Frontiers of Information Technology & Electronic Engineering
www.zju.edu.cn/jzus; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Symbolic representation based on trend features for knowledge discovery in long time series*

Hong YIN†[1,3], Shu-qiang YANG[1], Xiao-qian ZHU[1], Shao-dong MA[2], Lu-min ZHANG[1]

([1]*College of Computer, National University of Defense Technology, Changsha 410073, China*)
([2]*School of Engineering, University of Hull, Cottingham Road HU6 7RX, UK*)
([3]*Xiangyang School for NCOs, Xiangyang 441118, China*)
†E-mail: yinhonggfkd@aliyun.com

Received Nov. 2, 2014; Revision accepted May 27, 2015; Crosschecked Aug. 10, 2015

Abstract: The symbolic representation of time series has attracted much research interest recently. The high dimensionality typical of the data is challenging, especially as the time series becomes longer. The wide distribution of sensors collecting more and more data exacerbates the problem. Representing a time series effectively is an essential task for decision-making activities such as classification, prediction, and knowledge discovery. In this paper, we propose a new symbolic representation method for long time series based on trend features, called trend feature symbolic approximation (TFSA). The method uses a two-step mechanism to segment long time series rapidly. Unlike some previous symbolic methods, it focuses on retaining most of the trend features and patterns of the original series. A time series is represented by trend symbols, which are also suitable for use in knowledge discovery, such as association rules mining. TFSA provides the lower bounding guarantee. Experimental results show that, compared with some previous methods, it not only has better segmentation efficiency and classification accuracy, but also is applicable for use in knowledge discovery from time series.

Key words: Long time series, Segmentation, Trend features, Symbolic, Knowledge discovery
doi:10.1631/FITEE.1400376 **Document code:** A **CLC number:** TP311

1 Introduction

Advances in data collection and storage technologies achieved in recent decades have massively broadened the variety of sensory data. Accumulated data from a large-scale sensor network can easily exceed a storage capacity of thousands of gigabytes. One purpose of collecting the data is to reveal useful hidden information and connections. Undeniably, the analysis and mining of such data should be automated, particularly, in the emerging and dynamic applications of patient health evaluation, or smart city and power grid management. Each sensor node collects physically measured data or multimedia information, which can be interpreted and stored in the form of time series. Techniques such as knowledge discovery can be used efficiently in the data management and classification of time series to aid relevant decision-making. Knowledge discovery refers to the mining of previously unknown rules that can be understood and interpreted, and that can be automatically validated and evaluated (Guimarães and Ultsch, 1999).

The explosion of interest in knowledge discovery from time series was sparked by a requirement to extract sequential patterns from a database of transactions (Agrawal and Srikant, 1995). This was further extended to discover frequent episodes and episode rules (Mannila and Toivonen, 1996). The application of unsupervised neural networks to the mining rules

* Project supported by the National High-Tech R&D Program (863) of China (Nos. 2012AA012600, 2011AA010702, 2012AA01A401, and 2012AA01A402), the National Natural Science Foundation of China (No. 60933005), and the National Science and Technology of China (No. 2012BAH38B04)
ORCID: Hong YIN, http://orcid.org/0000-0002-0682-6781

between temporal patterns generated temporal grammatical rules for a symbolic knowledge representation (Guimarães *et al.*, 2001). Villafane *et al.* (2000) proposed a mining technique to discover containment relationships in a series of interval events derived from a numerical time series by a quantization step. Mining knowledge from time series has been applied to distinguish underlying temporal processes or anomalous behaviors, and to predict more intelligently through the use of historical data. However, most time series merely explicate the potential information of using themselves, and many more underlying rules are derived by mining algorithms. Another noticeable feature is that most sensor data exist as collections of consecutive values varying continuously in time. As a prerequisite, the consecutive data need to be discretized for data mining algorithms. The symbolic representation of time series is a compatible method, which has proved useful in facilitating the manipulation of discretized data in many areas, such as trend analysis of meteorological data (Mellit *et al.*, 2013), fault diagnosis of remote sensing data from space (Sarkar *et al.*, 2013), consistency checking in medical data (Vullings *et al.*, 1997), and anomaly detection in GPS positioning (Bu *et al.*, 2009).

The basic concept of time series symbolic representation is to convert the numerical form of a time series into a sequence of discrete symbols according to designated mapping rules. Many researchers have proposed high-level representations of time series, including discrete Fourier transform (DFT) (Faloutsos *et al.*, 1994), discrete wavelet transform (DWT) (Chan and Fu, 1999), piecewise aggregate approximation (PAA) (Keogh *et al.*, 2001), and singular value decomposition (SVD) (Korn *et al.*, 1997). The transformed sequences are characterized as discrete, non-real numbers with reduced dimensionality. So, a reasonable symbolic method can improve the efficiency of time series data mining. In this paper, we propose a generalized method for the entire process of knowledge discovery from long time series. First, the time series is partitioned using a two-step segmentation mechanism resulting in pieces whose intervals can be unequal. Then the trend feature symbolic approximation (TFSA) is used to symbolize these intervals. Finally, an apriori-based algorithm is used for discovering association rules from the symbol itemsets. The original contributions of this paper are an improvement in the segmentation efficiency of long time series, and the TFSA method which focuses on preserving the trend features of the original time series and interprets the rules obtained from mining time series.

2 Problem statement

The high dimensionality of time series is responsible mainly for increasing the access time and computation load of data mining algorithms. Also, the meanings of terms such as 'similar to' and 'cluster forming' are not definitive in high dimensional space. This complicates the application of knowledge discovery techniques to raw time series. To discover knowledge from long time series, there are two problems that need to be addressed: segmentation efficiency and validity of knowledge (or rules).

Time series segmentation is essential for knowledge discovery from time series. Existing segmentation algorithms can be divided into three categories:

1. Constrained length: In this type, the number of segments after division is pre-defined. For example, to obtain a fixed segmentation number N, the time series can be divided into equal widths without considering other conditions. This segmentation method is used in many data representation methods, such as PAA, because of its simplicity.

2. Given fitting errors: This type of method splits time series by controlling the segmentation error to find the appropriate segmentation points. Common methods include: sliding window (SW), top-down (TD), and bottom-up (BU).

3. Segmentation based on key points: Splitting time series by some key points, such as local extreme points, edge points, or turning points, can avoid missing important information from the original time series. Existing methods include important points (IPs) (Phetking *et al.*, 2008), perceptually important points (PIPs) (Yeh *et al.*, 2004), and turning points (TPs) (Bao and Yang, 2008).

Most high-level segmentation algorithms have to consider the challenges caused by the length of time series. For example, the time complexity of TD is $O(N^2)$ (N is the length of the time series). If N is extraordinarily large (e.g., $N \geq 10^7$), it will require a considerable amount of computation. In this paper, a two-step segmentation mechanism is proposed to reduce the computational complexity.

To obtain the desired rules, a symbolic form is usually preferred for knowledge discovery from time series. Though many discretization methods convert numeric time series to symbols, they ignore the trend features of the original series. Hence, symbols produced by the general discretization methods are not suitable using efficient mining rules that can be expressed in a natural language and used for predicting the future behavior of the time series. For example, symbolic aggregate approximation (SAX) (Lin *et al.*, 2003) is a prevailing symbolic method because of its simplicity and high computational efficiency. SAX is also reputed to have reliable performance when used for the data mining tasks of clustering, classification, indexing, and anomaly detection. However, these methods are unsuitable for use in knowledge discovery processes, such as association rule mining. For example, SAX uses the mean value of subsequences to represent the time series intervals, thereby ignoring many of the trend features. To improve knowledge use and interpretability, it is very important that the symbolic results of intervals should describe these trend features. In this paper, trend features are used to symbolize time series for two purposes. First, trend features are an important characteristic of a time series. In some applications, the way that time series values vary is considered to be very important, because it enables useful conclusions to be drawn (Kontaki *et al.*, 2005). For example, in a satellite fault monitoring system it is important to know which telemetry parameters show an increasing trend and which show a decreasing trend to avoid serious failures. Second, trend-based representation of time series is more closely aligned to human intuition (Kontaki *et al.*, 2008). Results from symbolic representation are very useful for making quick and valuable decisions in practical applications, such as medical diagnosis or financial analysis. The TFSA method, based on the trend features, is suitable for meaningful discretization of numeric time series. TFSA is focused on preserving the trend features, which are used in symbolizing intervals after segmentation. This approach improves the validity of the rules obtained. Note that monotonically increasing or decreasing time series is not considered in this paper because of their simplicity.

3 Two-step segmentation mechanism

For time series analysis, the sequence length is an important factor. The longer the time series, the more slowly the algorithm runs due to its computational complexity. When the length of time series N is huge, the computational burden becomes serious. The two-step segmentation mechanism first splits a long time series into k shorter segments, reducing the time complexity to $O(N^2/k^2)$. So, shorter subsequences after division can be segmented in the second step.

3.1 First step: searching for the key points of time series

The purpose of time series segmentation is to divide the original series into a set of independent subsequences. After division, a high-level representation of the processed series can be derived, and further mining techniques such as indexing, clustering, classification, and the association rule, become more efficient. To clarify the following presentation, the following definitions of time series properties are given. The definitions of 'time series' and 'subsequence' are based on those in Esling and Agon (2012).

Definition 1 (Time series) A time series $T=t_1, t_2, \ldots, t_m$ is a sequence of real-valued data collected at regular intervals over a period of time, where m is the length of the time series.

Definition 2 (Subsequence) Given a time series T of length m, a subsequence Q of T is a sampling of length n ($n \leq m$) with contiguous positions from T, that is, $Q=t_p, t_{p+1}, \ldots, t_{p+n-1}$, $1 \leq p \leq m-n+1$.

Definition 3 (k-segmentation of a time series) Time series T is divided into k subsequences $T=Q_1Q_2\ldots Q_k$, where Q_i ($i=1, 2, \ldots, k$) is not null. A group of segment boundaries is defined as $u_0, u_1, \ldots, u_k$, $u_i \in \{1, 2, \ldots, m\}$, $1=u_0<u_1<\ldots<u_k=m$. So, $Q_1=T(u_0, u_1)$, $Q_2=T(u_1+1, u_2)$, $Q_3=T(u_2+1, u_3)$, …, $Q_k=T(u_{k-1}+1, u_k)$.

It is very important to search for key points of time series where the series pattern changes. The important key points include extreme points, local extreme points, important points, and turning points. Many methods have been proposed for key point searching, but not all of them consider the global extreme points or global turning points due to the complexity of the algorithms. The significance of the key points, from a global perspective, especially for long time series, will be greatly reduced if they are not considered globally. In this paper we propose a method to find global key points, called CUSUM-based turning points (CBTP), based on cumulative sum control chart (Yeh *et al.*, 2004). This approach fully considers the global series change information, and thus is useful for identifying the turning point from one pattern to the next changed pattern. Accumulating the changes in the series and amplifying small shifts in the process of series change, is a particularly efficient way to find abrupt changes. Because of its simple computation, the algorithm is more efficient. The searching algorithm of CBTP is given in Algorithm 1.

```
Algorithm 1 CUSUM-based turning point searching
Input: time series $T=t_1, t_2, \ldots, t_m$.
Output: turning points $u_j$, $u_j\in\{1, 2, \ldots, m\}$.
Calculate the mean value of time series: $\overline{t}=\sum_{i=1}^{m} t_i/m$.
Set the initial CUSUM: $s_0=0$.
Calculate CUSUM of each points: $s_i=s_{i-1}+(t_i-\overline{t})$, $i=1, 2, \ldots, m$.
Set $s_{\max}=\max\{|s_i|\}$, $i=1, 2, \ldots, m$, and the output point is $u_j=i$.
```

A numerical example of a time series (Fig. 1) is used to illustrate the CBTP algorithm. First, the mean value of the time series is calculated, $\overline{t}=\sum_{i=1}^{19} t_i/m=11.4$. Then the cumulative sum s_i of each point is computed iteratively, which results in a dataset of s_i={−0.4, 1.2, 1.8, 2.4, 4.0, 7.6, 11.2, 12.8, 17.4, 17.0, 15.6, 14.2, 13.8, 9.4, 7.0, 3.6, 4.2, 1.8, 0.4}. The maximum value of s_i is 17.4 when $u_j=i=9$. Hence, t_9 is one of the turning points of the time series. The results are shown in Fig. 2.

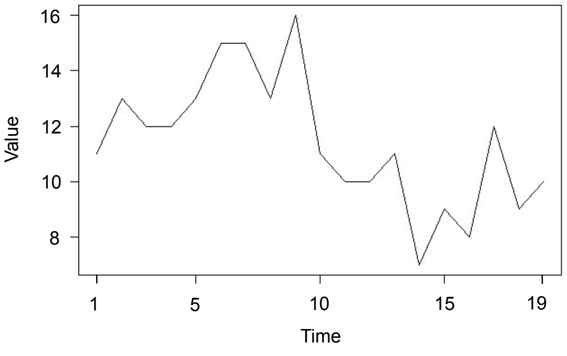

Fig. 1 An example time series

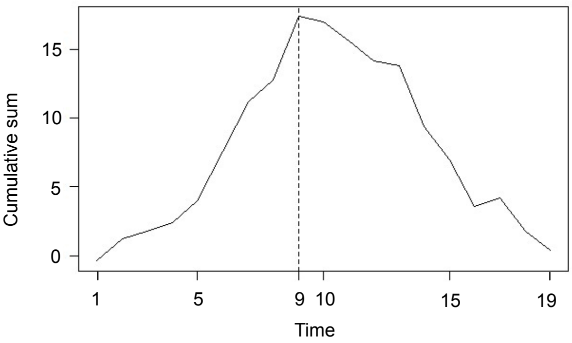

Fig. 2 Cumulative sum of every point

The CBTP algorithm does not need any predefined parameters, such as fitting error, which will change the segmentation results due to the different values. So, the performance of CBTP is stable. The above process can be repeated recursively to find other turning points of subsequences obtained from former segmentation until the pre-defined k segmentations are finished.

3.2 Second step: segmentation of shorter series

In the first step, a long time series is split into several shorter series to which the complicated segmentation algorithms can be applied without reducing efficiency greatly. Using the CBTP algorithm, the turning points that divide the original series into segments with different patterns can be obtained. At these points, a significant change has occurred in the latter part compared with the former. However, there is a drawback in that the recursion will produce a lot of repeated calculations if only the CBTP algorithm is used. So, after the first step of segmentation using CBTP, an adaptive segmentation algorithm based on the sliding window is attempted. Not only will the efficiency of segmentation be improved due to the shorter length, but also the parallel mechanism can be considered on these segmentations.

Definition 4 (Number of initial segmentations) Division of the time series into *k* pieces in the first step is made according to a pre-defined number of segments *k*.

Definition 5 (Sliding window) Given a time series *T* of length *m*, a subsequence of length *n* can be obtained using a window of size *n* sliding on the time series.

So, the process of the two-step segmentation is as follows: first, according to the initial segmentation *k*, the CBTP algorithm is used to segment the time series into *k* subsequences. Special consideration is given for the divided subsequences to avoid those of extreme length (too long or too short). Second, further segmentation can be based on the adaptive segmentation algorithm (Lavielle and Teyssière, 2006). To generate an approximation based on trend features, least squares regression (LSR) is used to fit a straight line through segmentation. The least squares method assumes that the best-fit line is the line where the sum *S* of squared residuals is a minimum:

$$S=\sum_{i=1}^{n}[y_i-f(x_i)]^2.$$

The general idea of the algorithm is as follows: Given a window size (*fit*) and an angle tolerance (*angle.tol*), the segmentation algorithm starts by finding the slope of the first *fit* points of the series through LSR. The window slides over one point and the new slope is computed for the points included by the new window. Comparing the new slope with the old one, if the change in slope exceeds *angle.tol*, a change-point is recorded as the rightmost point of the previous window. The routine then picks up again starting at the point just to the right of the change-point. If the change of slope does not exceed *angle.tol*, then the old slope will be updated. The algorithm continues until the sliding window reaches the edge of the time series. The adaptive segmentation algorithm is given in Algorithm 2.

```
Algorithm 2    Adaptive segmentation
Input: subsequence $Q_i=q_1q_2\ldots q_j$.
Output: change-points $u_i$ ($1\le i\le j$).
Step 1: set the size of sliding window $fit=l$.
Step 2: compute the slope of the first $fit$ points of
    series $k_1$.
Step 3: slide over one point and compute the new
    slope $k_2$.
Step 4: if $|k_1-k_2|>angle.tol$, a change-point $u_i=q_{l+1}$ and
    $k_1$ are recorded, the window slides to the point $q_{l+1}$,
    and the routine goes to step 2.
Step 5: if $|k_1-k_2|<angle.tol$, $k=(k_1+k_2)/2$, the routine
    goes to step 3.
Step 6: continue until the sliding window reaches the
    edge of series.
```

The adaptive segmentation algorithm not only can be used to find the key points of a time series, but also is very suitable for recognizing periodic features. This is important for time series with periodicity, such as remote sensing space data or medical data. Taking the time series in Fig. 3 as an example, the data are composed of several series following a normal distribution.

Using Algorithm 2 to segment the series gives the results as shown in Fig. 4. From the results, the key points for recognizing the periodic features are obtained, and the data are shown to be composed of

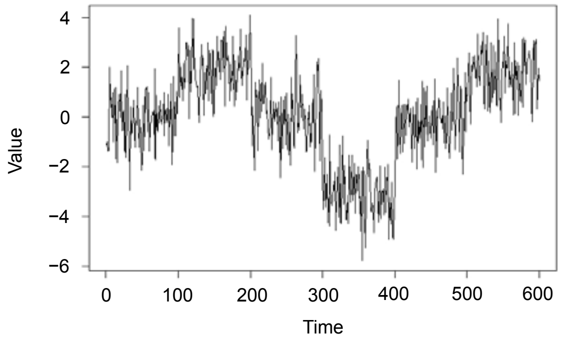

Fig. 3 Example data composed of a series obeying a normal distribution

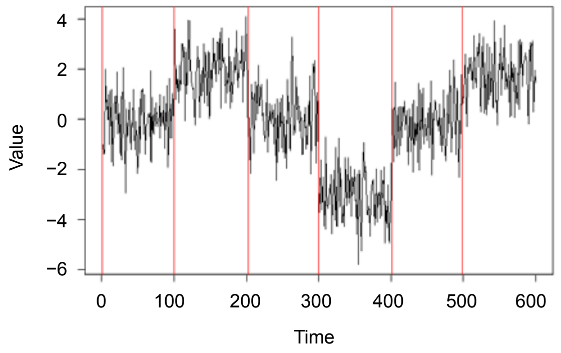

Fig. 4 The periodic features obtained using the adaptive segmentation algorithm

six series obeying a normal distribution with different mean values.

4 Symbolic representation based on trend features

In practice, many time series are not stationary or monotonous. Therefore, most time series will show different 'trends' as time progresses and these trends are important features of a time series. A trend should have a direction. That is, it should have a higher or lower value at the end of the series, so that it will seem generally to increase or decrease over time. Trend-based approximations have been studied extensively in the last decade. For example, Kontaki *et al.* (2005) used piecewise linear approximation (PLA) to transform a time series into a vector of symbols (using U to indicate an upward trend and D for a downward trend). In this paper, three kinds of trend (Fig. 5) are considered: increasing, decreasing, and stationary. Other trends, like fast increasing and drastic decreasing, are supplemented by the slope coefficient which is used to measure the magnitude of a trend change.

Definition 6 (Increasing trend) Given a time series T of length m, if $\{t_i<t_j|i<j<m\}$ for all i, and j is sufficiently large, then T has an eventually increasing trend.

Definition 7 (Decreasing trend) Given a time series T of length m, if $\{t_i>t_j|i<j<m\}$ for all i, and j is

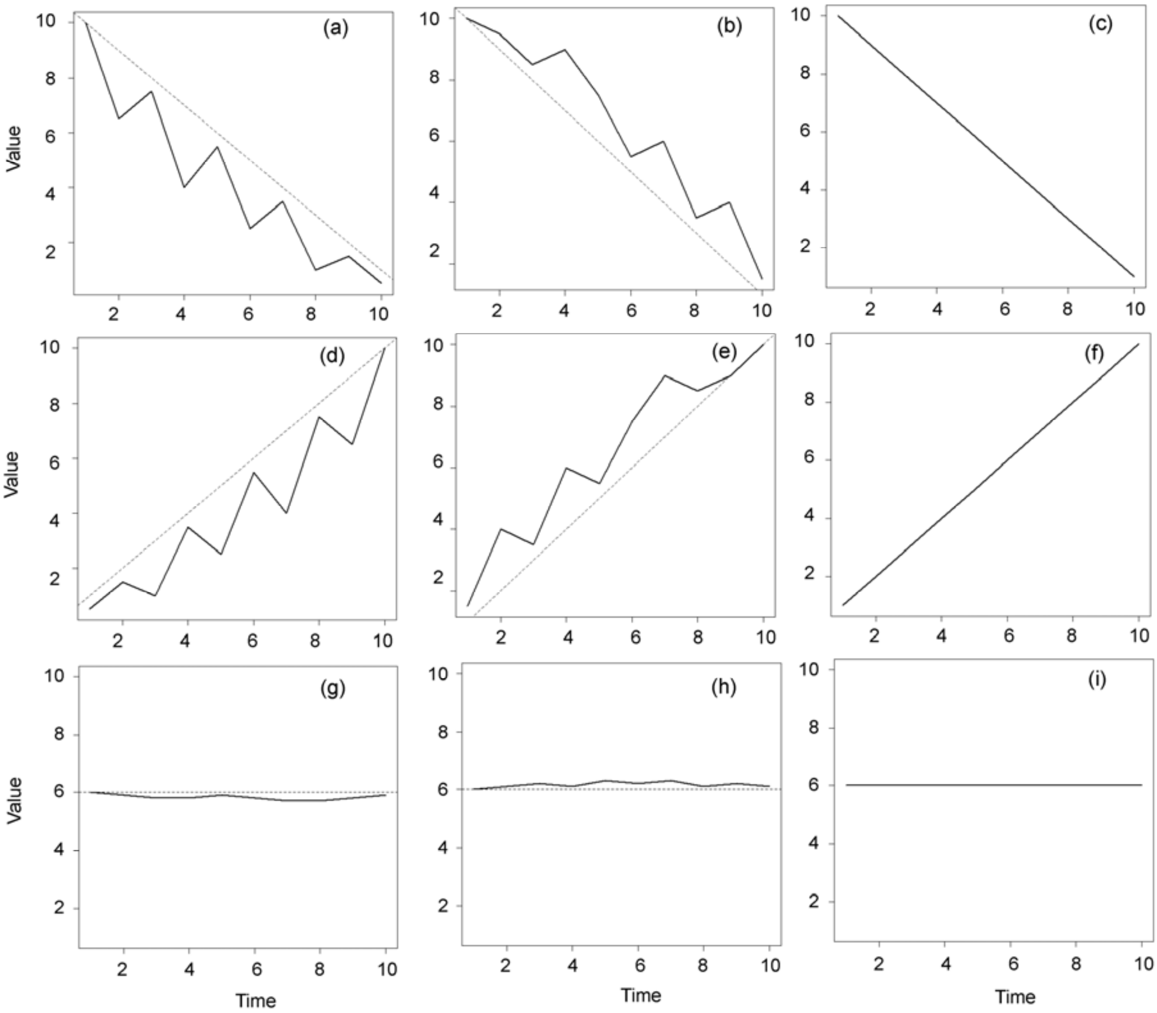

Fig. 5 The basic trends of the time series

(a) Concavely decreasing; (b) Convexly decreasing; (c) Linearly decreasing; (d) Concavely increasing; (e) Convexly increasing; (f) Linearly increasing; (g) Concavely stationary; (h) Convexly stationary; (i) Stationary sufficiently large, then T has an eventually decreasing trend.

Definition 8 (Stationary trend) Given a time series T of length m and a pre-defined minimum θ, if $\Delta t=|t_i-t_j|<\theta$, $i, j=1, 2, \ldots, m$ for all i, j, then T has a stationary trend.

Definition 9 (Increasing local trend) Let a, b be integers such that $0\le a\le b\le m$. If $t_i<t_j$ for all $a\le i\le j\le b$, then T has an increasing local trend on $[a, b]$.

Other local trends could be defined according to Definition 9, but here we omit them for brevity. Fig. 5 shows the basic trends that will be used in the following symbolic representation, and a symbolic representation using trends after segmentation is illustrated in Fig. 6.

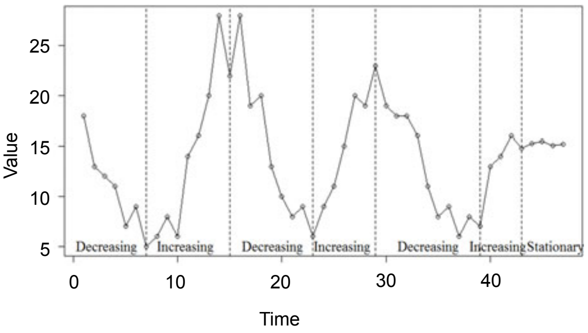

Fig. 6 Symbolic representation using trends after segmentation

4.1 Symbolization of intervals

After the two-step segmentation of the time series, the intervals obtained become imbalanced in length but the key points are retained.

The next steps involve extracting trend features from each of the intervals and representing the intervals by a set of symbols. In this paper, the trend features of interval series include: trend, slope, and last point. To symbolize the trends, symbols '01', '10', and '00' (or '11') are used to represent increasing, decreasing, and stationary trends, respectively. The slope is computed using LSR to fit the interval. Given an interval series S of length n, it is a finite set $\{(v_i, t_i)\}$, $0\le i\le n$, which is composed of a value v_i and a timestamp t_i. The slope a can be computed as

$$a=\left(n\sum_{i=1}^{n}v_i t_i-\sum_{i=1}^{n}v_i\cdot\sum_{i=1}^{n}t_i\right)\Bigg/\left(n\sum_{i=1}^{n}v_i^2-2\sum_{i=1}^{n}v_i\right). \quad (1)$$

The last point b of the intervals is an important feature, which not only represents the key points obtained from segmentation, but also can ensure the unique mapping between the symbolic results and the intervals.

The TFSA method provides specific symbolic representations of intervals (Table 1). A given example will illustrate the operation of our approach, and the original time series used for experiment is shown in Fig. 7. The long time series is not considered in the example for the conciseness of illustration.

Numerous studies have appreciated the importance of standardizing the time series before clustering, classification, and comparison of the similarity. Accordingly, our method normalizes the time series before symbolization into a standard sequence with zero mean and standard deviation. Using the two-step segmentation method to split the time series gives the results shown in Fig. 8.

In the process of segmentation, the slope a of the interval and the segmentation points b are retained. So, according to the symbols defined in Table 1, the symbolic results are: $Q_1=01_{-0.02}^{1.28}$, $k_1=1.28$, $T(u_1)=-0.02$, $Q_2=10_{-0.59}^{-1.29}$, $k_2=-1.29$, $T(u_2)=-0.59$, $Q_3=10_{1.75}^{5.32}$, $k_3=5.32$, $T(u_3)=1.75$, and so on. So, the symbolic representation of the time series is:

$$\begin{aligned}&Q_1Q_2Q_3Q_4Q_5Q_6Q_7Q_8\\&=01_{-0.02}^{1.28}10_{-0.59}^{-1.29}01_{1.75}^{5.32}11_{1.75}^{0}10_{0.64}^{-2.02}00_{0.64}^{0}10_{-1.25}^{2.15}01_{-0.99}^{0.26}.\end{aligned}$$

From the results of symbolization, it is easy to see that the time series initially tends to move upwards, and then drops downwards to −0.59. Then there is a drastic increase to 1.75. After a flat section, the time series declines until another flat region is

Table 1 Trend symbols from trend feature symbolic approximation

Symbol	Meaning
01_b^a	An upward trend interval
10_b^a	A downward trend interval
$00_b^0, 11_b^0$	A flat trend interval, using 11_b^0 to represent stationary after 01_b^a, and 00_b^0 to represent stationary after 10_b^a (the slope is zero)

a: degree of change, the value of which is the slope of the interval; b: the last point value of the interval after standardization reached. The descent of the time series continues before reaching a turning point at −0.99, followed by a slow rise to the finish. The trends are characterized and symbolized using the TFSA method (Fig. 9).

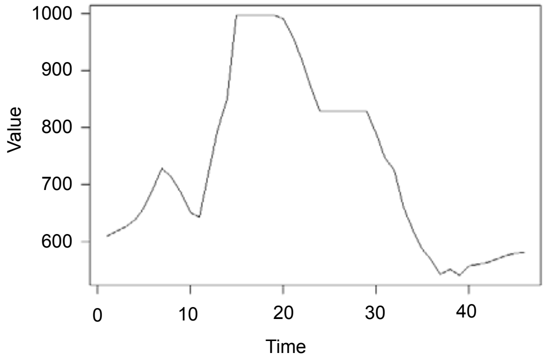

Fig. 7 The original time series

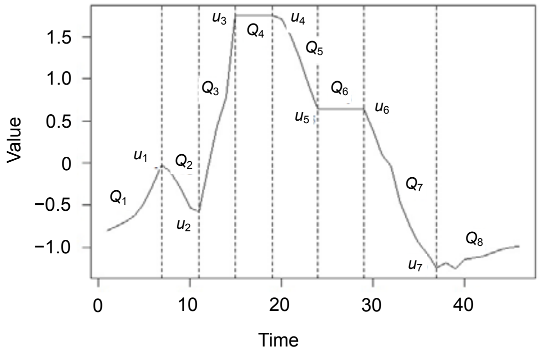

Fig. 8 Segmentation of a time series using two-step segmentation

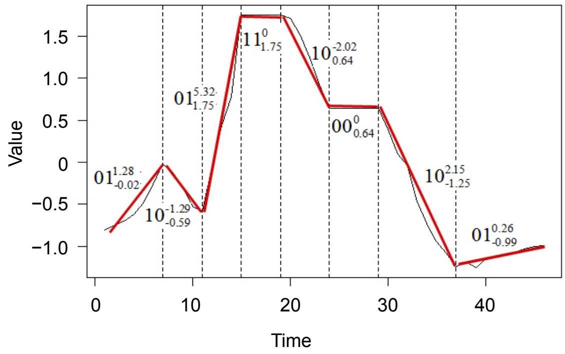

Fig. 9 Representation of trend features using trend feature symbolic approximation

4.2 Lower bounding guarantee

All symbolization methods need to be evaluated based on how closely the approximated symbol can represent the features of the original series. An important conclusion is drawn to ensure no false dismissals in the distance measure between the symbolic string and the true distance (Faloutsos *et al.*, 1994) The following condition must be satisfied:

$$D_{\text{symbolic}}(Q, C) \le D_{\text{true}}(A, B), \tag{2}$$

where A, B are the original time series measured by the true distance D_{true}, and Q, C are symbolic sequences of A, B measured by D_{symbolic}, respectively. This theory is also known as the lower bounding or the contractive property. The distance after symbolization should not exceed the true distance. In this study, the true distance was measured using Euclidean distance:

$$D_{\text{true}}(A,B) = D_{\text{Euclidean}}(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}. \tag{3}$$

According to the symbolic representation TFSA, $D_{\text{symbolic}}(Q, C)$ is defined by

$$\begin{aligned} & D_{\text{symbolic}}(Q,C) = D_{\text{TFSA}}(Q,C) \\ & = \sqrt{\frac{n}{w}\sum_{i=1}^{w} T_i \cdot (\text{qb}_i - \text{cb}_i)^2 \cdot \frac{\|\text{qa}_i| - |\text{ca}_i\|}{\max(|\text{qa}_i|, |\text{ca}_i|)}}, \end{aligned} \tag{4}$$

where w is the number of intervals after segmentation, n is the length of the original time series, T_i is the distance coefficient between different trends (a penalty coefficient, with its value shown in Eq. (5)), qb_i, cb_i are the last points of the ith interval of Q and C, respectively, and qa_i, ca_i are the slopes of the ith interval of Q and C, respectively.

$$T_i = \left(\frac{\bar{Q} - \bar{C}}{\text{qb}_i + f \cdot \text{cb}_i}\right)^2, \quad f = \begin{cases} 0, & \text{cb}_i \ge 0, \\ -1, & \text{cb}_i < 0, \end{cases} \tag{5}$$

where $\bar{Q}$ and $\bar{C}$ are the mean values of time series Q and C, respectively.

In the following section, the distance of the symbolic series generated using TFSA will be compared with the Euclidean distance of the original series to verify if the TFSA distance lower-bounds the Euclidean distance, i.e., if $D_{\text{TFSA}}(Q, C) \le D_{\text{Euclidean}}(A, B)$. This proof will be based on the condition where there is a single TFSA frame, i.e., w=1. A more generalized proof for w>1 can be obtained by applying the single-frame proof to each frame.

Proof Substituting Eqs. (3) and (4) into $D_{\text{TFSA}}(Q, C) \le D_{\text{Euclidean}}(A, B)$ gives

$$\sqrt{\sum_{i=1}^{n}(a_i-b_i)^2} \ge \sqrt{n\cdot T_i\cdot(\text{qb}_i-\text{cb}_i)^2\cdot\frac{||\text{qa}_i|-|\text{ca}_i||}{\max(|\text{qa}_i|,|\text{ca}_i|)}}. \tag{6}$$

Squaring both sides of inequality (6) gives

$$\sum_{i=1}^{n}(a_i-b_i)^2 \ge n\cdot T_i\cdot(\text{qb}_i-\text{cb}_i)^2\cdot\frac{||\text{qa}_i|-|\text{ca}_i||}{\max(|\text{qa}_i|,|\text{ca}_i|)}. \tag{7}$$

Because a_i can be represented as $a_i=\overline{Q}-\Delta a_i$, the same applies to $b_i=\overline{C}-\Delta b_i$. Thus, the left side of inequality (7) can be rewritten as

$$\sum_{i=1}^{n}[(\overline{Q}-\Delta a_i)-(\overline{C}-\Delta b_i)]^2=\sum_{i=1}^{n}[(\overline{Q}-\overline{C})-(\Delta a_i-\Delta b_i)]^2.$$

Furthermore, the left side of inequality (7) can be expanded as

$$\sum_{i=1}^{n}[(\overline{Q}-\overline{C})^2-2(\overline{Q}-\overline{C})(\Delta a_i-\Delta b_i)+(\Delta a_i-\Delta b_i)^2].$$

Using the distributive law gives

$$n(\overline{Q}-\overline{C})^2-2(\overline{Q}-\overline{C})\sum_{i=1}^{n}(\Delta a_i-\Delta b_i)+\sum_{i=1}^{n}(\Delta a_i-\Delta b_i)^2. \tag{8}$$

Because $a_i=\overline{Q}-\Delta a_i$, then $\Delta a_i=\overline{Q}-a_i$, and thus $\Delta b_i=\overline{C}-b_i$. Therefore,

$$\begin{aligned}\sum_{i=1}^{n}(\Delta a_i-\Delta b_i)&=\sum_{i=1}^{n}[(\overline{Q}-a_i)-(\overline{C}-b_i)]\\&=\left(\sum_{i=1}^{n}\overline{Q}-\sum_{i=1}^{n}a_i\right)-\left(\sum_{i=1}^{n}\overline{C}-\sum_{i=1}^{n}b_i\right)\\&=\left(n\overline{Q}-\sum_{i=1}^{n}a_i\right)-\left(n\overline{C}-\sum_{i=1}^{n}b_i\right)\\&=\left(\sum_{i=1}^{n}a_i-\sum_{i=1}^{n}a_i\right)-\left(\sum_{i=1}^{n}b_i-\sum_{i=1}^{n}b_i\right)\\&=0.\end{aligned}$$

Hence, the left side of inequality (7) becomes

$$n(\overline{Q}-\overline{C})^2+\sum_{i=1}^{n}(\Delta a_i-\Delta b_i)^2.$$

Since

$$T_i\le\left(\frac{\overline{Q}-\overline{C}}{\text{qb}_i-\text{cb}_i}\right)^2,\quad\frac{||\text{qa}_i|-|\text{ca}_i||}{\max(|\text{qa}_i|,|\text{ca}_i|)}\le 1,$$

the right side of inequality (7) satisfies

$$n\cdot T_i\cdot(\text{qb}_i-\text{cb}_i)^2\cdot\frac{||\text{qa}_i|-|\text{ca}_i||}{\max(|\text{qa}_i|,|\text{ca}_i|)}\le n(\overline{Q}-\overline{C})^2.$$

So, inequality $\sum_{i=1}^{n}(\Delta a_i-\Delta b_i)^2\ge 0$ is established, and inequality (6) holds true. The proof is complete.

4.3 Preprocessing for knowledge discovery

Due to the trend features, one feature of TFSA that distinguishes it from previous symbolization methods is that it facilitates subsequent data mining work, such as knowledge discovery. If the angle space of the slopes is (−90°, 90°), the space can be divided into a series of non-overlapping intervals. Each interval corresponds to a number (from 1 to 9) that indicates the change in the steepness or degree of trends. The angle space and corresponding numbers are defined in Table 2. It is observed that the slopes of the interval series can be transformed from infinity numbers into a limited set of symbols.

Table 2 Degree of trends and the corresponding angle ranges

Degree of trends	Angle range 1	Angle range 2
1	[0°, 10°)	(−10°, 0°]
2	[10°, 20°)	(−20°, −10°]
3	[20°, 30°)	(−30°, −20°]
4	[30°, 40°)	(−40°, −30°]
5	[40°, 50°)	(−50°, −40°]
6	[50°, 60°)	(−60°, −50°]
7	[60°, 70°)	(−70°, −60°]
8	[70°, 80°)	(−80°, −70°]
9	[80°, 90°)	(−90°, −80°]

However, the last point value b after standardization is still in infinity, so it is necessary to transform the infinity numbers to limited symbols for knowledge discovery. Given that normalized time series has highly Gaussian distribution, some 'split points' can be found, which will determine n equal-sized areas under the Gaussian curve. The split points are a sorted list of numbers $\beta_1, \beta_2, \ldots, \beta_{n-1}$ such that the area under an $N(0, 1)$ Gaussian curve from β_i to β_{i+1} is equal to $1/n$ (β_0 and β_n are defined as $-\infty$ and ∞, respectively). These split points can be found in a statistical table. Table 3 gives the split points for the value of n from 3 to 9.

Once the split points have been obtained from the lookup table, the value of the last point b can be discretized using the following method. First, the value of b is standardized. Then all points b that are below the smallest split point are mapped to symbol 'A', all points b greater than or equal to the smallest split point and less than the second smallest split point are mapped to symbol 'B', and so on. Fig. 10 illustrates the idea.

In this way, time series can be converted to a series of trend symbols, each with an understandable meaning. Take the series illustrated in Fig. 10 as an example. The results of symbolization can be further expressed as $Q_1Q_2Q_3Q_4Q_5Q_6Q_7Q_8 = 01^6_{\mathrm{B}}10^6_{\mathrm{B}}01^8_{\mathrm{D}}11^0_{\mathrm{D}}10^7_{\mathrm{C}}00^0_{\mathrm{C}}10^7_{\mathrm{A}}01^2_{\mathrm{A}}$. So, for knowledge discovery of the time series, the preprocessing is finished.

Table 3 A lookup table containing the split points which divide a Gaussian distribution into equiprobable regions

β_i	n=3	n=4	n=5	n=6	n=7	n=8	n=9
β_1	−0.43	−0.67	−0.84	−0.97	−1.07	−1.15	−1.22
β_2	0.43	0.00	−0.25	−0.43	−0.57	−0.67	−0.76
β_3	–	0.67	0.25	0.00	−0.18	−0.32	−0.43
β_4	–	–	0.84	0.43	0.18	0.00	−0.14
β_5	–	–	–	0.97	0.57	0.32	0.14
β_6	–	–	–	–	1.07	0.67	0.43
β_7	–	–	–	–	–	1.15	0.76
β_8	–	–	–	–	–	–	1.22

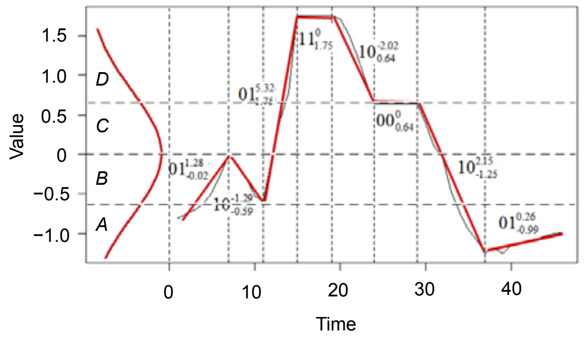

Fig. 10 The last point b mapped to symbols (n=4)

5 Experimental evaluation

The goal of these experiments was to demonstrate the efficiency and accuracy of TFSA used in long time series compared with those of other symbolic representation methods. Finally, the performance of TFSA for knowledge discovery was evaluated.

5.1 Efficiency of the two-step segmentation mechanism

In this experiment, satellite telemetry parameter data were chosen as the test data, whose characteristics are given below:

Large size: The size of the original satellite dataset was more than 4 GB for one day.

Multiple parameters: The number of parameters was more than one thousand.

The top-down (TD) and sliding window (SW) methods were selected to compare with two-step segmentation. For SW, the window size was set as 20 and the sliding step 1. The two-step segmentation was run with different numbers of initial segmentations k, pre-defined as 50 and 100, respectively. In the second step of segmentation, the intervals were split in parallel using Algorithms 1 and 2. So, the execution time of the two-step segmentation was computed as follows:

$$T = t_{\text{first-step}} + \max\{t_{Q_1}, t_{Q_2}, \ldots, t_{Q_k}\}. \tag{9}$$

The length of the time series varied from 1000 to 1 000 000. As the length of the time series increased beyond 100 000, two-step segmentation was clearly faster than TD and SW (Table 4). When the length of

Table 4 A comparison of segmentation efficiency with different time series lengths

Algorithm	Execution time (s)			
	1000	10 000	100 000	1 000 000
TD	1.80	10.38	120.12	1570.35
SW (fit=20, step=1)	1.13	3.01	70.93	648.51
Two-step segmentation				
k=50	2.29	4.49	10.52	21.86
k=100	2.18	3.72	7.75	15.68

the series reached 1 000 000, the two-step segmentation was about 72 to 100 times faster than TD (with the number of initial segmentations set as 50 and 100, respectively).

From the results in Table 4, when the length was 1000, SW was apparently the fastest algorithm and TD the second fastest. SW efficiently handled short time series thanks to its lower complexity ($O(n)$). When the length of the time series was less than 10 000, two-step segmentation was not as competitive as the other two approaches due to the allocations of tasks for parallel implementation. However, the advantage of two-step segmentation was gradually but increasingly unveiled with a slight increase in processing time as the time series length extended up to 1 000 000. Conversely, the TD and SW algorithms showed their weakness in handling time series longer than 100 000. In this experiment, the sliding step of SW was set as 1. Clearly, a longer sliding step would reduce the execution time of the algorithm, but it would also lead to the loss of key points.

5.2 Accuracy of TFSA

The accuracy of TFSA was examined using the classification results from a set of time series. TFSA was compared with the Euclidean distance and shape description alphabet (SDA) (André-Jönsson and Badal, 1997) methods. In the classification experiments, a synthetic dataset, cylinder-bell-funnel (CBF), was considered, as it has also been used in similarity comparison and clustering by Manganaris (1997) and Kadous (1999). The dataset contains three classes, resulting from the following equation:

$$\begin{cases} c(t)=(6+\eta)\cdot X_{[a,b]}(t)+\varepsilon(t), \\ b(t)=(6+\eta)\cdot X_{[a,b]}(t)\cdot(t-a)/(b-a)+\varepsilon(t), \\ f(t)=(6+\eta)\cdot X_{[a,b]}(t)\cdot(b-a)/(b-t)+\varepsilon(t), \\ X_{[a,b]}=\begin{cases} 0, & t<a, \\ 1, & a\le t\le b, \\ 0, & t>b. \end{cases} \end{cases} \tag{10}$$

Fig. 11 shows some examples of the cylinder, bell, and funnel classes classified by the three methods. The time series A to E are members of the funnel class, a to e are members of the bell class, and 1 to 5 are members of the cylinder class.

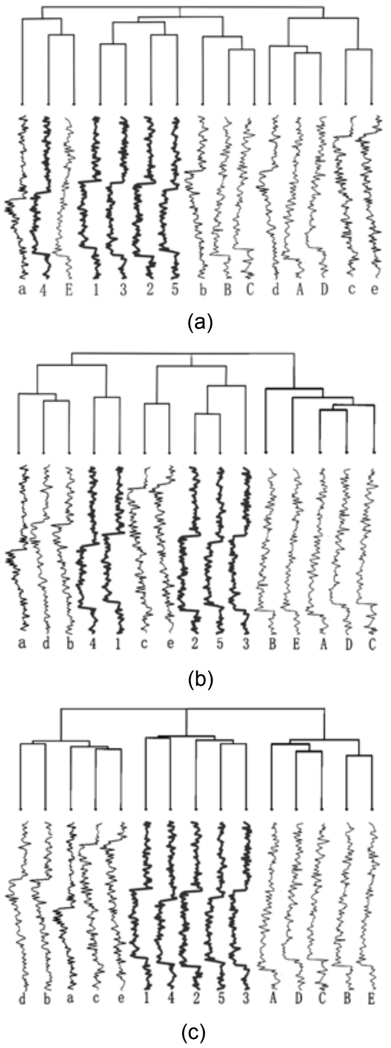

Fig. 11 Classification by Euclidean distance (a), shape description alphabet (b), and trend feature symbolic approximation (c), with a compression ratio of 80%

Although the Euclidean distance can be calculated quickly, the Euclidean distance approach is incapable of fully distinguishing the three classes. For SDA, prior knowledge of the data distribution of the time series is needed to set the split points and the discretized time series, but it does not preserve the general shape of the data. So, the accuracy of SDA was reduced, and only the funnel class was correctly classified. However, because it considers more trend features, the classification results obtained by TFSA were more acceptable. TFSA preserved most of the trend information. In this experiment, with a compression ratio of 80%, its classification results were the best.

5.3 Experiments for only long time series

Section 5.1 confirmed that TFSA is very suitable for symbolizing long time series. In this experiment, the datasets were changed to test the classification accuracy of only long time series. Some examples of the electrocardiogram (ECG) dataset (Moody and Mark, 1983) are shown in Fig. 12, including recordings of many common and life-threatening arrhythmias along with examples of normal sinus rhythm.

Parts of the dataset were chosen for this experiment, including three heartbeat classes. The numbers of series belonging to the three classes are shown in Table 5. The length of each series was about 100 000. Because different numbers of segmentations lead to different classification results, to guarantee the fairness of the experiment, SAX and TFSA used the same number of segmentations.

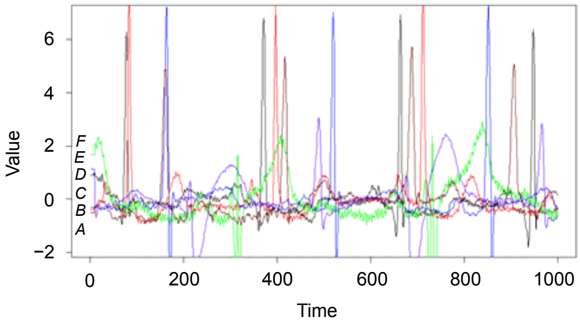

Fig. 12 Example series for a ECG dataset (for simplicity, only parts of the dataset are plotted)

Table 5 The heartbeat classes and the number of series

Number of records for supraventricular ectopic beat	Ventricular ectopic beat	Fusion beat
200	300 V	200 F

SAX, which is an excellent symbolic representation method, was selected for comparison with TFSA. The classification accuracy of TFSA was higher than that of SAX with the same number of segmentations (Fig. 13). In each class, SAX misclassified more series than TFSA. Although SAX can convert long time series very easily, it does not pay enough attention to their trends and will generate similar results for completely different time series. For long time series, the two-step segmentation mechanism is very useful for finding key points, and using trend symbols to represent these features will make TFSA more accurate in classification.

5.4 Knowledge discovery from time series

In Section 4.3, preprocessing for knowledge discovery has been proposed, and using the symbolic results, data mining of time series can be improved. The test data in the experiment came from the Second International Diagnostic Competition (DXC’10) (Poll *et al.*, 2010), and the data were from the ADAPT-Lite Electrical Power System (EPS) (Fig. 14). The sensors for the EPS are shown in Table 6, which illustrates the rate at which the data were collected. Anomaly detection and association rules mining were chosen to test the mining efficiency of TFSA for time series.

Anomaly detection: A common anomaly detection method for time series is to build a model by learning from previously observed normal data. Newly obtained data can be compared with those of this model and any lack of conformity is considered an anomaly (Dasgupta and Forrest, 1996). In this

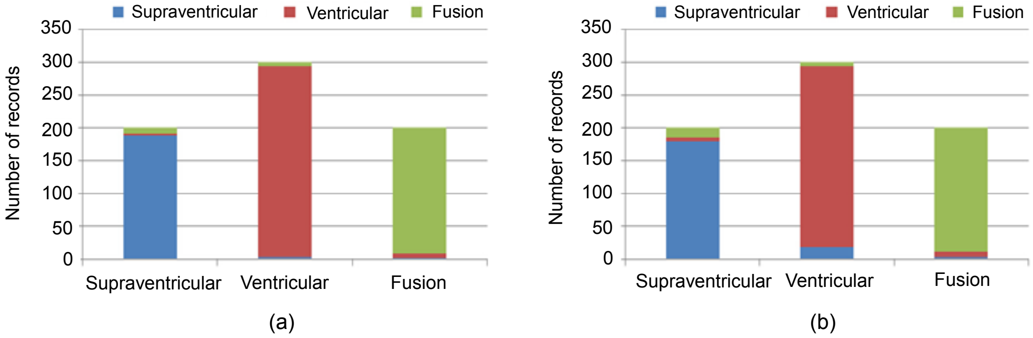

Fig. 13 Classification results from trend feature symbolic approximation (a) and symbolic aggregate approximation (b), with the same number of segmentations

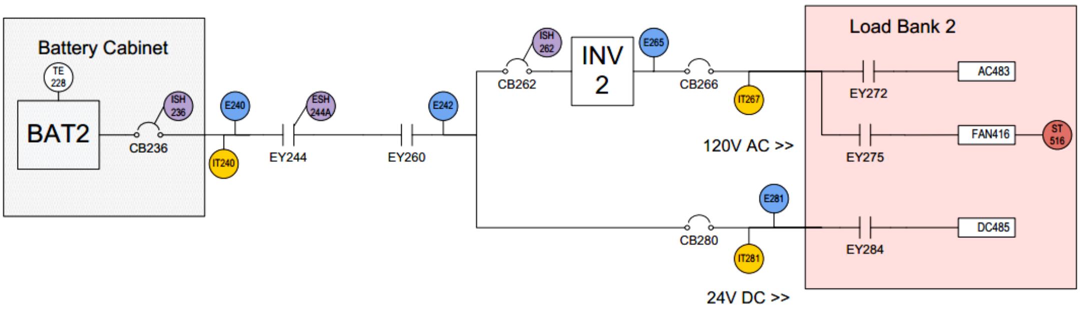

Fig. 14 ADAPT-Lite Electrical Power System (sensors are marked with circles)

study, the model was built on the symbols of trend features. Symbols ‘01’, ‘10’, ‘00’, ‘11’ were used to represent the trends, and the numbers ‘1, 2, …, 9’ the slopes. The n in Table 3 was set as 10 in this experiment, and thus the alphabetic series ‘A, B, …, J’ represents the areas to which the last points of intervals belonged. In Fig. 15, the data collected by the IT240 sensor are shown, and the abrupt increase indicates a fault.

Using TFSA to symbolize the series, the results were ‘10D1’, ‘01F1’, ‘01A1’, ‘10D1’, ‘01H1’, ‘01I3’, ‘10B1’, ‘10E1’, ‘10A1’, ‘01H1’, ‘10I1’, ‘10C1’, ‘01B1’, ‘01G1’, ‘10E1’, ‘10A1’, ‘01C1’, ‘10H1’, ‘10G1’, ‘01H1’, ‘01E1’, ‘10I1’, ‘10B1’, ‘10G1’, ‘01B1’, ‘10I1’, ‘01I1’, ‘01J1’. The normal data were from the interval of ‘10D1’ to ‘01H1’, and a simple model is that the slopes of these intervals are all 1, which indicates that the status of the sensor is stable. After these intervals, symbol ‘01I3’ was not conforming with the model, and it means that a fault occurred in the interval (the time axis is [719,722]). The detection result was right, according to Fig. 15.

Association rules mining: To mine the association rules, multiple time series were considered, and the symbols were prefixed using the sensor name, such as ‘E265-10A1’. The apriori algorithm (Borgelt and Kruse, 2002) was used for mining association rules, and the symbols with their prefix were the items. To generate the itemsets as the input of the algorithm, the time axis was divided into 24 equal-sized areas, where the symbols and faults belonging to the same time area were gathered to form an itemset. For brevity, only parts of the experimental results are shown in Table 7.

Rule 1 describes the relationship between the sensors IT240 and IT267, and means that when the state of IT240 is ‘10J1’, the state of IT267 is ‘01F1’ with a probability of 100%. This kind of rule is very important, and can be used for monitoring the sensor IT267. If the state ‘10J1’ of IT240 appears, but the state of IT267 is not ‘01F1’, it is very likely that some fault has occurred in IT267. A rule about faults is also found, such as rule 2, and this kind of rule can be used to prognose the relevant fault, which is a sensor offset fault in this rule.

Table 6 The sensor rate group

Frequency (Hz)	Sensor
1	TE228
2	E265, E281, IT267, IT281, ST516
10	E240, E242, ESH244A, ISH236, IT240

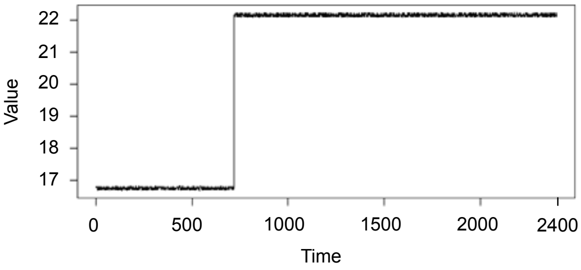

Fig. 15 Data from the IT240 sensor (the abrupt increase indicates a fault)

6 Conclusions

With huge amounts of data continuing to emerge, the length of time series tends to increase. In this paper, we propose a two-step segmentation mechanism which can be used in parallel to improve the segmentation efficiency for long time series. Unlike

Table 7 Association rules obtained from time series

ID	Rule	Support	Confidence
1	{IT240-10J1} => {IT267-01F1}	0.0430107	1.0000000
2	{IT240 offset fault} => {E240-01G1}	0.0537634	0.7142857
3	{ISH236-10I1} => {E281-10D1}	0.0430107	0.8000000
4	{ISH236-10F1} => {IT267-01F1}	0.0645161	0.7500000
5	{IT240-01B1} => {E242-01I1}	0.0752688	0.7000000
6	{E281-01J1} => {ESH244A-01I1}	0.0860215	0.6666667
7	{E265-10B1} => {E240-01G1}	0.1182795	0.6111111
8	{IT240-00C0, ISH236-01I1} => {ESH244A-01I1}	0.0430107	1.0000000
9	{IT240-01I1, E265-10B1, E240-01G1} => {IT267-01F1}	0.0430107	1.0000000

other symbolic methods, TFSA focuses on retaining most of the trend features and patterns of the original time series, and represents time series using trend symbols which are also suitable for knowledge discovery. Experimental results show that, especially for long time series, the segmentation efficiency and classification accuracy of TFSA are better than those of other methods.

The aim of this paper is to provide a new knowledge discovery method for time series. In section 5.4, we introduce how to use TFSA for simple association rules mining. The next step for the work is to consider massive data mining, and to study how to mine association rules from large time series symbolized using TFSA.

References

Agrawal, R., Srikant, R., 1995. Mining sequential patterns. Proc. 11th Int. Conf. on Data Engineering, p.3-14. [doi:10.1109/ICDE.1995.380415]

André-Jönsson, H., Badal, D.Z., 1997. Using signature files for querying time-series data. Proc. 1st European Symp. on Principles of Data Mining and Knowledge Discovery, p.211-220. [doi:10.1007/3-540-63223-9_120]

Bao, D., Yang, Z., 2008. Intelligent stock trading system by turning point confirming and probabilistic reasoning. *Expert Syst. Appl.*, **34**(1):620-627. [doi:10.1016/j.eswa.2006.09.043]

Borgelt, C., Kruse, R., 2002. Induction of association rules: apriori implementation. Proc. Computational Statistics, p.395-400. [doi:10.1007/978-3-642-57489-4_59]

Bu, Y., Chen, L., Fu, A.W.C., *et al.*, 2009. Efficient anomaly monitoring over moving object trajectory streams. Proc. 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, p.159-168. [doi:10.1145/1557019.1557043]

Chan, K.P., Fu, A.W.C., 1999. Efficient time series matching by wavelets. Proc. 15th Int. Conf. on Data Engineering, p.126-133. [doi:10.1109/ICDE.1999.754915]

Dasgupta, D., Forrest, S., 1996. Novelty detection in time series data using ideas from immunology. Proc. 5th Int. Conf. on Intelligent Systems, p.82-87.

Esling, P., Agon, C., 2012. Time-series data mining. *ACM Comput. Surv.*, **45**(1), Article 12. [doi:10.1145/2379776.2379788]

Faloutsos, C., Ranganathan, M., Manolopoulos, Y., 1994. Fast subsequence matching in time-series databases. Proc. ACM SIGMOD Int. Conf. on Management of Data, p.419-429. [doi:10.1145/191839.191925]

Guimarães, G., Ultsch, A., 1999. A method for temporal knowledge conversion. Proc. 3rd Int. Symp. on Advances in Intelligent Data Analysis, p.369-380. [doi:10.1007/3-540-48412-4_31]

Guimarães, G., Peter, J.H., Penzel, T., *et al.*, 2001. A method for automated temporal knowledge acquisition applied to sleep-related breathing disorders. *Artif. Intell. Med.*, **23**(3):211-237. [doi:10.1016/S0933-3657(01)00089-6]

Kadous, M.W., 1999. Learning comprehensible descriptions of multivariate time series. Proc. 16th Int. Conf. of Machine Learning, p.454-463.

Keogh, E., Chakrabarti, K., Pazzani, M., *et al.*, 2001. Locally adaptive dimensionality reduction for indexing large time series databases. Proc. ACM SIGMOD Int. Conf. on Management of Data, p.151-162. [doi:10.1145/375663.375680]

Kontaki, M., Papadopoulos, A.N., Manolopoulos, Y., 2005. Continuous trend-based classification of streaming time series. Proc. 9th East European Conf. on Advances in Databases and Information Systems, p.294-308. [doi:10.1007/11547686_22]

Kontaki, M., Papadopoulos, A.N., Manolopoulos, Y., 2008. Continuous trend-based clustering in data streams. Proc. 10th Int. Conf. on Data Warehousing and Knowledge Discovery, p.251-262. [doi:10.1007/978-3-540-85836-2_24]

Korn, F., Jagadish, H.V., Faloutsos, C., 1997. Efficiently supporting ad hoc queries in large datasets of time sequences. Proc. ACM SIGMOD Int. Conf. on Management of Data, p.289-300. [doi:10.1145/253260.253332]

Lavielle, M., Teyssière, G., 2006. Detection of multiple change-points in multivariate time series. *Lithuan. Math. J.*, **46**(3):287-306. [doi:10.1007/s10986-006-0028-9]

Lin, J., Keogh, E., Lonardi, S., *et al.*, 2003. A symbolic representation of time series, with implications for streaming algorithms. Proc. 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, p.2-11. [doi:10.1145/882082.882086]

Manganaris, S., 1997. Supervised Classification with Temporal Data. PhD Thesis, Vanderbilt University, USA.

Mannila, H., Toivonen, H., 1996. Discovering generalized episodes using minimal occurrences. Proc. Int. Conf. on Knowledge Discovery and Data Mining, p.146-151.

Mellit, A., Pavan, A.M., Benghanem, M., 2013. Least squares support vector machine for short-term prediction of meteorological time series. *Theor. Appl. Climatol.*, **111**(1-2): 297-307. [doi:10.1007/s00704-012-0661-7]

Moody, G.B., Mark, R.G., 1983. A new method for detecting atrial fibrillation using RR intervals. *Comput. Cardiol.*, **10**:227-230.

Phetking, C., Noor Md Sap, M., Selamat, A., 2008. A multiresolution important point retrieval method for financial time series representation. Proc. Int. Conf. on Computer and Communication Engineering, p.510-515. [doi:10.1109/ICCCE.2008.4580656]

Poll, S., de Kleer, J., Feldman, A., *et al.*, 2010. Second international diagnostics competition—DXC'10. Proc. 21st Int. Workshop on Principles of Diagnosis, p.1-15.

Sarkar, S., Mukherjee, K., Sarkar, S., *et al.*, 2013. Symbolic dynamic analysis of transient time series for fault detection in gas turbine engines. *J. Dynam. Syst., Meas. Contr.*, **135**(1):014506.1-014506.6. [doi:10.1115/1.4007699]

Villafane, R., Hua, K.A., Tran, D., *et al.*, 2000. Knowledge discovery from series of interval events. *J. Intell. Inform. Syst.*, **15**(1):71-89. [doi:10.1023/A:1008781812242]

Vullings, H.J.L.M., Verhaegen, M.H.G., Verbruggen, H.B., 1997. ECG segmentation using time-warping. Proc. 2nd Int. Symp. on Advances in Intelligent Data Analysis Reasoning about Data, p.275-285. [doi:10.1007/BFb0052847]

Yeh, A.B., Lin, D.K.J., Venkataramani, C., 2004. Unified CUSUM charts for monitoring process mean and variability. *Qual. Technol. Quant. Manag.*, **1**(1):65-86.