CHANGE POINT DETECTION IN TIME SERIES DATA USING SUPPORT VECTORS

FATIH CAMCI

Computer Engineering Department
Fatih University, Istanbul, Turkey
fcamci@fatih.edu.tr

Change Point Detection in time series data is of interest in various research areas including data mining, pattern recognition, statistics, etc. Even though there are several effective methods in the literature for detecting changes in mean, and an increase in variance, there are none for decrease in variance. Effective detection of decreased variance has been reported as future work in earlier papers. In addition, most, if not all, methods require some model like AR to fit into the time series data in order to extract noise information, which is assumed to be independent and identically distributed (i.i.d.) and follow standard normal distribution (white noise). Thus, effectiveness of the methods is tied to the fitness degree of the AR model to the time series data. This paper presents a change point detection method based on support vectors that targets changes in mean and variance (including variance decrease) without any assumption of model fitting or data distribution. The data is represented by a hyper-sphere in a higher dimensional space using kernel trick. The change is identified by the change in the radius of the hyper-sphere. A comparison of this method with other methods is presented in the paper.

Keywords: Change point detection; data mining; time series; support vector machine; variance change detection.

1. Introduction

Change point detection in time series data is of interest in various research areas including pattern recognition, economics–finance, ecosystems, health, mathematics, network security, and data mining. Change point detection (also called activity monitoring, event change detection) is the process of identifying the time point, in which the properties of data (e.g. mean, variance) suddenly change. Change in heart rate trend, ecological indicators due to climate changes, yen/$ exchange rate, and stock indexes are examples of application areas for the change point detection problem.

Change point detection focuses on finding the transition point from one underlying time series generation model to another. Change point is mostly represented as sudden change in either mean or variance, or both. Existing methods in the literature are able to effectively detect the changes in mean and increase in variance; however,
this is not true in variance decrease.\textsuperscript{6,9,11,13,14,16,20,25,33–35} In addition, most, if not all, methods require some model like Auto-Regressive (AR) to fit into the time series data in order to extract the noise information for effective detection. The effectiveness of the methods is tied to the fitness degree of the AR model to the time series data since noise is assumed to be independent and identically distributed (i.e. i.i.d) and follow standard normal distribution, which is called white noise. These two issues, which have also been reported as future work in Ref. 25 by Takeuchi and Yamanishi, have derived the research of this paper. In Ref. 25, the design of an algorithm to detect change points, in which variance decreases, and the extension of the proposed method to the more complex time series model (auto-regressive integrated moving average: ARIMA) are stated as future work. This paper aims to address these issues by introducing a Support Vector based Change Point Detection (SVCPD) method targeting changes in variance and/or mean without any assumption of model fitting or data distribution. SVCPD does not use any time series model (AR or ARIMA) for model fitting, targets both increase and decrease in mean and variance, and compares the results with the methods presented in Ref. 25 and others.\textsuperscript{6,11,15}

Sections 2 and 3 present related work and problem setting, respectively. Section 4 discusses support vector based change detection method. Section 5 illustrates the presented method in benchmarking data sets and reports its comparison with other methods. Section 6 concludes the paper.

2. Related Work

Change point detection methods can be categorized in two groups as posterior (off-line) and sequential (on-line). Posterior methods collect data first, then try to detect the change points by analyzing all collected data, whereas sequential methods receive data sequentially and analyze previously obtained data to detect the possible change in current time. The primary interest in most application domains such as finance, failure detection, health, network security, etc. is sequential (on-line) analysis. Our method is also based on sequential analysis.

Early work in change detection based on the statistical test in mean and/or variance such as Autoregressive time series,\textsuperscript{10} Autoregressive Integrated Moving Average (ARIMA)\textsuperscript{15} for changes in mean and CUmulative SUM of squares (CUSUM) test\textsuperscript{15} for changes in variance. Similarly, MOving SUM of squares (MOSUM)\textsuperscript{13} and Modified Levene’s Moving Block (MLMB)\textsuperscript{6} are applied in finance and ecosystem data, respectively. In these methods, a window size is defined and the window is shifted by one data as time used. The properties of data within the window are identified and compared with previous windows using sum of squares or standard deviations. The probability of change is defined as a parameter such as p-value obtained by Modified Levene’s test. Since these tests can be applied to independent identically distributed (i.i.d) data, the time series data should be fitted to an AR model and the difference between data and AR models (i.e. noise) should be
calculated, which is expected to be i.i.d. The results to be obtained are highly dependent on the fitness degree of AR model to the time series data and noise i.i.d. degree.

In Ref. 11, a segmented function is used to fit time series data and change point is detected when the total fitting error is minimized for earlier and later data. Takeuchi and Yamanishi have presented a unified framework for change point and outlier detection in Ref. 25 and reported better results than in Ref. 11. Their method is a variant of maximum likelihood method for AR model that gradually discounts the effects of past data by forgetting them as time used. This method calculates Kullback–Leibler divergence real-time for fast change detection. Even though the results obtained using these methods are effective for mean change and variance increase; they are not able to detect variance decrease as stated and illustrated in the paper. Even though novelty detection method for nonstationary data presented in Ref. 3 removes the requirement of fitness to an AR model and reports better results than the method presented in Ref. 25 for mean change and variance increase, its detection ability for variance decrease is poor.

Artificial neural network (ANN) is also employed in Ref. 20 for change detection. However, data modeling for possible changes are required for training ANN, which is not applicable in most domains. Even if some changes can be modeled, there will always be some other possible un-modeled changes.

In some change detection methods,8,21 the detection algorithms are applied to time-frequency information such as wavelet coefficients21 or time-frequency subimages extracted from Wigner–Ville time frequency representation.8 In Ref. 8, a dissimilarity measure between data points based on kernel theory is presented and a threshold is applied on the presented dissimilarity measure for change point detection. The focus of the applications presented in the paper is the detection of changes in the frequency components of the given signal rather than changes in mean and variance. Even though these methods8,21 present good results for changes in the time-frequency domain, mean and variance change in time domain may not appear significant enough in time-frequency domain. This paper focuses on changes in time domain. Readers are referred to Refs. 1, 2 and 12 for more details about change point detection.

As a summary, the existing change detection methods generally suffer from the following:

- Inability or inefficiency in detecting variance decrease.
- Assumption about the statistical distribution of the data, which are generally obtained as error of fitting the AR model to the time series data.
- Necessity of training the model with possible changes.

The aim of this paper is to present a support vector based change detection method that removes the difficulties mentioned above. The presented method will be compared with methods in Refs. 6, 11, 15 and 25. These methods report results with change in mean and variance. Even though their results in change in mean and
increase in variance are good, they are inefficient in the detection of variance decrease. Since the presented method is based on support vector based one-class classification, it is discussed in the following subsection.

2.1 Support vector based one-class classification

This section discusses support vector based one-class classification and its usage in change point detection will be discussed in Sec. 4. Support vector machine (SVM) is a famous classification tool used for many purposes in the literature from credit card fraud detection to face recognition.3,5,11,17,18,22–24,31,32,36 Even though SVM was designed for two-class classification initially, it has been used for multi-class classification17 and one-class classification.18,26 Support vector based one-class classification gives the minimum volume closed spherical boundary around the data that belong to the class under consideration, represented by center $c$ and radius $r$. Minimization of the volume is achieved by minimizing $r^2$, which represents structural error19:

$$\text{Min } r^2$$

Subject to : $\|x_i - c\|^2 \leq r^2 \quad \forall i, \quad x_i: \text{ith data point}$ (2)

The formulation above does not allow any data to fall outside of the sphere. In order to make provision within the model for potential outliers within the training set, a penalty cost function for data that lie outside of the sphere is introduced as follows:

$$\text{Min } r^2 + C \sum_i \xi_i$$

Subject to : $\|x_i - c\|^2 \leq r^2 + \xi_i, \quad \xi_i \geq 0 \quad \forall i$ (4)

where $C$ is the coefficient of penalty for each outlier and $\xi_i$ is the distance between the $i$th data point and the hyper-sphere. This quadratic optimization problem is then converted to its dual form by introducing Lagrange multipliers ($\alpha_i, \alpha_i \geq 0$) for constraints as in (5) and (6). The Lagrange multipliers for data points located inside of the hyper-sphere will be zero ($\alpha_i = 0$). If the Lagrange multiplier is nonzero less than $C(0 < \alpha_i < C)$, then the data point is located on the boundary of the hyper-sphere and called support vector. Note that the data set can be represented by these support vectors. If it is equal to $C(\alpha_i = C)$, then the data point is located outside of the hyper-sphere.29

$$\text{Max } \sum_i \alpha_i (x_i \cdot x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j)$$

Subject to : $0 \leq \alpha_i \leq C \quad \forall i, \quad \sum_i \alpha_i = 1$ (6)

where, $x_i \cdot x_i$ is the inner product of $x_i$ and $x_j$. 

76  F. Camci
In general, it is highly unlikely that a hyper-sphere can offer a good representation for the boundary of data in the “original input space”. Hence, data ought to be transformed to a “higher dimensional feature space” where it can be effectively represented using a hyper-sphere. SVMs employ kernels to achieve this transformation without compromising computational complexity. Thus, the dot product in (5) is replaced by a Kernel function, leading us once again to the following quadratic programming problem:

$$\text{Max } \sum \alpha_i K(x_i, x_i) - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j)$$

Gaussian kernel among several others has been shown to particularly offer better performance over other kernels for one-class classification problems.\textsuperscript{18,26} The issue with Gaussian kernel is optimization of the scale parameter, which has led several different support vector based one-class classifiers.\textsuperscript{4,27,37} As seen from the equations above, no assumptions about data distribution or independency have been made.

3. Problem Formulation

Formal definition of the change point detection problem is presented in this section. Consider a time series data \((x_1, x_2, x_3, \ldots)\), where each data point \((x_i)\) is a \(d\)-dimensional real valued vector drawn from a certain stochastic process \((p)\), which is a probability density function of joint probability distribution. Change point is defined as the time that the stochastic process has changed, say from \(p^1\) to \(p^2\). Assume that data are one-dimensional following Gaussian distribution \(p^1\) from time 1 to \(t - 1\) and \(p^2\) from \(t\) to \(n\), where \(p^1 \sim N(\mu_1, \sigma^2_1)\), \(p^2 \sim N(\mu_2, \sigma^2_2)\). Mean change occurs when \(\mu_1 \neq \mu_2\) and \(\sigma^2_1 = \sigma^2_2\), variance change occurs when \(\mu_1 = \mu_2\) and \(\sigma^2_1 \neq \sigma^2_2\), and both occur when \(\mu_1 \neq \mu_2\) and \(\sigma^2_1 \neq \sigma^2_2\). The goal of change detection methods is to detect the change as soon as possible after the change occurs.

Change detection problem is the process of identifying \(y_1\) through \(y_{m-1}\) given the time series data \((x_1^1 x_2^1 \cdots x_i^1 x_2^2 \cdots x^2 \cdots x_{y_m-1}^m x_1^m x_2^m \cdots x_{y_m}^m)\), where \(x_i^1, x_2^2, \ldots, x_{y_m-1}^m, x_1^m, x_2^m, \ldots, x_{y_m}^m\) have the same statistical properties (mean and variance) for a given \(i\), different properties for different \(i\)’s. The problem can be represented as follows:

$$\text{Find } y_1 \cdots y_{m-1} \text{ given } x_1^1 x_2^1 \cdots x_i^1 x_2^2 \cdots x^2 \cdots x_{y_m-1}^m x_1^m x_2^m \cdots x_{y_m}^m,$$

where \(\mu_i = \mu_j\) and \(\sigma_i = \sigma_j \Leftrightarrow i = j\).

In many change detection methods, time series data is fitted to an AR model. Some methods in the literature directly use noise data \((\varepsilon_t)\) assuming that the AR model is already fitted to the time series data.\textsuperscript{6} Others estimate an AR model and calculate the noise within the model.\textsuperscript{25} The fitness of the data to the AR model play a key role in the effectiveness of the models. This may not been seen as a problem in the simulated data since the data is obtained using an AR model. However, the real data might not fit an AR model perfectly causing noise data to be different from expected.
Our model neither requires an AR or some other model to fit the time series data nor does it make any assumption about the distribution of the data or noise.

4. Support Vector Based Change Detection (SVCPD)

Support vector machine is a powerful tool that has been used in many problems such as one-class, multiclass classification, and regression. Our method (Support Vector based Change Point Detector — SVCPD) is based on one-class classification modeling of support vector machine. A hyper-sphere with minimum value that captures the data is drawn in order to represent the given class. The data, which is not really a sphere in the original space, becomes a hyper-sphere in higher dimensional space without increasing computational complexity due to the kernel trick. All data inside and outside of the hyper-sphere are labeled as in-class and out-of-class, respectively. In our method, this hyper-sphere is employed instead of statistical properties in order to represent the data.

Similar to Ref. 6, our method also employs a window (with size \( w \)) that is shifted as new data becomes available through the time series data. A hyper-sphere representing the given data within the window is created. The window size is related to the sensitivity of the method to the changes. When a small size is used, the method becomes sensitive to the changes (fast detection) with high false alarms. On the contrary, big size window leads to slow change in detection with low false alarms. The selection of window size depends on the nature of the data. Data with high noise might require large window size in order to reduce the false alarms. Small window size will cause fast detection for clean data.

Figure 1 illustrates the data and hyper-sphere representing the data in two-dimensional space. First, initial \( w \) number of data points are used to create the first hyper-sphere \((h_1)\). The radius of the hyper-sphere is represented as \( r_1 \). When new data becomes available, the oldest data point (data point of far most time to the current time) is dropped from the window and newly available data is added to the window. Another hyper-sphere is created with data in the updated window. This process is repeated as new data becomes available. Hyper-spheres \( h_1 h_2 h_3 \cdots \) with radii of \( r_1 r_2 r_3 \cdots \) are obtained.

The newly available data might be either similar to (i.e. not a change point) or different from (i.e. a change point) previous data within the window. In the former case, the newly available data will be within the new hyper-sphere \((h_n)\) and the radius of the new hyper-sphere \( (r_n) \) will be close to the radius of previous hyper-sphere \( (r_1 r_2 \cdots r_{n-1}) \) as illustrated in Fig. 1(a). In the latter case, newly available data may be inside or outside of the hyper-sphere depending on its difference to other data and penalty parameter in the support vector based method \((C)\). If it stays outside of the hyper-sphere \((\xi_1 \text{ in (3) becomes positive})\), it is labeled as a class different from data within the hyper-sphere. In other words, a change point is detected. Note that the radius of the new hyper-sphere will not be so different from the radii of previous hyper-spheres, since newly available data is left out. Figure 1(b)
illustrates this scenario. On the other hand, if new data happens to be within the hyper-sphere ($\xi_i$ in (3) becomes zero) given there exists a change at current time, this indicates that the hyper-sphere has updated itself to represent the changed data. Thus, the radius of the hyper-sphere will either shrink or grow to represent the newly available data. Fortunately, the change can be detected by comparing the radius of the new hyper-sphere with the radii of the previous hyper-spheres as illustrated in Fig. 1(c) (increasing variance) and Fig. 1(d) (decreasing variance).

Remember that the hyper-sphere is a perfect space in a higher dimensional space, not necessarily in the original space. Each hyper-sphere calculated using data in different windows might be in different dimensional spaces due to the empirical nature of the support vector logic (based on the data used). Thus, the comparison of
radii of hyper-spheres in higher dimensional space will not be valuable in change
detection and the approximated radius in the original space will be used by com-
parison of radius values. Approximated radius term is calculated as the mean of
distance of support vectors to their center as in (9). The center of the support vectors
is the point located in middle of the support vectors and calculated as in (10).

\[ r = \frac{\sum_{i}^{n} \text{dis}(sv_i - c)}{n} \]  \hspace{1cm} (9)

\[ c = \frac{\sum_{i}^{n} sv_i}{n} \]  \hspace{1cm} (10)

\( r \): Approximated radius
\( sv_i \): \( i \)th support vector
\( c \): Center of the support vectors
\( \text{dis}(sv_i - c) \): Distance between support vector and center of the hyper-sphere
\( n \): Number of support vectors not left outside (0 < \( \alpha_i < C \))

Comparison of approximate radius values are quantified as radius ratio (\( h \)) as in (11). Radius ratio is the ratio of radius of hyper-sphere in time \( t \) to the average of radii of hyper-spheres obtained since the most recent change point.

\[ h_t = \frac{r_t}{\text{mean}(r_{yt-1})} \]  \hspace{1cm} (11)

\( r_t \): Approximated radius of hyper-sphere in time \( t \)
\( \text{mean}(r_{yt-1}) \): Average of previous approximated radius values
\( y_t \): Time of last change point before \( t \) or 1 if not any
\( w \): Window size

The presented method identifies the change point when the new data point is left
out of the hyper-sphere (mean change or variance increase), or when the radius ratio
is increased (mean change or variance increase) or decreased (decreased variance). In
the first case (data is left out of the hyper-sphere), the change is detected at that time
point. However, the change in the radius should be significant enough to raise an
alarm for change detection in the second and third cases (increased or decreased
radius). Thus, low and high thresholds (\( th_{\text{low}}, th_{\text{high}} \)) are employed for change
detection in the case of radius change. If the radius change is lower than the low
threshold (\( th_{\text{low}} \)) or higher than the high threshold (\( th_{\text{high}} \), the alarm for change
detection is raised. The algorithm is summarized in Table 1.

Note that when there is a change in mean or increase in variance, it may be
detected either by leaving the new data out of the hyper-sphere or measuring the
radius increase. This depends on the penalty value (\( C \)) in (3). When the penalty
value is high, new data will not be left out by extending the radius of the hyper-
sphere. When penalty is low, it will be easily left out of the hyper-sphere without
changing the radius of the hyper-sphere.
Three major difficulties were reported in Sec. 2: necessity of AR model fitting, necessity of change modeling in the training, and inefficiency in variance decrease. Note that no time series model (AR or ARIMA) is being used for model fitting in any part of the algorithm. Thus, the necessity of time series model fitting to obtain i.i.d. white noise is removed. In addition, neither have possible changes been modeled and nor used in training in the algorithm. Hence the algorithm targets any possible change. The efficiency of the algorithm in variance decrease is demonstrated in the next section (data sets 3, 4, 5 and 6), which discusses the experimental results of the presented method.

5. Experimental Results

In this section, the method is applied to six different data sets. First three data sets are obtained from papers in Refs. 11 and 25 in order to compare the presented method with methods CF, SC (versions of smartsifter),25 and GS.11 The fourth data set is obtained from Ref. 6 in order to compare our method with MLMB6 and CUSUM.15 The fifth dataset is in three-dimensional space and is used to evaluate the method in 2D space. The last data set is real data obtained from a patient from a hospital in Beth Israel Hospital in Boston.7,28

The data in the first three data sets are modeled in second-order AR model as in (12). Their sizes are 10,000. The change is made in $y_{\frac{1000}{10}}$th data point ($y = 1, 2, 3, \ldots, 9$). The effectiveness of the detection method is quantified by two parameters: average benefit and far (false alarm rate). Detection within 20 data points after the shift is awarded as an increase in benefit as in (13) and (14) and false detections are punished as an increase as far as in (15). The evaluation is based on early detection of the shift with minimum false alarms.25 Early detections lead to higher benefit. False alarm is the ratio of false alarms to the total number of alarms as in (15).

$$x_t = 0.6x_{t-1} - 0.5x_{t-2} + \varepsilon_t$$

(12)

$$\text{benefit}(y) = 1 - (t_y - t_y^*)/20$$

(13)
\( t_y \): Detection time of change \( y \)
\( t_y^* \): Real time of change point change \( y \)

\[
\text{average benefit} = \frac{\sum_{y=1}^{N(\text{Detection})} \text{benefit}(y)}{N(\text{Detection})}
\]

(14)

\[
\text{far} = \frac{N(\text{false changes})}{N(\text{detections})}
\]

(15)

\( N(\text{false changes}) \): Total number of falsely detected change points
\( N(\text{detections}) \): Total number of change point detections

In the first data set, the mean change is modeled by changing mean value by \( \Delta(y) \), which is fixed to 5 for this experiment (\( \Delta(y) = 5 \)), where \( y \) is the order of the change point. Figure 2 displays the data set #1, radius ratios (\( h \)), and change detections (\( -1 \) indicates a change, \( 1 \) indicates no change). As seen from the figure, the radius ratio increases at change points leading to immediate detection. Benefit value with SVPCD is 1, which is the highest possible, with 0 false alarm rate. Window size of 40 and \( C = 0.95 \) are used. Figure 3 displays benefit versus false alarm rate values for CF, SC, GS and SVCPD. Several values obtained by different window sizes are displayed in the figure (displayed as star). Top left corner of the figure gives the best
results with zero false alarms and immediate detection of the changes. As seen from the figure, our method highly outperforms the other methods in mean change.

In the second data set, the mean and variance change are modeled together. The mean is changed by $\Delta(y) = 10 - y$, where $y$ is the order of the change points $y = 1, 2, \ldots, 9$. The standard deviation of noise term ($\varepsilon$) is defined as $0.1/(0.01 + (10,000 - t)/10,000)$. Figure 4 displays the data set #2, radius ratios ($h$), and change detections ($-1$ indicates a change, 0 indicates no change). The penalty value ($C$) is defined as 0.95. As seen from the figure, the radius ratio increases in change points. Figure 5 displays benefit versus false alarm rate values for CF, SC, GS and SVCPD. Results of four different window sizes (10, 14, 18, 22) are plotted. When the window size decreases, “false alarm rate” increases. Increasing window size will increase the computational time. As seen from the figure, SVCPD highly outperforms the other methods regarding change in mean and increase in variance.

In the third dataset, the variance change is modeled by decreasing and increasing it. The variance changes from 1.0 to 9.0, when $y$ is odd and from 9.0 to 1.0 when $y$ is even. Figure 6 displays the data set #3, radius ratios ($h$), and change detections ($-1$ indicates a change, 1 indicates no change). As seen from the figure, the radius ratio increases when the variance increases, decreases when the variance decreases. SVCPD is able to detect all change points. Note that all the variance decrease points (2001, 4001, 6001, 8001) were detected. SVCPD is able to detect not only changes in mean and increase in variance, but also a decrease in variance, which is the main

Fig. 3. Benefit versus FAR in data set #1 using CF, SC, GS and SVCPD.
Dataset # 2

Fig. 4. Data set # 2, radius ratios and change detections.

Dataset #2

Fig. 5. Benefit versus FAR in data set # 2 using CF, SC, GS and SVCPD.
The weakness of the other methods in Refs. 6, 11, 15 and 25. The penalty value \( C \) is defined as 0.95. Figure 7 displays benefit versus false alarm rate values for CF, SC, GS and SVCPD. As seen from the figure, SVCPD highly outperforms the other methods. The results of different window sizes are plotted in the figure.

The fourth data set is used for comparison of the present method with MLMB and CUSUM. The data representing six segments with five change points (3 variance decreases, 2 variance increases) is plotted in Fig. 8. Note that all variance decrease points (274, 666, and 1010) have been detected as well as variance increase points. Figure 9 plots the benefit versus far values for the three methods. As displayed in Table 2, the data set consists of six segments. Number of observations, standard deviation, deviation ratio (current standard deviation/previous standard deviation), and changing time points in each segment are given in second through fifth columns. Deviation ratio less than 1 implies variance decrease; greater than 1 implies variance increase. When the deviation ratio gets close to 1, detection becomes harder. The last three columns give the detection points for the given segment with SVCPD, MLMB and CUSUM. As seen from the table, SVPCD gives better results than MLMB and CUSUM, also displayed in Fig. 9. The benefit obtained in this data set (around 0.65) is worse than the benefit values obtained in the first three data sets (around 1). The reason for this drop in benefit lies in the level of change in the change point. In this data set, the variance change is smaller compared to the previous data sets leading to late detection and less benefit. The penalty value \( C \) is defined as 0.95 for this data set.
Fig. 7. Benefit versus FAR in data set #3 using CF, SC, GS and SVCPD.

Fig. 8. Data set #4, radius ratios and change detections.
Many methods presented in the literature can only be applied to one-dimensional data. For multidimensional data, these methods should be applied to each dimension distinctly. SVCPD can be applied directly to multidimensional data. The fifth dataset presents the ability of SVPCD for multidimensional space (2D space as displayed in Fig. 10, in this example).

The $x$-axis represents the time information. Data starts with high variance and the variance decreases in time 201 and increases again in time 401. Both variance changes (decrease and increase) are detected by SVCPD as displayed in Fig. 11. Table 3 shows data properties and detection results. Three segments with number of observations, standard deviations, deviation ratios, changing time points and detection times with SVPCD are given respectively in the columns in Table 3.

![Fig. 9. Benefit versus FAR for data set # 4.](image)

Table 2. Summary of results obtained with SVCPD on data set # 4.

<table>
<thead>
<tr>
<th>Segment No.</th>
<th>Number of Obs.</th>
<th>Standard Deviation</th>
<th>Deviation Ratio</th>
<th>Real Change Point</th>
<th>Found</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SVCPD</td>
</tr>
<tr>
<td>1</td>
<td>274</td>
<td>2.45</td>
<td></td>
<td>275</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>187</td>
<td>1.17</td>
<td>0.48</td>
<td>462</td>
<td>285:288</td>
</tr>
<tr>
<td>3</td>
<td>205</td>
<td>3.06</td>
<td>2.61</td>
<td>667</td>
<td>470</td>
</tr>
<tr>
<td>4</td>
<td>216</td>
<td>2.30</td>
<td>0.75</td>
<td>883</td>
<td>668</td>
</tr>
<tr>
<td>5</td>
<td>128</td>
<td>2.65</td>
<td>1.15</td>
<td>1011</td>
<td>1018:1020</td>
</tr>
<tr>
<td>6</td>
<td>201</td>
<td>1.21</td>
<td>0.46</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fig. 10. Data set # 5.

Fig. 11. Radius ratio and detected change points in data set # 5.
The last dataset is recorded from a patient in the sleep laboratory of the Beth Israel Hospital in Boston, Massachusetts and publicly available time series data in Refs. 7 and 28. Data consist of heart rate, chest volume and blood oxygen concentration. During the data collection, one person continuously observes the patient and notes if the state of the patient is being awake, asleep, or intermediate states. The real problem is identifying the state of the patient by analyzing the three parameters. Since we are only interested in the change detection, only chest volume is used in our method. Chest volume has been categorized as small, big and medium. The problem we are interested in is the identification of change points between chest volume categories. Since the data is too long, part of the time series (3600 data points) is selected. There are 38 change points in the selected part of the data set as illustrated in Fig. 12. Table 4 gives the details of the change points.

In the application of SVCPD, we used window size \( w = 5, 15, 25, 35 \) and 45. For low \( (t_{\text{low}}) \) and high \( (t_{\text{high}}) \) thresholds, 42 combinations (6 high: 1.13, 1.23, 1.33, 1.43, 1.53 and 1.63; 7 low: 0.23, 0.33, 0.43, 0.53, 0.63, 0.73 and 0.83) are used. In order to compare the effects of parameters, scatter plots of mean time between false alarms (MTBFA) and average detection delay have been used in the literature.1,30

Table 3. Summary of results obtained with SVCPD on data set # 5.

<table>
<thead>
<tr>
<th>Segment No.</th>
<th>Number of Obs.</th>
<th>Standard Deviation</th>
<th>Deviation Ratio</th>
<th>Real Change Point</th>
<th>Found SVCPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>5</td>
<td></td>
<td>201</td>
<td>215:232</td>
</tr>
<tr>
<td>2</td>
<td>200</td>
<td>1</td>
<td>0.2</td>
<td>201</td>
<td>201:232</td>
</tr>
<tr>
<td>3</td>
<td>200</td>
<td>3</td>
<td>3</td>
<td>401</td>
<td>403:421</td>
</tr>
</tbody>
</table>

Fig. 12. Real data set obtained from a patient.
Table 4. Details of data set #6 (L: large, M: medium, S: small).

<table>
<thead>
<tr>
<th>Segment No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Point</td>
<td>91</td>
<td>138</td>
<td>166</td>
<td>186</td>
<td>222</td>
<td>252</td>
<td>412</td>
<td>430</td>
<td>455</td>
<td>571</td>
<td>580</td>
<td>659</td>
<td>669</td>
<td>831</td>
<td>845</td>
<td>934</td>
<td>945</td>
<td>1020</td>
<td>1036</td>
</tr>
<tr>
<td>Number of Data</td>
<td>90</td>
<td>47</td>
<td>28</td>
<td>20</td>
<td>36</td>
<td>30</td>
<td>160</td>
<td>18</td>
<td>25</td>
<td>116</td>
<td>9</td>
<td>79</td>
<td>10</td>
<td>162</td>
<td>14</td>
<td>89</td>
<td>11</td>
<td>75</td>
<td>16</td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>L</td>
<td>M</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>M</td>
<td>S</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>S</td>
<td>L</td>
<td>S</td>
</tr>
<tr>
<td>Segment No.</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>27</td>
<td>28</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Point</td>
<td>1078</td>
<td>1102</td>
<td>1135</td>
<td>1167</td>
<td>1210</td>
<td>1236</td>
<td>1280</td>
<td>1300</td>
<td>1331</td>
<td>1371</td>
<td>1407</td>
<td>1444</td>
<td>1489</td>
<td>1517</td>
<td>1559</td>
<td>1590</td>
<td>1639</td>
<td>1673</td>
<td>1797</td>
</tr>
<tr>
<td>Number of Data</td>
<td>42</td>
<td>24</td>
<td>33</td>
<td>32</td>
<td>43</td>
<td>26</td>
<td>44</td>
<td>20</td>
<td>31</td>
<td>40</td>
<td>36</td>
<td>37</td>
<td>45</td>
<td>28</td>
<td>42</td>
<td>31</td>
<td>49</td>
<td>34</td>
<td>124</td>
</tr>
<tr>
<td><strong>Mode</strong></td>
<td>L</td>
<td>S</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>M</td>
</tr>
</tbody>
</table>
Fig. 13. Mean time between false alarms versus average detection delay.

Fig. 14. Mean time between false alarms versus number of correctly detected change points.
Figure 13 displays MTBFA and average detection delay with different window size and different low and high threshold values. The best result would be located in the bottom right corner of the figure with smallest detection delay and highest MTBFA. As seen from the figure, when the window size is too low (say five), all results (obtained with different thresholds) are located around the bottom left corner with low detection delay and low MTBFA. This is true because the number of data points is not enough to learn the characteristics of chest volume causing frequent alarms. When the window size increases, the mean time between false alarms increases (false alarms decreases), however the detection delay increases.

Since there are a lot of change detection points in the data set (38 change points), the number of correctly detected changes can be employed instead of detection delay as in Fig. 14. In this figure, the best results are located in the top right corner of the figure with large number of correctly detected changes and low false alarms. Figure 15 displays the radius ratio and detected changes for threshold values of 1.43 and 0.33.

6. Conclusion
A support vector based change point detection method is presented in this paper. Most methods in the literature have difficulty in detecting change that occurs as variance decreases. In addition, most methods apply a AR model to the time series data in order to obtain identically distributed data with normal distribution, since
they make assumption about data distribution and independence of data. The support vector based change point detection method presented in this paper targets finding changes in mean and/or variance without making any assumption about the data distribution and independency. The presented method is compared with other methods and the results show that it outperforms other methods.

References

Fatih Camci is an Assistant Professor in the Department of Computer Engineering at Fatih University, Istanbul Turkey. He worked as senior project engineer at Impact Technologies in Rochester, NY for two years in the development of diagnostic/prognostic methods for military systems. He received his Ph.D. in Industrial Engineering from Wayne State University. He received his B.S. degree in computer engineering from Istanbul University (Turkey) and M.S. degree in computer engineering from Fatih University (Turkey).

His areas of expertise include engineering intelligent systems, optimization in engineering systems, and condition based maintenance.