. Yeh, C.-W. Huang and S.-J. Lee, Multi-kernel support vector clustering for multi-class classification, *International Journal of Innovative Computing, Information and Control*, vol.6, no.5, pp.2245-2262, 2010.