Bidirectional piecewise linear representation of time series with application to collective anomaly detection

Shi, Wen; Azzopardi, George; Karastoyanova, Dimka; Huang, Yongming

Published in:
Advanced Engineering Informatics

DOI:
10.1016/j.aei.2023.102155

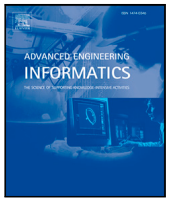
Publication date:
2023

Advanced Engineering Informatics 58 (2023) 102155

Contents lists available at ScienceDirect

Advanced Engineering Informatics

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



Full length article

Bidirectional piecewise linear representation of time series with application to collective anomaly detection

Wen Shi [a,b,*], George Azzopardi [a], Dimka Karastoyanova [a], Yongming Huang [b]

[a] Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Nijenborgh 9, 9747 AG, Groningen, The Netherlands
[b] School of Automation Engineering, Southeast University, 210006, Nanjing, China

ARTICLE INFO

Keywords:
Time series data
Data representation
Anomaly detection
Bidirectional piecewise linear representation (BPLR)
Similarity measurement

ABSTRACT

Directly mining high-dimensional time series presents several challenges, such as time and space costs. This study proposes a new approach for representing time series data and evaluates its effectiveness in detecting collective anomalies. The proposed method, called bidirectional piecewise linear representation (BPLR), represents the original time series using a set of linear fitting functions, which allows for dimensionality reduction while maintaining its dynamic characteristics. Similarity measurement is then performed using the piecewise integration (PI) approach, which achieves good detection performance with low computational overhead. Experimental results on synthetic and real-world data sets confirm the effectiveness and advantages of the proposed approach. The ability of the proposed method to capture more dynamic details of time series leads to consistently superior performance compared to other existing methods.

1. Introduction

With the rapid development of Internet of Things (IoT), a large number of sensors are capable of collecting and transmitting real-time data [1]. These data are often presented in the form of time series, which is a sequence of data points recorded in chronological order [2,3]. By effectively processing and analyzing time series data, the massive information provided by IoT can be better utilized. The benefits of dimensionality reduction in enhancing the efficiency and effectiveness of time series mining have gained widespread attention [4–6]. As a result, this paper concentrates on the development of a novel method for the representation of time series data in a lower-dimensional space.

Time series anomaly detection is a crucial subfield of time series data mining [7], which aims to identify unexpected behavior in the entire dataset. As anomalies are often caused by different mechanisms, they lack specific criteria for definition. In practice, data that exhibits expected behavior tend to receive greater attention, while anomalous data is often perceived as noise and subsequently disregarded or eliminated. However, anomalies can contain useful information, making their detection highly significant. For instance, in cybersecurity, anomalies in network traffic or user behavior can signal potential security breaches or attacks [8]. Additionally, precise anomaly detection can help mitigate unnecessary adverse effects in various fields, such as the environment [9], industry [10], finance [11], and others.

Anomalies in time series can be classified into the following three categories [12]; (1) Point anomalies: a data point is regarded as anomalous, with respect to the rest of the data points. These anomalies are often caused by measurement errors, sensor malfunctions, data input errors, or other exceptional events; (2) Contextual anomalies: in a specific context, a data point is considered anomalous, but otherwise not; and (3) Collective anomalies: a subsequence of a time series that exhibits abnormal behavior. This is quite a challenging task because such anomalies may not be considered anomalous when analyzed individually. Instead, it is the collective behavior of the group that is anomalous. Collective anomalies can also provide valuable insights into the underlying system or process being analyzed, as they may indicate a group-level problem or issue that needs to be addressed. Detecting collective anomalies can thus be an essential task in many fields, such as cybersecurity, finance, and healthcare [13]. In this paper, we focus on evaluating the proposed approach for the detection of collective anomalies.

The high dimensionality of time series data necessitates significant computational resources when using the original data to identify anomalies. However, to enhance anomaly detection efficiency, a typical approach involves reducing the dimensionality first and then utilizing a distance measure to perform the task in the transformed representation

* Corresponding author at: Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Nijenborgh 9, 9747 AG, Groningen, The Netherlands.
E-mail addresses: s.wen@rug.nl (W. Shi), g.azzopardi@rug.nl (G. Azzopardi), d.karastoyanova@rug.nl (D. Karastoyanova), huang_ym@seu.edu.cn (Y. Huang).

https://doi.org/10.1016/j.aei.2023.102155
Received 30 June 2023; Received in revised form 16 August 2023; Accepted 25 August 2023
Available online 6 September 2023

space. Therefore, we propose a novel bidirectional segmentation algorithm for piecewise linear representation, which we call BPLR. With this new method, we can transform the original time series into a low-dimensional expression form, which is suitable for efficient analysis. We also propose a novel similarity measurement based on the idea of piecewise integration (PI), which performs effective similarity measure computation with a relatively low computing overhead. Finally, the proposed BPLR and PI methods are combined to detect collective anomalies in time series.

The proposed BPLR method contributes to the field in the following ways:

- It enables efficient selection of segmentation points, which can significantly reduce the volume of original data and minimize computational complexity.
- It facilitates data visualization and allows for the identification of intrinsic characteristics of time series data.
- It enhances sensitivity to the volatility of the original time series, as the newly obtained time series reflects its dynamic characteristics after data representation.
- It provides a stable similarity measurement that yields more precise detection results than other approaches.

The remainder of this paper is organized as follows. Section 2 reviews the methodologies for detecting collective anomalies in time series. Section 3 introduces several definitions of time series and subsequences. Section 4 presents the details of the proposed BPLR-based method. Section 5 illustrates the computational complexity and evaluation criteria of the proposed method. Section 6 validates the effectiveness of the proposed method through comparison with other existing methods. Finally, the conclusions are drawn in Section 7.

2. Related works

Anomaly detection methods for collective anomalies can be mainly divided into three categories [5]: (1) prediction-based [14,15]; (2) classification-based [16,17]; and (3) representation-based [5,13,18], which are introduced in what follows.

2.1. Prediction- and classification-based methods

Prediction and classification-based methods for anomaly detection use supervised learning techniques to train the detection model that can separate normal data from anomalous data. These two methods have the following drawbacks; (1) Dependency on labels: classification and prediction-based methods require labeled training data, while obtaining labeled data can be difficult or expensive in many practical applications [5], (2) Inapplicable to unknown anomalies: the model can only recognize known anomaly patterns, and it may not effectively detect anomalies when faced with new ones [19], and (3) Difficult to interpret: classification and prediction models use statistical and machine learning techniques to identify patterns and make predictions based on input data. However, they are complex and difficult to interpret their inference [20].

2.2. Representation-based methods

Methods based on representation learning identify the internal structure and patterns of data using a new representational space. In this space, similarity assessments measure the differences between normal and anomalous data. Samples that significantly deviate in similarity from most samples are considered anomalous. This kind of method is unsupervised learning and does not require labeled anomaly data. This paradigm has a wide range of applications, can handle high-dimensional data, can discover complex anomalies, and can achieve high accuracy and robustness [13]. Representation-based methods contain two main steps, namely data representation and similarity measurement.

2.2.1. Data representation

Data representation refers to extracting core characteristics of given time series data and representing it in different forms. For capturing collective anomalies, the original time series need to be divided into a group of sub-series of data, called subsequences. Then, several approaches, such as piecewise aggregate approximation (PAA) [21], symbolic aggregate approximation (SAX) [22,23], information granulation theory [24], piecewise linear representation (PLR) [25,26], and others, can be used to represent each subsequence. Among these approaches, PAA works by dividing a time series into fixed-length segments and representing each segment with its average, thereby reducing the dimensionality of the data [21]. SAX takes this a step further by discretizing these averages into symbols, allowing the time series data to be represented at even lower dimensions [22]. Due to its remarkable capacity for reducing dimensionality while preserving essential features, SAX has garnered significant attention and undergone notable development in recent years For instance, Park and Jung [27] employ SAX to symbolize time series, and then use association rule mining for discovering frequent rules among the symbols of deviant events. Regarding information granulation theory, Duan et al. [28] propose a novel method for time series granulation, and apply it to construct a framework of time series clustering. Guo et al. [29] propose a trend-based granular representation method for time series and evaluate its effectiveness in a clustering task. The mentioned representation methods all achieve good results in their respective tasks. However, these techniques might inadvertently neglect or mitigate sudden changes or anomalous trends in the original data [30,31]. As a result, they are more popular for classification and clustering tasks. In contrast, PLR identifies key segmentation points within the original sequence where significant shifts in data behavior occur, and then uses several straight lines to connect the segmentation points. This approach decomposes nonlinear relationships into multiple linear segments for easier processing, enabling it to effectively capture and describe abrupt deviations and anomalous patterns within the data [12]. Therefore, PLR is more suitable for anomaly detection tasks. Kong et al. [13] effectively use PLR in combination with the weighted local outlier factor, achieving notably high accuracy in detecting anomalies within time series data. Existing segmentation criteria of PLR, include the bottom-up approach [32], top-down approach [33,34], and optimal partitioning approach [2], among others. These methods, however, have high computational complexity. To address this issue, we propose BPLR to select segmentation points more efficiently. Moreover, the proposed BPLR method fully considers the volatility of the original time series, which helps reduce the fitting error between representative subsequences and the original ones.

2.2.2. Similarity measurement

After data representation, the next step is to measure the similarity of all the subsequences in the representation space, and then the anomaly score of each subsequence can be obtained. Subsequences whose anomaly scores are higher than a given threshold are labeled as anomalies. There are several commonly utilized methods for similarity measurement, namely Longest common subsequence similarity (LCSS) [35,36], dynamic time warping (DTW) [37], and local outlier factor (LOF) [38], to name a few. These methods come with a high computational burden. Research in recent years has addressed this issue by attempting to propose new algorithms or optimize existing ones. For instance, Zhou et al. [39] employ efficient data structures to enhance the performance of LCSS in heart disease classification. Choi et al. [40] introduce two novel methods, namely fast Sakoe–Chiba DTW (SC-DTW) and fast incremental DTW (I-DTW), which exhibit faster computational speeds compared to the conventional DTW. Zhang et al. [41] propose a time series similarity measurement method, which utilizes series decomposition and DTW. This method possesses lower complexity than DTW and demonstrates notable efficiency in classification tasks. Liu et al. [42] propose a top-n local outlier detection method based on

Table 1
Symbols and abbreviations in this paper.

Symbol	Meaning
T	A time series
Y	A subsequence
M	Length of Y
ttp_r	A trend turning point
F_r	Importance factor of ttp_r
$VD(\cdot)$	Vertical distance
β	Deviation tolerance factor
δ_β	Maximum deviation of Y
TTP	Trend turning point
TTP_o	The list of trend turning points sorted in descending order of their importance factor
$BPLR$	Bidirectional piecewise linear representation
PI	Piecewise integration

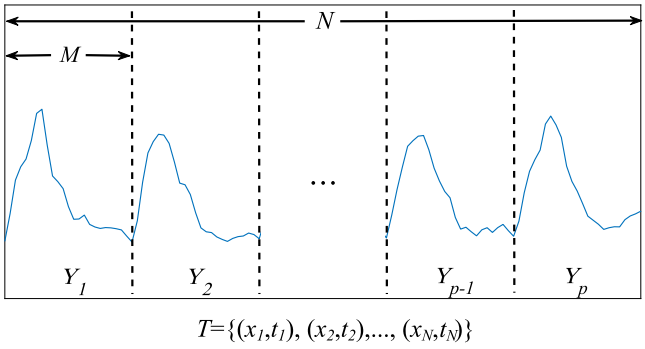


Fig. 1. An example of time series with partitioned subsequences.

Kernel Density Estimation (KDE), which minimizes the computational cost and performs well in detecting local outliers in data streams. The aforementioned methods, designed to calculate the distance between corresponding data points in two time series, have achieved good results on their respective datasets. Nevertheless, when confronted with the more abstract notion of linear segments, these methods fall short due to their lack of customization to account for the unique structure and distinct characteristics intrinsic to these segments [43]. Consequently, these methods do not offer suitable solutions for calculating distances between such segments. Therefore, we present a PI-based similarity method that can perform an effective similarity measure computation for the linear segments, with a relatively low computing overhead.

3. Preliminary

This section provides definitions related to the proposed anomaly detection approach based on BPLR. Table 1 contains a list of symbols and abbreviations frequently employed throughout this paper.

3.1. Time series and subsequences

Definition 1 (*Time Series [44]*)**.** A sequence of pairs, $T = \{(t_1, x_1), (t_2, x_2), \dots, (t_N, x_N)\}$, where $t_1 < t_2 < \cdots < t_N$, x_i represents the value of a data point at time t_i.

Definition 2 (*Subsequence [22]*)**.** Given a time series $T = \{(t_1, x_1), (t_2, x_2), \dots, (t_N, x_N)\}$, a subsequence Y of T can be obtained by extracting a window of size $M (M \le N)$, Fig. 1. One can split T in p−element non-overlapping subsequences $\{Y_1, Y_2, \dots, Y_p\}$, where $p = \lfloor N/M \rfloor$ and $\lfloor \cdot \rfloor$ denote rounding down.

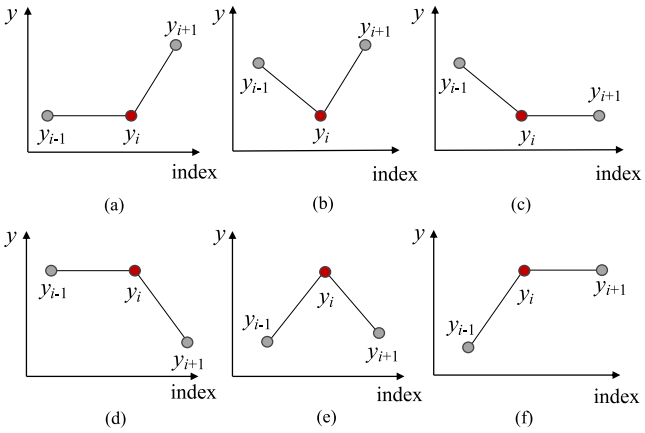


Fig. 2. Illustration of all possible trend turning points (TTPs).

3.2. Definitions of terms within a subsequence

Considering a time series $T = \{Y_1, Y_2, \dots, Y_p\}$, $Y_k = \{(t_{k,1}, y_{k,1}), (t_{k,2}, y_{k,2}), \dots, (t_{k,M}, y_{k,M})\}$ denotes a subsequence in T, where $k = 1, 2, \dots, p$. Note that we use Y, y_i and t_i to represent Y_k, $y_{k,i}$ and $t_{k,i}$ in the following text, for simplification reasons. Thus, we have $Y = \{(t_1, y_1), (t_2, y_2), \dots, (t_M, y_M)\}$. Then, we give the following definitions for a subsequence Y.

Definition 3 (*Trend Turning Points (TTPs)*)**.** Given a subsequence Y, each element (t_i, y_i) in Y that satisfies Eq. (1) is labeled as a trend turning point (TTP). $TTPs$ in Y can be expressed as $TTPs = \{ttp_1, ttp_2, \dots, ttp_r, \dots, ttp_Z\}$, where $1 \le r \le Z$, and $Z \le M$.

$$\begin{aligned} &\{y_i \in Y : y_i \ge y_{i-1} \text{ and } y_i > y_{i+1}\} \\ \cup\ &\{y_i \in Y : y_i > y_{i-1} \text{ and } y_i = y_{i+1}\} \\ \cup\ &\{y_i \in Y : y_i \le y_{i-1} \text{ and } y_i < y_{i+1}\} \\ \cup\ &\{y_i \in Y : y_i < y_{i-1} \text{ and } y_i = y_{i+1}\}, \end{aligned} \tag{1}$$

where $1 < i < M$.

From Eq. (1), it can be deduced that there are six situations where (t_i, y_i) can be defined as a trend turning point, as illustrated in Fig. 2. The essential structure of a given subsequence can be represented by its volatility characteristics. To this end, we introduce the concept of TTPs, which capture the morphological features of the original time series. Since a large number of TTPs may result from significant fluctuations in the time series, we rank them based on their importance according to the factor defined below.

Definition 4 (*Importance Factor (F)*)**.** Given a subsequence Y, F is the vertical distance [2] between the corresponding TTP and the mean value MV of Y:

$$F_r = VD(MV, ttp_r), r = 1, 2, \dots, Z, \tag{2}$$

where $VD(\cdot)$ denotes the vertical distance, and

$$MV = \frac{1}{M}\sum_{i=1}^{M} y_i. \tag{3}$$

Fig. 3 shows an example of a subsequence with $TTPs = \{ttp_1, ttp_2, ttp_3, ttp_4, ttp_5, ttp_6, ttp_7, ttp_8\}$.

Definition 5 (*BPLR of Y*)**.** Given a subsequence Y, BPLR is dedicated to identifying segmentation points in places where the behavior of the sequence changes significantly. Then, by connecting all segmentation points, the BPLR of Y can be defined as a group of linear segments, as shown in Fig. 4. It is notable that the first and last elements in Y are considered segmentation points by default.

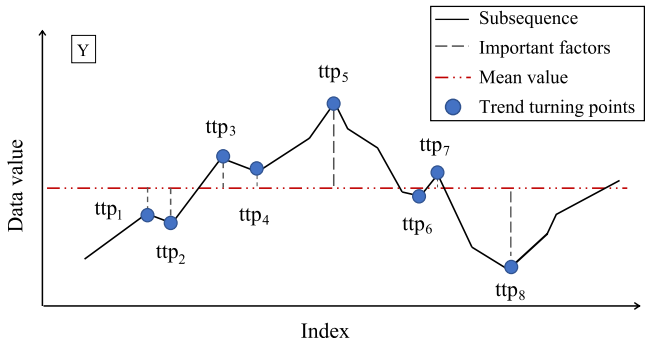


Fig. 3. An example of a subsequence with indicated trend turning points.

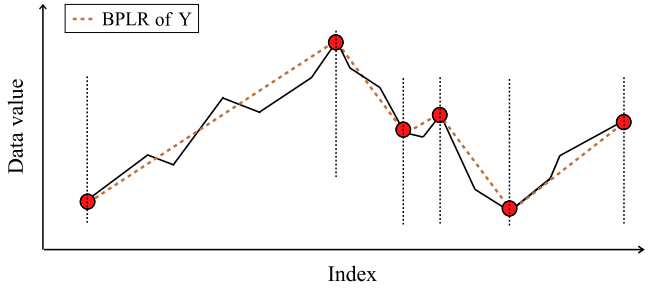


Fig. 4. BPLR of a subsequence. The red markers indicate the segmentation points.

Definition 6 (*Maximum Deviation*)**.** We denote by δ_β the maximum deviation of the segments in Y, that we use as a measure of segment quality. Within a segment, the vertical distances from all data points to the fitting line representing the segment must be less than or equal to δ_β, otherwise the segment is considered invalid.

$$\delta_\beta = |\max(Y) - \min(Y)| \cdot \beta, \tag{4}$$

where β denotes the deviation tolerance factor, which we set as a hyperparameter.

4. Methodology

Anomaly detection based on the proposed BPLR method contains two stages on which we elaborate below: (1) time series representation, and (2) similarity measurement.

4.1. Time-series representation based on BPLR

To complete the representation task, we need to find several sets of segmentation points in each subsequence, and then transform the original subsequence into a set of linear segments, as mentioned in Definition 5. In what follows, we give a specific description of the linear representation process for subsequence Y.

4.1.1. Determination of an ordered TPP list

Given a time series $T = \{(t_1, x_1), (t_2, x_2), \ldots, (t_N, x_N)\}$, firstly, it is divided into non-overlapping segments of equal width; $T = \{Y_1, Y_2, \ldots, Y_p\}$. The width can be either given or determined automatically, such as using auto cross-correlation [45]. To be specific, we begin by identifying peaks in the autocorrelation function of T and determining their corresponding lags. Following this, we calculate the differences between the lags of adjacent peaks, which represent the potential periods of the signal. Finally, we select the mode of these differences to define the width of the subsequences, denoted as M. Fig. 5 presents an example for calculating the value of M. In this particular instance,

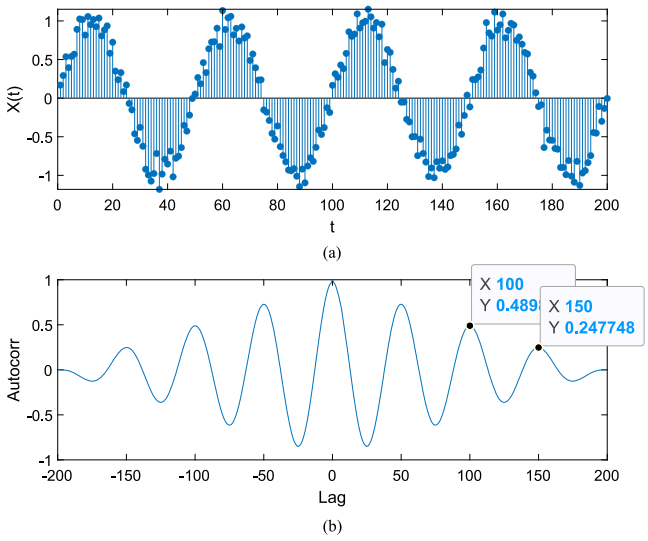


Fig. 5. Calculation process of M. (a) Time series. (b) Autocorrelation of the time series. In this example, M = 50.

we set $M = 50$ based on the differences between the lags of adjacent peaks.

Secondly, considering a subsequence $Y = \{(t_1, y_1), (t_2, y_2), \ldots, (t_M, y_M)\}$ in T, all $TTPs$ in Y can be obtained according to Eq. (1). Then, we denote by TTP_o the list of $TTPs$ sorted in descending order of their importance factor F. Algorithm 1 shows the calculation process of TTP_o.

Algorithm 1 Pseudocode for the determination of an ordered TPP list

Input: A subsequence $Y = \{(y_1, t_1), \ldots, (y_M, t_M)\}$
Output: TTP_o
Process:

```
function F_EVALUATION(Y_k)
    i ← 1;
    S ← 0;
    TTPs ← [];
    TTP_list ← [];
    while i ≤ M do
        Set S ← S + y_i;
        if (t_i, y_i) is a TTP then
            Set TTPs ← [TTPs , (y_i, t_i)];
        end if
        Set i ← i + 1;
    end while
    MV ← S/M;
    for r ← 1 : length(TTPs) do
        ttp_r ← TTPs^r;
        F_r ← VD(MV, ttp_r);
        Insert ttp_r into TTP_o in descending order according to the value of F_r;
    end for
    return TTP_o;
end function
```

4.1.2. Determination of linear segments

We use the ordered TTP list TTP_o to transform a given subsequence of raw values into a set of linear segments. This is achieved as follows. We start with the first TTP item in TTP_o, i.e. TTP^1 and use it as a starting position of the first two linear segments, one on the left-hand

side and one on the right-hand side. The endpoint of the first linear segment $BPLR^1$ is determined by scanning backward sequentially all the raw data points preceding TTP^1 and choose the last raw data point (t_j, y_j) such that all points between (t_j, y_j) and TTP^1 have a vertical distance less than δ_β (see Definition 6). If any of the elements in TTP_o are members of the set of raw points between (t_j, y_j) and TTP^1 then they are removed from TTP_o. Similarly, we determine the endpoint of the second linear segment $BPLR^2$ by scanning forward all raw data points succeeding TTP^1. We keep repeating this process by taking the next element in TTP_o to be the starting position of the next pair of linear segments, until TTP_o is empty. The pseudocode of this procedure is given in Algorithm 2.

Algorithm 2 Transform a subsequence of raw values into a set of linear segments

Require: A subsequence of raw values Y, an ordered TTP list TTP_o, and β
Ensure: A set of linear segments $\{BPLR\}$
1: $i \leftarrow 1$;
2: $BPLR \leftarrow []$;
3: **while** TTP_o is not empty **do**
4: Let (t_i^*, y_i^*) be the i-th element of TTP_o;
5: Find the last raw data point (t_j, y_j) by moving backwards from (t_i^*, y_i^*) such that $\forall\ (t_m, y_m) \in Y$ with $t_j \le t_m \le t_i^*$, $|(t_m - t_j)(y_j - y_i^*)/(t_j - t_i^*) + y_j - y_m| < \delta_\beta$;
6: Set $BPLR^1 \leftarrow \{(t_j, y_j), (y_i^*, t_i^*)\}$;
7: Remove any elements in TTP_o that are members of the set of raw points between t_j and t_i^*;
8: Find the last raw data point (t_k, y_k) by moving forward from (t_i^*, y_i^*) such that $\forall\ (t_m, y_m) \in Y$ with $t_i^* \le t_m \le t_k$, $|(t_m - t_k)(y_k - y_i^*)/(t_k - t_i^*) + y_k - y_m| < \delta_\beta$;
9: Set $BPLR^2 \leftarrow \{(t_i^*, y_i^*), (t_k, y_k)\}$;
10: Remove any elements in TTP_o that are members of the set of raw points between t_i^* and t_k;
11: $BPLR = \{BPLR, BPLR^1, BPLR^2\}$;
12: Set $i \leftarrow i + 1$;
13: **end while**
14: **return** $\{BPLR\}$

The value of the parameter β is directly related to the dimensionality reduction of the time series. Dimensionality reduction increases and fitting imprecision decreases with an increasing value of β. Fig. 6 shows an example of linear segments with different values of β, where $\beta_1 > \beta_2$.

4.2. Distance measure between two subsequences

For a given time series $T = \{Y_1, Y_2, \ldots, Y_p\}$, we denote by BPLR(T) its BPLR representation of a set of linear segments.

We denote by $d_{i,j}(T)$ the distance measure between two subsequences $BPLR_i(T)$ and $BPLR_j(T)$, which we define as the area between the two subsequences, Fig. 7. In practice, we compute this similarity by taking the absolute difference between the areas under $BPLR_i(T)$ and $BPLR_j(T)$:

$$d_{i,j}(T) = \left| \int BPLR_i(T) - \int BPLR_j(T) \right|, \tag{5}$$

where the piecewise integration of $BPLR_k(T)$ is defined as:

$$\int BPLR_k(T) = \sum_{a=1}^{\#BPLR_k(T)} \int BPLR_k^a(T), \tag{6}$$

where $BPLR_k^a(T)$ refers to the ath linear segment in $BPLR_k(T)$, and $\#BPLR_k(T)$ denotes the cardinality of the set $BPLR_k(T)$; i.e. the number of linear segments in $BPLR_k(T)$.

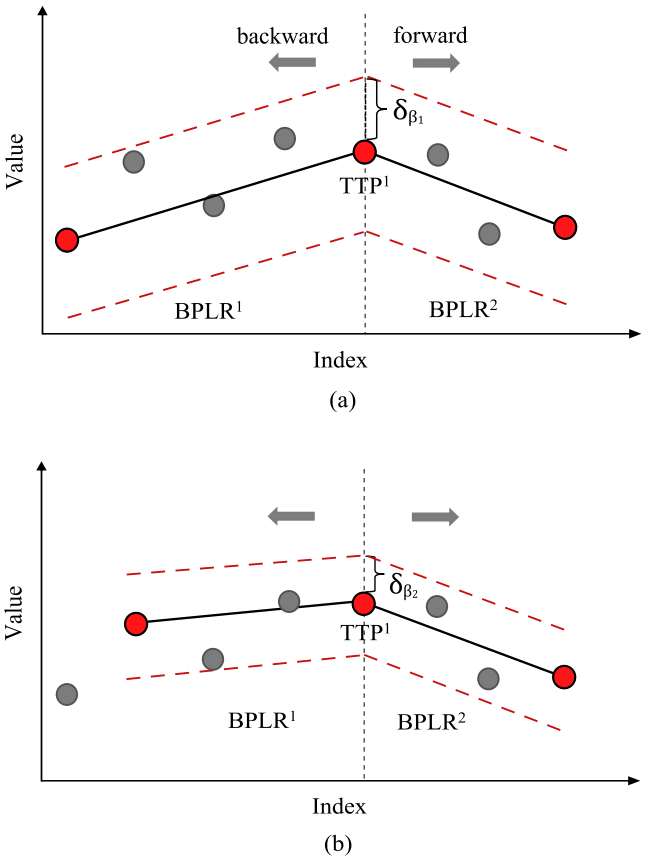


Fig. 6. Linear segments with different values of β. Red markers indicate the segmentation points.

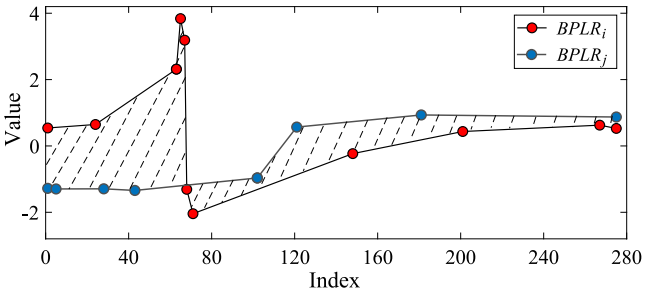


Fig. 7. Data representation of Y_i and Y_j.

4.3. Anomaly detection

Upon evaluating the pairwise distances between any two subsequences within the set $BPLR(T)$, a distance matrix denoted by M_{dist} can be obtained:

$$\mathrm{M}_{dist} = \begin{vmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,p} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ d_{p,1} & d_{p,2} & \cdots & d_{p,p} \end{vmatrix} = \begin{vmatrix} 0 & d_{1,2} & \cdots & d_{1,p} \\ d_{2,1} & 0 & \cdots & d_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ d_{p,1} & d_{p,2} & \cdots & 0 \end{vmatrix}. \tag{7}$$

By applying Eq. (8), the total distance D_i can be computed as the summation of the distances between $BPLR_i$ and the other subsequences. In essence, D_i represents the sum of elements in the ith row

of M_{dist}. In this study, we define the anomaly score A_i based on D_i, as shown in Eq. (9):

$$D_i = \sum_{j=1, i\neq j}^{p} d_{i,j}, \tag{8}$$

$$A_i = \frac{D_i \cdot p}{\sum_{i=1}^{p} D_i}, \tag{9}$$

where p represents the number of subsequences. If the value of A_i exceeds a predetermined threshold A^*, we identify A_i as an anomaly.

5. Analysis of the proposed approach

5.1. Computational complexity

When considering the computational complexity of the proposed approach, it is important to analyze the following three phases:

(1) Data representation based on the bidirectional piecewise linear representation (BPLR) method: Given a time series $T = \{(t_1, x_1), (t_2, x_2), \ldots, (t_N, x_N)\}$, it is divided into subsequences $T = \{Y_1, Y_2, \ldots, Y_p\}$, where $p = \lfloor N/M \rfloor$. In this phase, the algorithm calculates all the trend turning points in each subsequence to obtain TTP_o, which requires $O(pM \log M)$ operations. Additionally, the BPLR method determines all the segmentation points using the proposed segmentation criterion, taking $O(pMK)$ operations, K denotes the average number of the trend turning points in each subsequence, and $K \ll M$ generally. Hence, the first phase has a computational complexity of $O(pM \log M) + O(pMK)$, equivalent to $O(N \log M) + O(NK)$.

(2) Similarity measurement based on Piecewise Integration (PI): To compute the anomaly score A_i, the distances between P_i and the other PLR representations P_j (where $j \neq i$) must be calculated. This process has a complexity of $O(p^2)$.

(3) Anomaly detection based on the anomaly score: Comparing all p anomaly scores with the threshold A^* allows us to obtain the detection results, with a complexity $O(p)$.

In summary, the computational complexity of the proposed approach can be expressed as:

$$C = O(N \log M) + O(NK) + O(p^2) + O(p) \tag{10}$$

This indicates that the overall computational complexity is dominated by the terms $O(N \log M)$, $O(NK)$ and $O(p^2)$. Considering the data representation step, the computational complexity of our proposed BPLR method is $O(N \log M) + O(NK)$. However, the computational complexity of the most common segmentation criteria for PLR, such as bottom-up approach [32] and top-down [33], is $O(N) + O(N^2)$. In comparison, the BPLR method has a lower computational cost.

5.2. Evaluation criteria

Considering the validity evaluation of the proposed anomaly detection approach, we employ anomaly accuracy rate AR [5,13], and confidence index CI [13] to complete the evaluation task:

$$AR = \frac{m_k}{d_k}, \tag{11}$$

where m_k denotes the number of anomalies that are correctly detected, and d_k denotes the number of all known anomalies, and

$$CI = \frac{\text{mean}\left\{A_{anomaly}\right\}}{\text{mean}\left\{A_{all}\right\}}, \tag{12}$$

where $\text{mean}\left\{A_{anomaly}\right\}$ represents the mean of anomaly scores for all abnormal subsequences, and $\text{mean}\left\{A_{all}\right\}$ represents the mean of anomaly scores for all subsequences. According to Section 4.3, we need to calculate the anomaly score for each subsequence. The larger the anomaly score for anomaly patterns and the smaller the anomaly score for normal patterns, the better the performance of data anomaly resolution. Data anomaly resolution is the ability of the method to distinguish between normal and abnormal data, which is evaluated by CI in this paper.

Larger values of the two indicators imply better performance.

6. Experiments, results, and discussion

In this section, we present the framework of the proposed BPLR-based method and introduce the evaluation criteria. We proceed to demonstrate the performance of our approach in comparison with other major existing methods. For all datasets in this section, we generate receiver operating characteristic (ROC) curves [46] using different thresholds (in the range of 1 to 2 with a step size of 0.1) for the anomaly score and identify the optimal threshold based on the area under the curve (AUC). Following the steps mentioned above, the threshold is set as $A^* = 1.5$ in our method. The deviation tolerance factor in Eq. (4) is defined as $\beta = 5\%$. Figs. 8 to 16 display the original data, the BPLR representation of the original data, the BPLR representation of each subsequence, and the anomaly scores of each subsequence for the seven datasets. Anomalies in these datasets are highlighted in red.

6.1. Synthetic data

To start, we analyze a set of synthetic data generated using the following procedure [47]:

$$X(t) = \sin\left(\frac{40\pi t}{K}\right) + n(t) + e_1(t) + e_2(t), t \in [0, 1000], \tag{13}$$

where $X(t)$ represents the time series data with two manually added anomalies, denoted as $e_1(t)$ and $e_2(t)$, and $K = 1000$. Here, $n(t)$ refers to Gaussian noise with a standard deviation of 0.1. The expressions for $e_1(t)$ and $e_2(t)$ are given as follows:

$$e_1(t) = \begin{cases} -0.8X(t) + normrnd(0, 0.2), t \in [150, 189] \\ 0, \quad \text{otherwise} \end{cases}, \tag{14}$$

$$e_2(t) = \begin{cases} normrnd(0, 0.8), \quad t \in [900, 949] \\ 0, \quad \text{otherwise} \end{cases}. \tag{15}$$

Here, $normrnd(\cdot)$ represents a normal distribution function. In this study, the length of the sliding window is determined based on the period of the time series. Fig. 8 illustrates the BPLR representation and detection result of $X(t)$, with the sliding window size set to $M = 50$. Based on Fig. 8d, the evaluation indicators can be obtained, namely $AR = 2/2$ and $CI = 2.1349$.

6.2. Publicly available data

In this subsection, we conduct experiments using real-world data obtained from the UCR time series database[1] [48].

Figs. 9 to 12 display the BPLR representations and detection results of four electrocardiogram (ECG) datasets. The anomalies are highlighted in red. It can be observed in Figs. 9b, 10b, 11b, and 12b that the proposed BPLR method accurately captures the morphological details and trend variations of the original data. Furthermore, all anomalies are detected by the proposed approach, as depicted in Figs. 9d, 10d, 11d, and 12d. In the case of the anomalous subsequences in Figs. 9 to 11, these anomalies show clear numerical deviations or trend changes compared to the normal subsequences. Therefore, such anomalies are relatively easier to detect. However, the situation becomes much more complex with the anomalous subsequence in Fig. 12. Upon initial observation, this anomaly seems very similar to the normal subsequences, but the crucial difference lies in that their two peaks appear earlier than those in the normal subsequences. This type of anomaly is more challenging to capture compared to the previous types, as it involves subtler time series analysis, specifically recognizing the change in the order of peak occurrence. Nonetheless, our method can still accurately capture this type of anomaly. The evaluation indicators for the four ECG datasets are presented in Table 2, where m represents the sliding window size.

[1] https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

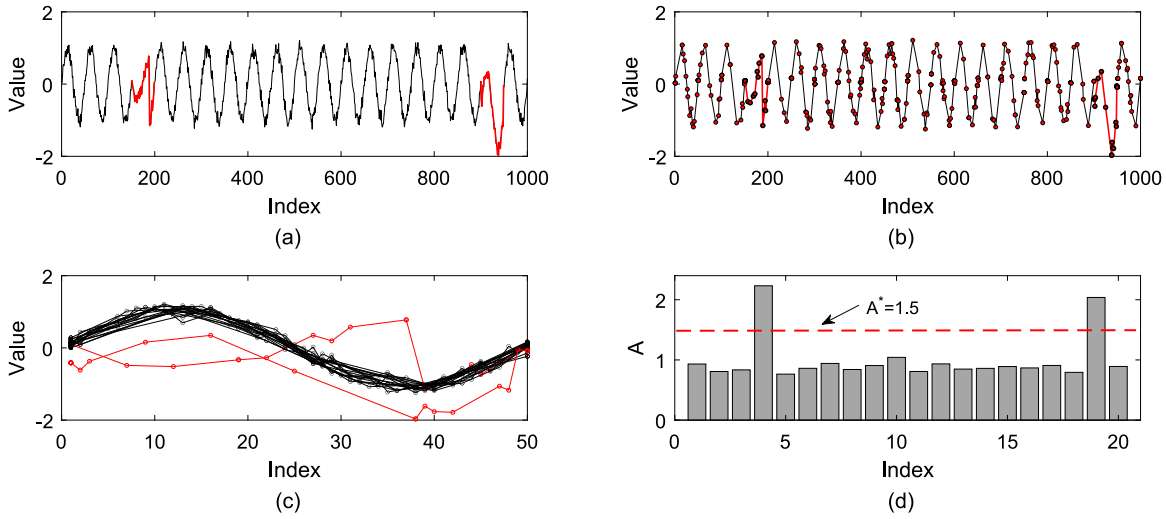


Fig. 8. Anomaly detection results (indicated in red) on synthetic data. (a) Original data. (b) BPLR of original data. (c) Superposition of the BPLRs of all subsequences. (d) Anomaly scores of all subsequences.

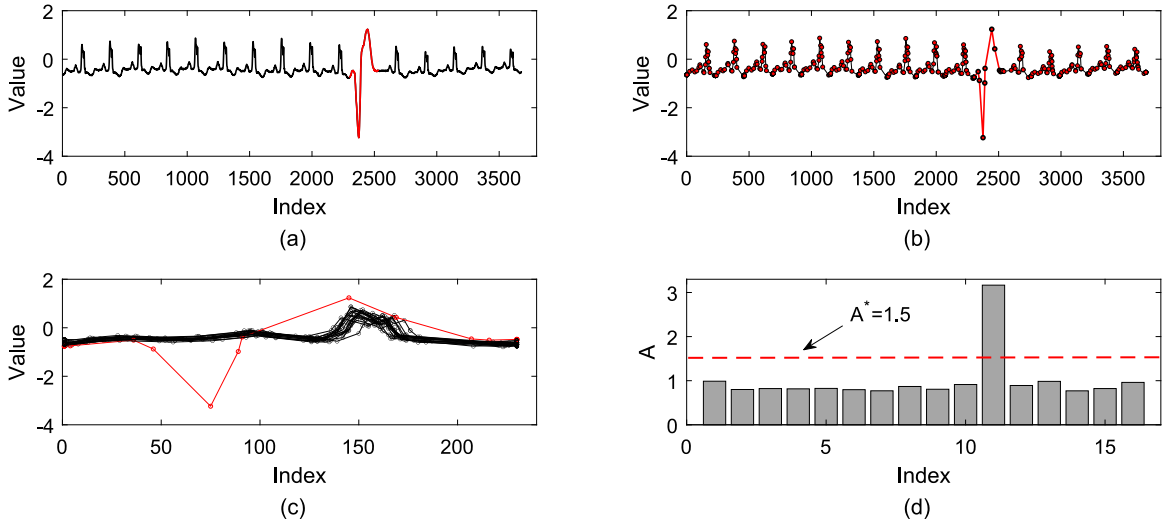


Fig. 9. Anomaly detection results (indicated in red) on ECG data. (a) Original ECG data. (b) BPLR of ECG data. (c) Superposition of the BPLRs of all subsequences. (d) Anomaly scores of all subsequences.

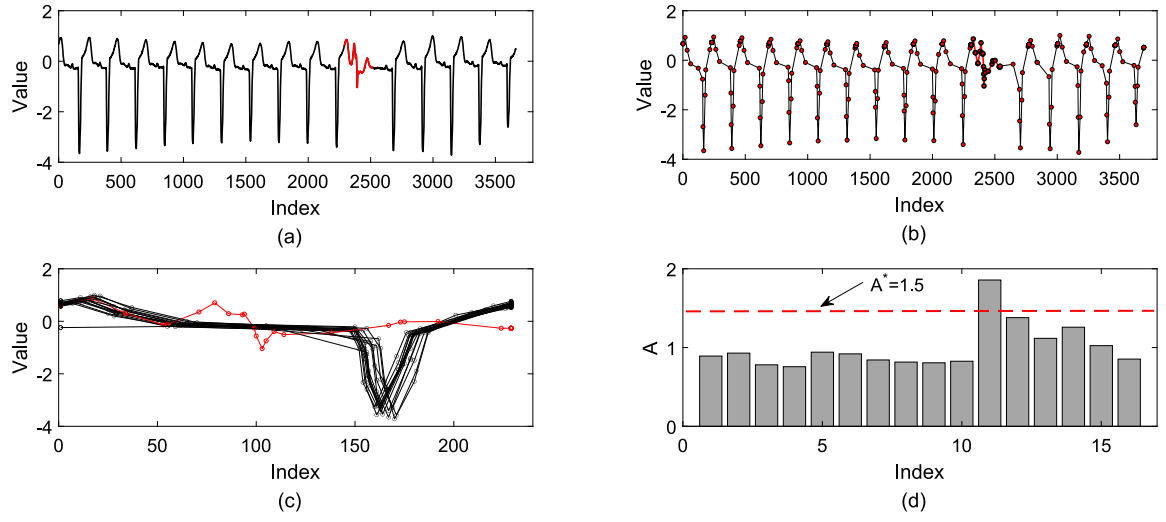


Fig. 10. Anomaly detection results (indicated in red) on ECG data. (a) Original ECG data. (b) BPLR of ECG data. (c) Superposition of the BPLRs of all subsequences. (d) Anomaly scores of all subsequences.

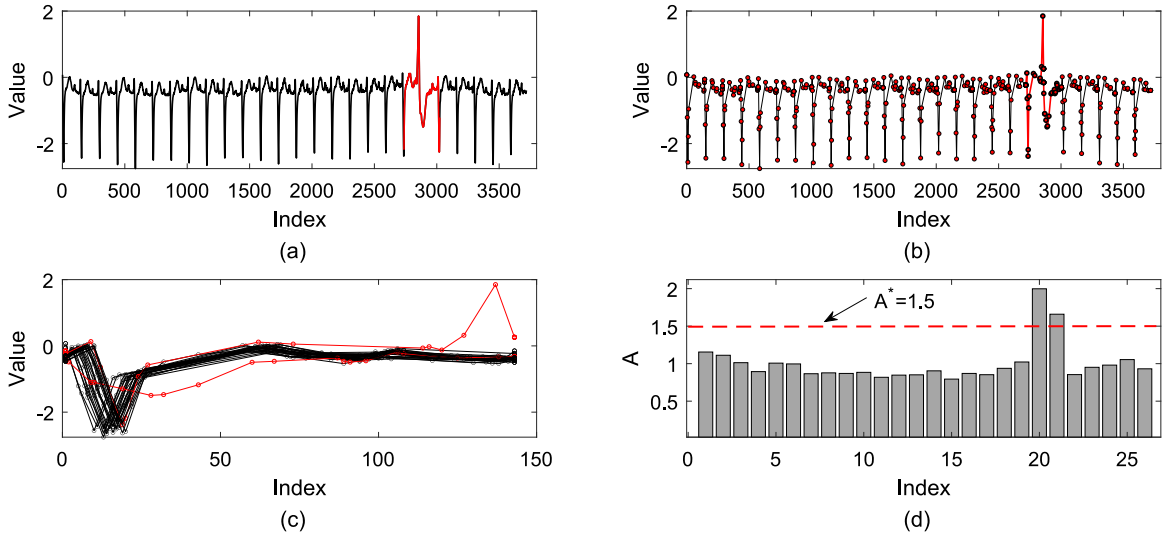


Fig. 11. Detection result on ECG data. (a) Original ECG data. (b) BPLR of ECG data. (c) BPLR of each subsequence. (d) Anomaly scores of all subsequences.

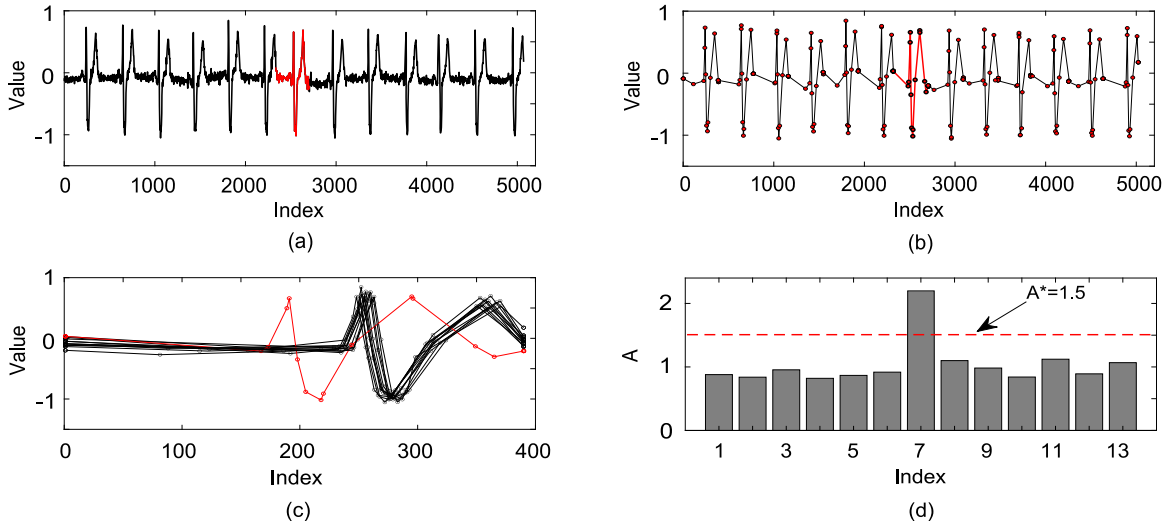


Fig. 12. Detection result on ECG data. (a) Original ECG data. (b) BPLR of ECG data. (c) BPLR of each subsequence. (d) Anomaly scores of all subsequences.

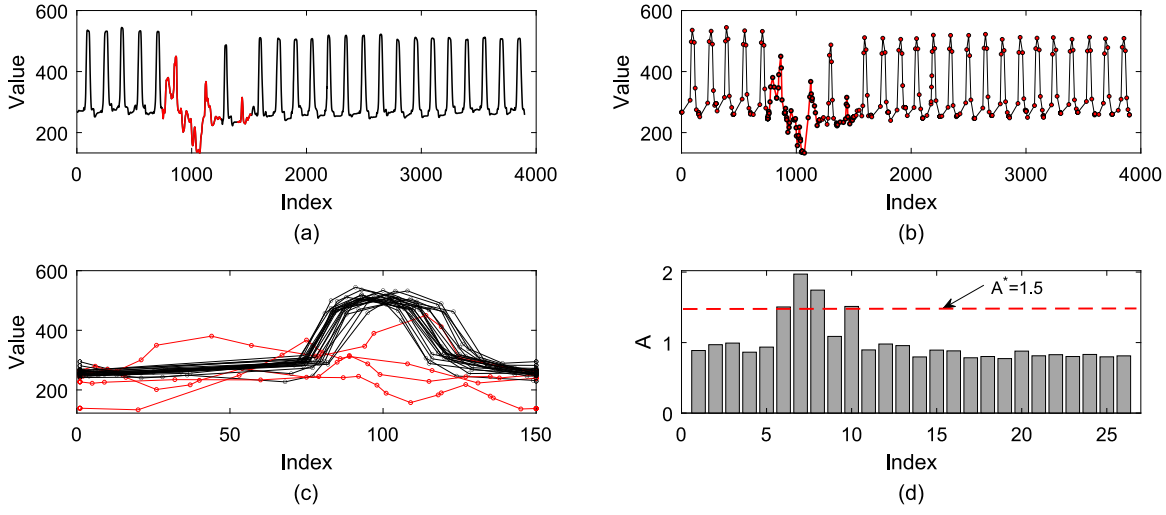


Fig. 13. Anomaly detection results (indicated in red) on video data. (a) Original video data. (b) BPLR of video data. (c) Superposition of the BPLRs of all subsequences. (d) Anomaly scores of all subsequences.

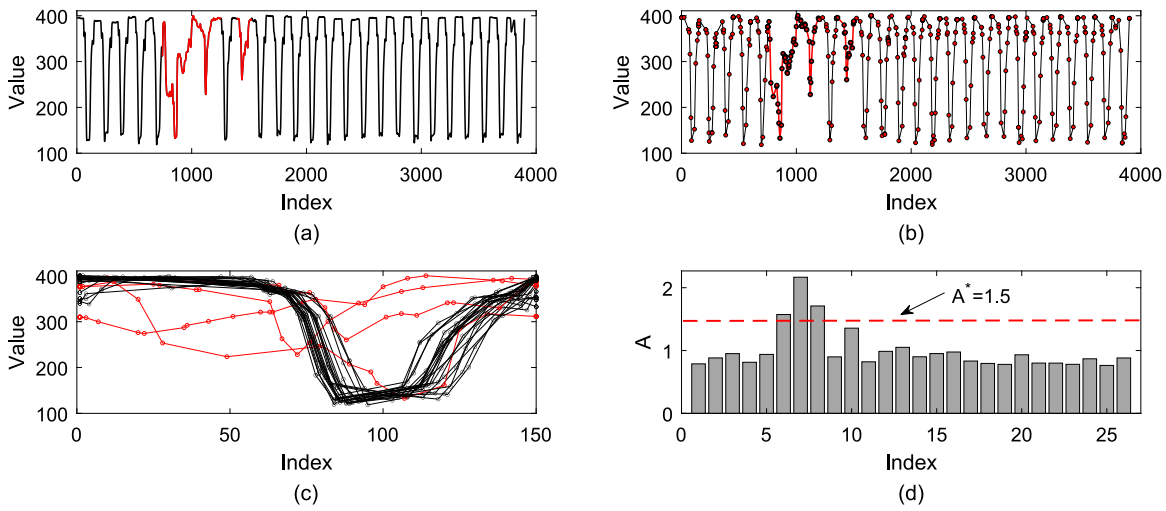


Fig. 14. Anomaly detection results (indicated in red) on video data. (a) Original video data. (b) BPLR of video data. (c) Superposition of the BPLRs of all subsequences. (d) Anomaly scores of all subsequences.

Table 2
Detection results on ECG data.

Dataset	Length	M	AR	CI
*chfdb_chf*01_275(1)	3750	230	1/1	2.806
*chfdb_chf*01_275(2)	3750	230	1/1	1.812
*chfdb_chf*13_45590_1	3750	143	2/2	1.829
*stdb*_308_0_1	5400	390	1/1	2.135

Table 3
Detection results on video data.

Dataset	Length	M	AR	CI
*ann_gun_CentroidA*_2(1)	3900	150	4/4	1.683
*ann_gun_CentroidA*_2(2)	3900	150	3/4	1.702

Figs. 13 and 14 show the data extracted from a video, which is more complex compared to ECG data. In Figs. 13a and 14a, it can be observed that these data contain anomalies caused by amplitude and shape variations. Fig. 13d displays the detection results of *ann_gun_CentroidA*_2(1), where all the anomalies are detected by the proposed approach. In Fig. 14d, one amplitude anomaly is missed, but its anomaly score is very close to A^*. Table 3 presents the evaluation indicators for these two video datasets.

Fig. 15 displays the BPLR representations and detection results on a spacecraft dataset. It is observed in Fig. 15a that the anomalous subsequence seems very similar to the normal ones in terms of their numerical values and trends. Specifically, the key characteristic of the anomalous subsequence lies in its narrower shape compared to the normal ones, a feature vividly depicted in Fig. 15c. Although such a difference might be challenging to highlight using common statistical features, our method uniquely equips us to identify and capture this type of anomaly accurately.

Fig. 16a illustrates a dataset representing arterial blood pressure of a man. As depicted in Fig. 16b, the morphology of the original data is well preserved. It can be observed that there is a shape anomaly in this dataset, which is accurately detected by the proposed approach.

In summary, the proposed BPLR method effectively reduces data dimensionality while precisely capturing the dynamic characteristics of the original data. The proposed PI-based similarity measurement method demonstrates good performance on the seven datasets mentioned above. Although one anomaly is missed in Fig. 14d, ranking the abnormal scores in descending order reveals that the top four scores correspond to the known anomalies. Therefore, it is crucial to develop a strategy that allows flexible adjustment of the threshold A^*, which is the focus of our future work.

Next, we conduct further comparative experiments with existing prominent representation methods. Specifically, we employ the PLR method [25] and PAA method [21] for data representation, and the DTW distance [34,37] for similarity measurement. These three methods are combined with the proposed method to perform comparative experiments, namely BPLR_PI (proposed method), PLR_PI, PAA_DTW, and PAA_PI. Additionally, three anomaly detection methods proposed in [5,13], and [36] are compared with our study. These three methods, as well as our BPLR_PI approach, are representation-based anomaly detection techniques. They identify anomalies by measuring the similarity among subsequences of time series data and allocating an anomaly score to each subsequence. The detection results of the comparative experiments are presented in Tables 4 and 5. The results detected by the proposed method are highlighted in bold. From Table 4, it can be observed that BPLR_PI achieves an improvement of 7.9 percentage points with respect to PLR_PI, and an improvement of 25.5 percentage points over PAA_DTW, and PAA_PI. From Table 5, it can be seen that our newly proposed BPLR_PI method demonstrates superior performance in terms of detection accuracy when contrasted with the methods outlined in [5,13], and [36]. BPLR_PI achieves an improvement of 6 percentage points over both methods in [5,13], and an improvement of 3.9 percentage points with respect to [36]. Furthermore, the CI value for BPLR_PI is observed to be larger than that of the methods proposed in [13,36].

The superior performance of BPLR_PI can be attributed to two main factors: (1) The BPLR-based data representation method fully considers the volatility of the original time series, resulting in a representation space that accurately reflects the dynamic characteristics of the original data. (2) The PI-based similarity measurement method is more stable and effective in detecting anomalies. Next, we conduct further comparative experiments with existing prominent representation methods. Specifically, we employ the PLR method [25] and PAA method [21] for data representation, and the DTW distance [34,37] for similarity measurement. These three methods are combined with the proposed method to perform comparative experiments, namely BPLR_PI (proposed method), PLR_PI, PAA_DTW, and PAA_PI.

The superior performance of BPLR_PI can be attributed to two main factors: (1) The BPLR-based data representation method fully considers the volatility of the original time series, resulting in a representation space that accurately reflects the dynamic characteristics of the original data. (2) The PI-based similarity measurement method is more stable and effective in detecting anomalies.

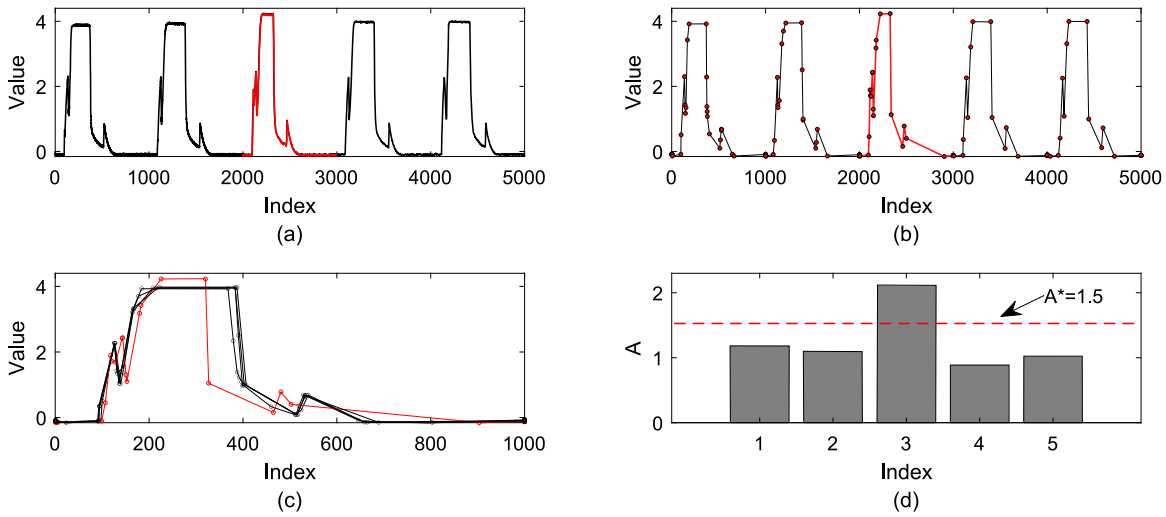


Fig. 15. Anomaly detection results (indicated in red) on spacecraft data. (a) Original spacecraft data. (b) BPLR of spacecraft data. (c) Superposition of the BPLRs of all subsequences. (d) Anomaly scores of all subsequences.

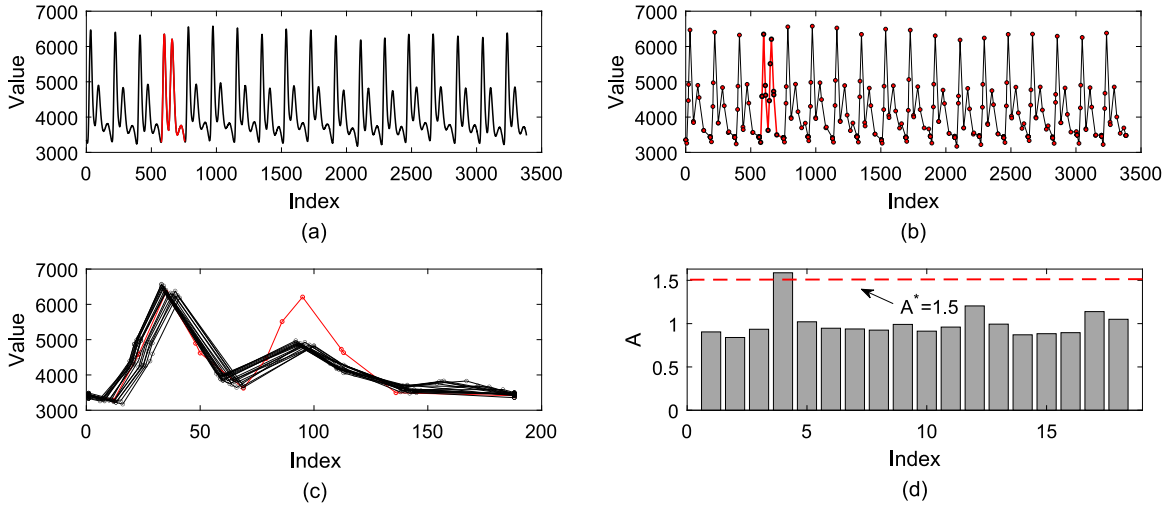


Fig. 16. Anomaly detection results (indicated in red) on blood pressure data. (a) Original blood pressure data. (b) BPLR of blood pressure data. (c) Superposition of the BPLRs of all subsequences. (d) Anomaly scores of all subsequences.

Table 4
Comparison experiment results with different representation methods.

Index	Dataset	Length	M	BPLR_PI		PLR_PI		PAA_DTW		PAA_PI	
				AR	CI	AR	CI	AR	CI	AR	CI
D1	*chfdb_chf*01_275_2	3750	230	**2/2**	**2.693**	2/2	2.406	1/2	1.781	1/2	1.834
D2	*chfdb_chf*01_275_3	3750	230	**2/2**	**2.309**	2/2	2.069	2/2	1.950	1/2	1.616
D3	*chfdb_chf*13_45590_1	3750	143	**2/2**	**1.829**	1/2	1.957	2/2	1.774	1/2	1.654
D4	*chfdb_chf*13_45590_2	3750	143	**2/2**	**2.101**	2/2	1.611	1/2	1.610	1/2	1.742
D5	*xmitdb_x*108_0_2	5400	350	**2/2**	**1.714**	2/2	1.721	1/2	1.781	1/2	1.812
D6	*xmitdb_x*108_0_3	5400	350	**2/3**	**1.857**	2/3	1.676	2/3	1.905	2/3	1.599
D7	*mitdb*_100_180	5400	120	**3/3**	**2.115**	2/3	1.732	2/3	1.565	2/3	1.851
D8	*mitdbx*108_1	12 000	330	**3/3**	**2.812**	3/3	2.105	2/3	1.975	2/3	1.469
D9	*mitdbx*108_2	12 000	330	**2/3**	**2.152**	2/3	1.671	2/3	1.460	2/3	1.573
D10	*qtdbsele*0606_1	15 000	143	**5/6**	**2.725**	5/6	1.373	4/6	1.358	4/6	1.401
D11	*qtdbsele*0606_2	15 000	143	**3/3**	**1.726**	2/3	2.215	1/3	1.498	2/3	1.676
D12	*qtdbsel*102_1	15 000	143	**1/1**	**2.928**	1/1	1.826	1/1	2.976	1/1	2.033
D13	*qtdbsel*102_2	15 000	143	**1/1**	**1.615**	1/1	1.699	1/1	1.951	1/1	1.617
D14	*chfdbchf*15	15 000	250	**4/5**	**1.693**	3/5	1.885	2/5	1.502	3/5	1.379
D15	*mitdbx_mitdbx*_108	20 000	340	**11/13**	**2.812**	10/13	1.448	7/13	1.532	7/13	1.139
Total				**44/51**	**33.081**	40/51	27.394	31/51	26.618	31/51	24.395
Average				**86.3%**	**2.205**	78.4%	1.826	60.8%	1.775	60.8%	1.626
Threshold A^*				**1.5**		1.4		1.4		1.4	

Table 5
Comparison experiment results with different anomaly detection methods.

Index	Dataset	Length	M	BPLR_PI		[5]		[13]		[36]	
				AR	CI	AR	CI	AR	CI	AR	CI
D1	*chfdb_chf*01_275_2	3750	230	**2/2**	**2.693**	1/2	/	1/2	2.031	1/2	2.179
D2	*chfdb_chf*01_275_3	3750	230	**2/2**	**2.309**	2/2	/	2/2	2.519	2/2	2.510
D3	*chfdb_chf*13_45590_1	3750	143	**2/2**	**1.829**	2/2	/	2/2	1.670	2/2	2.016
D4	*chfdb_chf*13_45590_2	3750	143	**2/2**	**2.101**	2/2	/	2/2	1.972	2/2	1.621
D5	*xmitdb_x*108_0_2	5400	350	**2/2**	**1.714**	2/2	/	2/2	1.854	2/2	1.792
D6	*xmitdb_x*108_0_3	5400	350	**2/3**	**1.857**	1/3	/	2/3	1.518	3/3	1.633
D7	*mitdb*_100_180	5400	120	**3/3**	**2.115**	2/3	/	2/3	2.507	2/3	2.257
D8	*mitdbx*108_1	12 000	330	**3/3**	**2.812**	3/3	/	2/3	1.836	2/3	2.098
D9	*mitdbx*108_2	12 000	330	**2/3**	**2.152**	2/3	/	2/3	2.335	2/3	1.993
D10	*qtdbsele*0606_1	15 000	143	**5/6**	**2.725**	5/6	/	5/6	2.575	5/6	2.558
D11	*qtdbsele*0606_2	15 000	143	**3/3**	**1.726**	2/3	/	2/3	1.593	2/3	1.625
D12	*qtdbsel*102_1	15 000	143	**1/1**	**2.928**	1/1	/	1/1	2.148	1/1	2.362
D13	*qtdbsel*102_2	15 000	143	**1/1**	**1.615**	1/1	/	1/1	1.958	1/1	1.735
D14	*chfdbchf*15	15 000	250	**4/5**	**1.693**	4/5	/	4/5	1.522	4/5	1.713
D15	*mitdbx_mitdbx*_108	20 000	340	**11/13**	**2.812**	11/13	/	11/13	2.096	11/13	2.682
Total				**44/51**	**33.081**	41/51	/	41/51	30.134	42/51	30.774
Average				**86.3%**	**2.205**	80.3%	/	80.3%	2.009	82.4%	2.052
Threshold A^*				**1.5**		/		1.5		1.5	

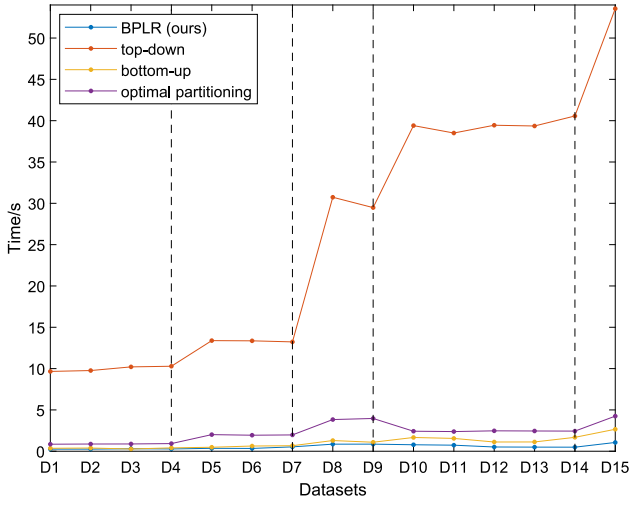


Fig. 17. Running time for different segmentation criteria of PLR. The vertical lines denote transitions in dataset lengths in terms of time points: 3750 for D1-D4, 5400 for D5-D7, 12 000 for D8-D9, 15 000 for D10-D14, and 20 000 for D15.

6.3. Time complexity comparison

In order to compare the running time of various segmentation criteria within PLR, specifically BPLR, bottom-up [32], top-down [33,34], and optimal partitioning [2], an experiment was conducted. The experiments involved 100 random validations across 15 different datasets from the UCR database, namely D1 to D15, as shown in Tables 4 and 5. The lengths of these datasets vary, ranging from 3750 to 20,000 data points.

Fig. 17 illustrates the average running time for each of the four segmentation criteria, providing insight into their comparative performance. It can be observed that as the length of the time series increases, the running time also generally increases. Among these four segmentation methods, the proposed BPLR method has the least running time across all 15 datasets. Bottom-up and optimal partitioning are approximately twice and three times the running time of BPLR, respectively, while top-down has the highest running time, far exceeding the other three methods.

7. Conclusion

In this paper, we introduce a novel data representation method for time series and evaluated its effectiveness in the task of collective anomaly detection. The proposed method, called BPLR, represents the original time series using a collection of linear fitting functions. This approach enables dimensionality reduction while preserving the dynamic characteristics of the data by capturing its volatility. Furthermore, we have employed the idea of PI-based similarity measurement, which achieves excellent detection performance with a relatively low computational overhead.

Experimental results from both synthetic and real-world datasets emphasize the efficacy and precision of our technique. Notably, in real-world datasets, our method showcases its unique capability to detect intricate anomalies, even those without overt numerical deviations or distinct trend shifts. When compared to major existing methods, our approach consistently achieves higher detection accuracy.

In future research, we will focus on exploring alternative representation forms for time series data, aiming to extract more essential information for data representation. Additionally, we intend to extend our approach to analyze multivariate time series, broadening its applicability in various domains.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data in Section 6.2 of the attached manuscript.

Acknowledgments

This work was supported in part by the China Scholarship Council under Grant 20220609190, in part by the Natural Science Foundation of Jiangsu Province China under Grant BK20220332, and in part by the Key Research and Development Program of Jiangsu Province under Grant BE2022154.

References

[1] G. Zhang, C.-H. Chen, X. Cao, R.Y. Zhong, X. Duan, P. Li, Industrial internet of things-enabled monitoring and maintenance mechanism for fully mechanized mining equipment, Adv. Eng. Inform. 54 (2022) http://dx.doi.org/10.1016/j.aei.2022.101782.

[2] X. Liu, Z. Lin, H. Wang, Novel online methods for time series segmentation, IEEE Trans. Knowl. Data Eng. 20 (12) (2008) 1616–1626.

[3] H. Li, Z. Liu, Multivariate time series clustering based on complex network, Pattern Recognit. 115 (2021) 107919.

[4] B.D. Fulcher, N.S. Jones, Highly comparative feature-based time-series classification, IEEE Trans. Knowl. Data Eng. 26 (12) (2014) 3026–3037, http://dx.doi.org/10.1109/TKDE.2014.2316504.

[5] Y. Zhou, H. Ren, Z. Li, et al., An anomaly detection framework for time series data: An interval-based approach, Knowl.-Based Syst. 228 (2021) 107153.

[6] H. Guo, M. Wan, L. Wang, X. Liu, W. Pedrycz, Weighted fuzzy clustering for time series with trend-based information granulation, IEEE Trans. Cybern. (2022) 1–12, http://dx.doi.org/10.1109/TCYB.2022.3190705.

[7] M. Ma, L. Han, C. Zhou, BTAD: A binary transformer deep neural network model for anomaly detection in multivariate time series data, Adv. Eng. Inform. 56 (2023) http://dx.doi.org/10.1016/j.aei.2023.101949.

[8] Z. Yang, X. Liu, T. Li, et al., A systematic literature review of methods and datasets for anomaly-based network intrusion detection, Comput. Secur. 116 (2022) 102675.

[9] D.J. Hill, B.S. Minsker, Anomaly detection in streaming environmental sensor data: A data-driven modeling approach, Environ. Model. Softw. 25 (9) (2010) 1014–1022.

[10] W. Fang, Y. Shao, P.E.D. Love, T. Hartmann, W. Liu, Detecting anomalies and denoising monitoring data from sensors: A smart data approach, Adv. Eng. Inform. 55 (2023) http://dx.doi.org/10.1016/j.aei.2022.101870.

[11] C. Rohitash, C. Shelvin, Evaluation of co-evolutionary neural network architectures for time series prediction with mobile application in finance, Appl. Soft Comput. 49 (2016) 462–473.

[12] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection for discrete sequences: A survey, IEEE Trans. Knowl. Data Eng. 24 (5) (2012) 823–839.

[13] X. Kong, Y. Bi, D.H. Glass, Detecting anomalies in sequential data augmented with new features, Artif. Intell. Rev. 53 (2020) 625–652.

[14] M. Li, Y. Shen, Q. Ren, H. Li, A new distributed time series evolution prediction model for dam deformation based on constituent elements, Adv. Eng. Inform. 39 (2019) 41–52, http://dx.doi.org/10.1016/j.aei.2018.11.006.

[15] Y. Yao, J. Ma, Y. Ye, KfreqGAN: Unsupervised detection of sequence anomaly with adversarial learning and frequency domain information, Knowl.-Based Syst. 236 (2022) 107757.

[16] X. Jin, Y. Guo, S. Sarkar, et al., Anomaly detection in nuclear power plants via symbolic dynamic filtering, IEEE Trans. Nucl. Sci. 58 (1 PART 2) (2011) 277–288.

[17] B. Lu, D. Xu, B. Huang, Deep-learning-based anomaly detection for lace defect inspection employing videos in production line, Adv. Eng. Inform. 51 (2022) http://dx.doi.org/10.1016/j.aei.2021.101471.

[18] H. Ren, M. Liu, X. Liao, et al., Anomaly detection in time series based on interval sets, IEEJ Trans. Electr. Electron. Eng. 13 (5) (2018) 757–762.

[19] M.A. Pimentel, D.A. Clifton, L. Clifton, et al., A review of novelty detection, Signal Process. 99 (2014) 215–249.

[20] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413–422.

[21] E. Keogh, K. Chakrabarti, M. Pazzani, et al., Dimensionality reduction for fast similarity search in large time series databases, Knowl. Inf. Syst. 3 (3) (2001) 915–937.

[22] E. Keogh, J. Lin, A. Fu, HOT sax: efficiently finding the most unusual time series subsequence, in: Fifth IEEE International Conference on Data Mining (ICDM'05), 2005, pp. 226–233.

[23] Y. Wan, X. Gong, Y.-W. Si, Effect of segmentation on financial time series pattern matching, Appl. Soft Comput. 38 (2016) 346–359.

[24] W. Pedrycz, W. Homenda, Building the fundamentals of granular computing: A principle of justifiable granularity, Appl. Soft Comput. 13 (10) (2013) 4209–4218.

[25] D. Yankov, E. Keogh, U. Rebbapragada, Disk aware discord discovery: finding unusual time series in terabyte sized datasets, Knowl. Inf. Syst. 17 (2008) 241–262.

[26] Q. Xie, C. Pang, X. Zhou, et al., Maximum error-bounded piecewise linear representation for online stream approximation, VLDB J. 23 (6) (2014) 915–937.

[27] H. Park, J.-Y. Jung, SAX-ARM: Deviant event pattern discovery from multivariate time series using symbolic aggregate approximation and association rule mining, Expert Syst. Appl. 141 (2020) 112950.

[28] L. Duan, F. Yu, W. Pedrycz, X. Wang, X. Yang, Time-series clustering based on linear fuzzy information granules, Appl. Soft Comput. 73 (2018) 1053–1067, http://dx.doi.org/10.1016/j.asoc.2018.09.032.

[29] H. Guo, L. Wang, X. Liu, W. Pedrycz, Trend-based granular representation of time series and its application in clustering, IEEE Trans. Cybern. 52 (9) (2022) 9101–9110, http://dx.doi.org/10.1109/TCYB.2021.3054593.

[30] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing SAX: a novel symbolic representation of time series, Data Min. Knowl. Discov. 15 (2) (2007) 107–144.

[31] X. Wang, A. Mueen, H. Ding, et al., Experimental comparison of representation methods and distance measures for time series data, Data Min. Knowl. Discov. 26 (2013) 275–309.

[32] E. Keogh, S. Kasetty, On the need for time series data mining benchmarks: A survey and empirical demonstration, Data Min. Knowl. Discov. 7 (4) (2003) 349–371.

[33] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, et al., Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping, ACM Trans. Knowl. Discov. Data 7 (3) (2013) 1–31, http://dx.doi.org/10.1145/2500489.

[34] H. Guo, L. Wang, X. Liu, et al., Information granulation-based fuzzy clustering of time series, IEEE Trans. Cybern. 51 (12) (2021) 6253–6261.

[35] M. Vlachos, G. Kollios, D. Gunopulos, Discovering similar multidimensional trajectories, in: Proceedings 18th International Conference on Data Engineering, 2002, pp. 673–684, http://dx.doi.org/10.1109/ICDE.2002.994784.

[36] T. Nguyen, P. Phuc, C. Yang, et al., Time-series anomaly detection using dynamic programming based longest common subsequence on sensor data, Expert Syst. Appl. 213 (2023) 118902.

[37] H. Izakian, W. Pedrycz, I. Jamal, Fuzzy clustering of time series data using dynamic time warping distance, Eng. Appl. Artif. Intell. 39 (2015) 235–244.

[38] M. Breunig, H. Kriegel, R. Ng, et al., LOF: Identifying density-based local outliers, Sigmod Record 29 (2) (2000) 93–104.

[39] X. Zhou, L. Chen, P. Li, J. Luo, X. Xiao, F. Lin, A novel symbolic representation for heart disease classification with lightgbm, in: 2016 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, IEEE, 2016, pp. 1200–1205.

[40] W. Choi, J. Cho, S. Lee, Y. Jung, Fast constrained dynamic time warping for similarity measure of time series data, IEEE Access 8 (2020) 222841–222858, http://dx.doi.org/10.1109/ACCESS.2020.3043839.

[41] Q. Zhang, C. Zhang, L. Cui, X. Han, Y. Jin, G. Xiang, Y. Shi, A method for measuring similarity of time series based on series decomposition and dynamic time warping, Appl. Intell. 53 (2023) 6448–6463.

[42] F. Liu, Y. Yu, P. Song, Y. Fan, X. Tong, Scalable KDE-based top-n local outlier detection over large-scale data streams, Knowl.-Based Syst. 204 (2020) http://dx.doi.org/10.1016/j.knosys.2020.106186.

[43] Y. Yu, D. Zhu, J. Wang, Y. Zhao, Abnormal data detection for multivariate alarm systems based on correlation directions, J. Loss Prev. Process Ind. 45 (2017) 43–55, http://dx.doi.org/10.1016/j.jlp.2016.11.011.

[44] D. Yankov, E. Keogh, U. Rebbapragada, Disk aware discord discovery: Finding unusual time series in terabyte sized datasets, in: Seventh IEEE International Conference on Data Mining (ICDM 2007), 2007, pp. 381–390.

[45] A.V. Oppenheim, R.W. Schafer, Discrete-Time Signal Processing, third ed., Prentice Hall, Upper Saddle River, NJ, 2010.

[46] T. Fawcett, An introduction to ROC analysis, Pattern Recognit. Lett. 27 (8) (2006) 861–874.

[47] Y. Qiu, X. Shi, W. Kai, Probabilistic distance based abnormal pattern detection in uncertain series data, Knowl.-Based Syst. 36 (2012) 182–190.

[48] H.A. Dau, A. Bagnall, K. Kamgar, et al., The UCR time series archive, IEEE/CAA J. Autom. Sin. 6 (6) (2019) 1293–1305.