

Bidirectional Piecewise Linear Representation of Time Series and Its Application in Clustering

Wen Shi¹, Dimka Karastoyanova¹, Member, IEEE, Yongming Huang², and Guobao Zhang¹

Abstract—The high dimensionality of time-series data presents challenges for direct mining, including time and computational resource costs. In this study, a novel data representation method for time series is proposed and validated in a hierarchical clustering task. First, the bidirectional segmentation algorithm, called BPLR, is introduced for piecewise linear representation (PLR). Through this method, the original time series is transformed into a set of linear fitting (LF) functions, thereby producing a concise, lower-dimensional LF time series that encapsulates the original data. Next, based on dynamic time warping (DTW) distance, a new similarity measure is proposed to compute the distance between any two LF time series, which is called LF-DTW distance. The proposed LF-DTW distance exhibits good performance in handling time-scale distortions between time series. Finally, hierarchical clustering is realized based on the proposed LF-DTW distance. The efficiency and advantages of the proposed approach are validated through experimental results using real-world data. Owing to its ability to capture the inherent structure of time series, the proposed approach consistently outperforms methods based on classic distance metrics and other existing clustering algorithms.

Index Terms—Bidirectional piecewise linear representation (BPLR), hierarchical clustering, linear fitting (LF) time series, similarity measure, time-series data.

I. INTRODUCTION

THE development of the Internet of Things (IoT) has been a significant trend in recent years, driven by advancements in communication technologies, cloud computing, and data analytics [1]. IoT devices, sensors, and systems generate a vast amount of time-series data, which is a sequence of data points collected in chronological order [2]. Time-series data is prevalent in various fields, including finance [3], [4], [5], meteorology [6], [7], [8], and industry [9], [10], [11], [12].

Manuscript received 17 August 2023; accepted 12 September 2023. Date of publication 25 September 2023; date of current version 5 October 2023. This work was supported in part by the China Scholarship Council under Grant 20220609190, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20220332, and in part by the Key Research and Development Program of Jiangsu Province under Grant BE2022154. The Associate Editor coordinating the review process was Dr. Yu Yang. (*Corresponding author: Yongming Huang*)

Wen Shi is with the School of Automation Engineering, Southeast University, Nanjing 210006, China, and also with the Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, 9747 AG Groningen, The Netherlands (e-mail: s.wen@rug.nl).

Dimka Karastoyanova is with the Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, 9747 AG Groningen, The Netherlands (e-mail: d.karastoyanova@rug.nl).

Yongming Huang and Guobao Zhang are with the School of Automation Engineering, Southeast University, Nanjing 210006, China (e-mail: huang_ym@seu.edu.cn; guobaozh@seu.edu.cn).

Digital Object Identifier 10.1109/TIM.2023.3318728

among others. Time-series data mining helps in revealing the underlying structure, relationships, and dependencies in the data, leading to better decision-making and predictions [13]. It encompasses various tasks, including time-series clustering [14], prediction [15], [16], and anomaly detection [17], among others. Time-series data exhibit high-dimensional characteristics, making it difficult to analyze and mine directly. Dimensionality reduction techniques can be used to transform high-dimensional time-series data into a lower-dimensional representation, which can greatly reduce computation and improve efficiency in data mining [18], [19]. Therefore, this article concentrates on the development of a novel method for representing time-series data in a lower-dimensional space and evaluates its performance in a clustering task.

Time-series clustering is an important subfield in time-series data mining, aiming to group time series with similar characteristics or behaviors [20]. For time-series clustering, a typical idea is to first reduce the dimensionality and then employ a distance measure to cluster the time series in the transformed representation space. There are several commonly used data representation methods, including piecewise aggregate approximation (PAA) [21], symbolic aggregate approximation (SAX) [22], [23], information granulation theory [18], [24], piecewise linear representation (PLR) [25], [26], [27], and others. Among these methods, PLR can better represent the prototypes of data and provide meaningful interpretations of the reduced representation. PLR aims to find a set of segmentation points in the sequence where there is a significant change in behavior and then to fit a straight line between these points [28]. The segmentation points can effectively partition the original sequence into regions where it can be approximated by a linear function with acceptable errors. Thus, a set of linear functions can be used to represent the original time series, and the resulting time series in the representation space is called the linear fitting (LF) time series. However, the computation complexity of existing segmentation criteria of PLR, such as the bottom-up method [29], [30], top-down method [29], [31], sliding window-based method [28], and optimal partitioning method [32], among others, is quite high. To address this issue, we propose a novel bidirectional segmentation criterion for PLR, which we call BPLR, for the efficient selection of segmentation points. Moreover, the proposed BPLR method fully considers the volatility of the original time series, which can capture its dynamic features.

After reducing the dimensionality of the time series, the next step is to cluster them in the representation space.

For obtaining accurate clustering results, a suitable distance measure is required. The obtained LF time series may have different lengths, and in time series of the same class, similar patterns may not appear in the same time intervals. Therefore, traditional distance measures, such as Euclidean distance [33], [34] and Hausdorff distance [35], are not suitable for our method. The similarity measurement methods proposed in [18] and [19] are relatively new similarity measures that have achieved good results. However, these two measures are based on information granulation methods and are not suitable for the LF time series obtained in this article. By allowing nonlinear warping in the time dimension, dynamic time warping (DTW) distance [36], [37], [38], [39] can find the optimal alignment between two time series, regardless of their length, making it a suitable method for comparing time series of unequal lengths. Thus, based on the DTW distance, this article proposes a new similarity measure called LF-DTW distance, which can be used to calculate the distance between any two unequal-length LF time series. Based on LF-DTW distance, the hierarchical clustering task can be completed.

The main advantages of this study are exhibited as follows.

- 1) Efficient selection of segmentation points. After segmentation, the original data volume can be significantly decreased, contributing to a reduction in computational complexity.
- 2) Data visualization can be realized and the intrinsic characteristics of time-series data can be mined.
- 3) Sensitive to the volatility of original time series. After data representation, the newly obtained LF time series can reflect the dynamic attributes of the original one.
- 4) A stable similarity measure is provided, which allows for the nonlinear warping of time series and can measure the distance between unequal-length LF time series.

The remainder of this article is organized as follows. Section II introduces several definitions within a time series. Section III presents the details of the proposed BPLR-based representation method. Section IV elaborates on the clustering process for the obtained LF time series. Section V validates the effectiveness of the proposed method through comparison with other existing methods. Finally, the conclusions are given in Section VI.

II. PRELIMINARY

Definition 1 (Time Series): A sequence of pairs, $X = \{(t_1, x_1), (t_2, x_2), \dots, (t_N, x_N)\}$, where $t_1 < t_2 < \dots < t_N$, x_i denotes the value of data point at time t_i .

Definition 2 (Trend Turning Points): Given a time series X , each element (t_i, x_i) in X that satisfies the following equation is labeled as a trend turning point:

$$\begin{aligned} \{x_i \in X : x_i \geq x_{i-1} \text{ and } x_i > x_{i+1}\} \\ \cup \{x_i \in X : x_i > x_{i-1} \text{ and } x_i = x_{i+1}\} \\ \cup \{x_i \in X : x_i \leq x_{i-1} \text{ and } x_i < x_{i+1}\} \\ \cup \{x_i \in X : x_i < x_{i-1} \text{ and } x_i = x_{i+1}\} \end{aligned} \quad (1)$$

where $1 < i < N$.

According to (1), we can deduce that in six situations, (t_i, x_i) can be defined as a trend inflection point, as shown

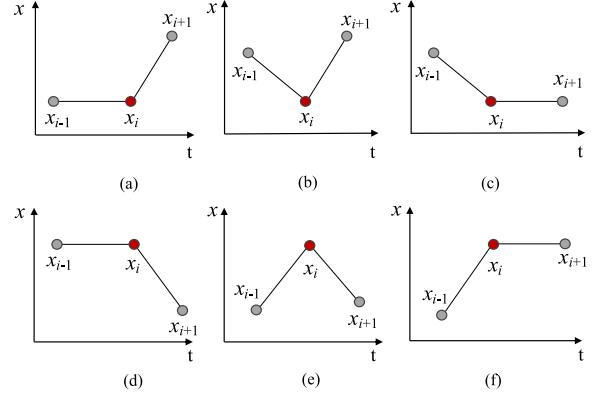


Fig. 1. Illustration of the trend turning points. (a) $x_i = x_{i-1}$ and $x_i < x_{i+1}$. (b) $x_i < x_{i-1}$ and $x_i < x_{i+1}$. (c) $x_i < x_{i-1}$ and $x_i = x_{i+1}$. (d) $x_i = x_{i-1}$ and $x_i > x_{i+1}$. (e) $x_i > x_{i-1}$ and $x_i > x_{i+1}$. (f) $x_i > x_{i-1}$ and $x_i = x_{i+1}$.

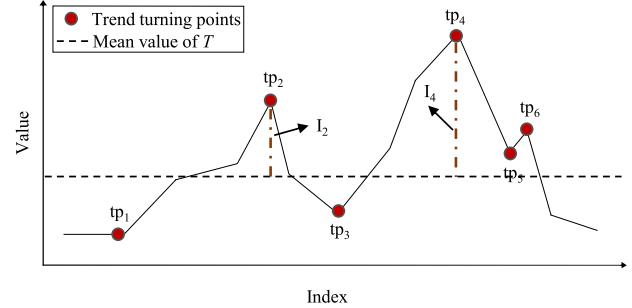


Fig. 2. Example of a time series with indicated trend turning points.

in Fig. 1. The dynamic characteristics can reveal the inherent structure of X . Therefore, this article defines trend turning points, which can fully reflect the volatility information of X . Trend turning points in X can be expressed as $\text{TTP} = \{tp_1, tp_2, \dots, tp_j, \dots, tp_M\}$, where $M \leq N$.

Definition 3 (Important Factor): Given a time series X , an important factor denotes the distance between the value of a trend turning point $tp_j = (t_j, x_j)$ and the mean value of X , as follows:

$$I_j = |\text{MV} - x_j|, \quad j = 1, 2, \dots, M \quad (2)$$

where

$$\text{MV} = \frac{1}{N} \sum_{i=1}^N x_i \quad (3)$$

where I_j refers to the important factor of tp_j and MV refers to the mean value of X .

The important factor is defined to measure the importance of trend turning points in reflecting the volatility characteristics of X . In other words, the larger the value of I_j , the more important the dynamic features contained in tp_j . Fig. 2 depicts a time series, which contains six trend turning points $\text{TTP} = \{tp_1, tp_2, tp_3, tp_4, tp_5, tp_6\}$.

Definition 4 (Maximum Error Tolerance): Maximum error tolerance δ , as shown in the following equation, is defined as a measure to evaluate the goodness of fit for a potential segment. Within a segment, the distance from a point to the fitting line along the vertical direction cannot be greater

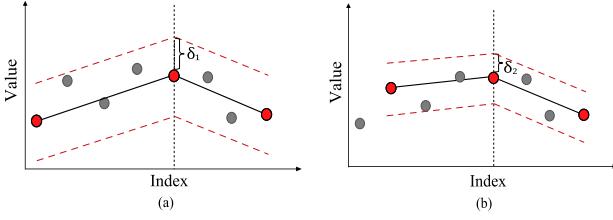


Fig. 3. Linear segments with different values of β . Red markers indicate the segmentation points. (a) β has a relatively large value. (b) β has a relatively small value.

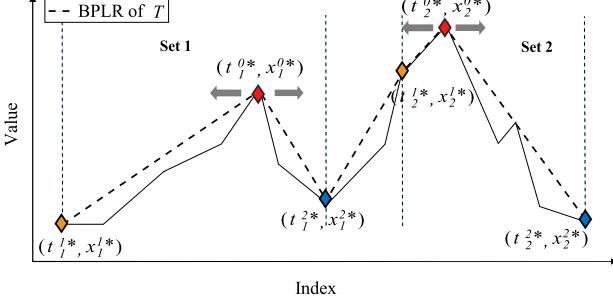


Fig. 4. BPLR of X . The red, blue, and yellow markers indicate the starting, backward, and forward segmentation points, respectively.

than the maximum error tolerance; otherwise, the segment is considered invalid

$$\delta = |\max(X) - \min(X)| \cdot \beta, \quad 0 \leq \beta \leq 1 \quad (4)$$

where β denotes the error tolerance factor, and it is user-specified.

The impact of δ on the segmentation results is illustrated in Fig. 3. The value of the parameter δ is directly associated with the dimensionality reduction of the time series. As the value of δ increases, the dimensionality reduction also increases. Fig. 3 provides an example of linear segments with differing values of δ , where $\beta_1 > \beta_2$ and $\delta_1 > \delta_2$.

III. BIDIRECTIONAL PIECEWISE LINEAR REPRESENTATION FOR TIME SERIES

This section elaborates on the proposed BPLR method for representing a numeric time series, through which dimensionality reduction is realized and a collection of LF functions is formed.

Given a time series $X = \{(t_1, x_1), (t_2, x_2), \dots, (t_N, x_N)\}$, we need to find several sets of segmentation points to complete the representation task, and each set consists of a starting segmentation point (t^{0*}, x^{0*}) , a backward segmentation point (t^{1*}, x^{1*}) , and a forward segmentation point (t^{2*}, x^{2*}) , as mentioned in Fig. 4. Then, connect all the segmentation points with straight lines, as shown in Fig. 4. A BPLR of X can be defined as a group of linear functions. For each set of segmentation points, we first calculate a starting segmentation point and then derive a forward segmentation point and a backward segmentation point based on it. Specific procedures are given in what follows.

A. Calculation of the Starting Segmentation Points

First, identify all trend turning points in X and calculate the important factors contributing to these points. Subsequently,

Algorithm 1 Pseudo-Codes for Calculation Process of a Starting Segmentation Point (t^{0*}, x^{0*})

Input: A time series $X = \{(t_1, x_1), (t_2, x_2), \dots, (t_N, x_N)\}$

Output: TTP_{list} and (t^{0*}, x^{0*})

Process:

```

1:  $i=0$ ;
2:  $V=0$ ;
3:  $TTP=[]$ ;
4:  $TTP_{list}=[]$ ;
5: while  $i \leq N$  do
6:    $V = V + x_i$ ;
7:   if  $(t_i, x_i)$  satisfies any inequalities in (1) then
8:      $TTP=[TTP, (t_i, x_i)]$ ;
9:   end if
10:   $i = i + 1$ ;
11: end while
12:  $MV = V/N$ ;
13: for  $j = 1 : length(TTP)$  do
14:    $tp_j = TTP(j)$ ;
15:    $F_j = VD(MV, tp_j)$ ;
16:   Insert  $tp_j$  into  $TTP_{list}$  in the descending order according to the value of  $F_j$ ;
17: end for
18:  $(t^{0*}, x^{0*}) = TTP_{list}(1)$ ;
19: return  $TTP_{list}$  and  $(t^{0*}, x^{0*})$ ;
```

sort these trend turning points in descending order based on the values of their important factors. In this way, we can obtain a priority list, TTP_{list} , of the trend turning points in X . Finally, select the first point in TTP_{list} as a starting segmentation point (t^{0*}, x^{0*}) . The calculation process of (t^{0*}, x^{0*}) is shown in Algorithm 1.

B. Calculation of the Forward and Backward Segmentation Points

After obtaining the first starting segmentation point, we have $X = \{(t_1, x_1), \dots, (t^{0*}, x^{0*}), \dots, (t_N, x_N)\}$. Next, divide X into two parts, namely $\{(t_1, x_1), \dots, (t^{0*}, x^{0*})\}$ and $\{(t^{0*}, x^{0*}), \dots, (t_N, x_N)\}$. Then, based on (t^{0*}, x^{0*}) , select the first backward segmentation point (t^{1*}, x^{1*}) in $\{(t_1, x_1), \dots, (t^{0*}, x^{0*})\}$ and the first forward segmentation point (t^{2*}, x^{2*}) in $\{(t^{0*}, x^{0*}), \dots, (t_N, x_N)\}$. Here, we give the specific calculation procedures of the forward segmentation point (t^{2*}, x^{2*}) .

Considering a part of X , $\{(t^{0*}, x^{0*}), \dots, (t_q, x_q), \dots\}$, and δ , where δ denotes the maximum error tolerance, we define the upper bound and lower bound for the points containing in $\{(t^{0*}, x^{0*}), \dots, (t_q, x_q)\}$ and we have

$$\begin{cases} \text{up}(p) = (t_p, x_p + \delta) \\ \text{low}(p) = (t_p, x_p - \delta) \end{cases} \quad (5)$$

where $1 \leq p \leq q$.

Let ω be the index of (t^{0*}, x^{0*}) and $l_{\omega,q}$ be the fitting line between (t^{0*}, x^{0*}) and (t_q, x_q) . Then, based on δ , we define the following property.

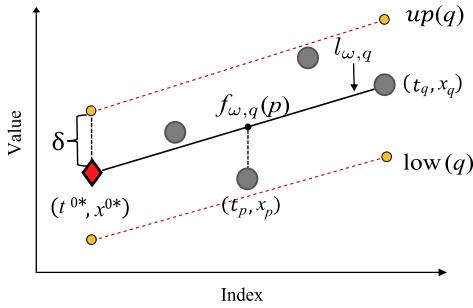
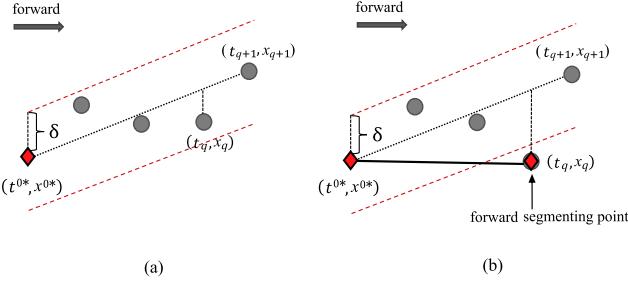


Fig. 5. Illustration of Property 1.

Fig. 6. Segmentation criterion based on δ . (a) (t_p, x_p) satisfies Principle 1. (b) (t_p, x_p) does not satisfy Principle 1.

Property 1: For $\{(t^{0*}, x^{0*}), \dots, (t_p, x_p), \dots, (t_q, x_q)\}$, a part of X , the above sequence can be considered as a segment, iff $|x_p - f_{\omega,q}(t_p)| \leq \delta$.

In other words, if (t_p, x_p) is located between $(f_{\omega,q} - \delta$ and $f_{\omega,q} + \delta)$, there are no segmentation points within $\{(t^{0*}, x^{0*}), \dots, (t_q, x_q)\}$. The linear function $f_{\omega,q}$, representing $l_{\omega,q}$, can be given by $f_{\omega,q} = (x_q - x^{0*}/t_q - t^{0*})(t - t^{0*}) + x^{0*}$. Fig. 5 illustrates Property 1. Based on the above property, we propose a new segmentation criterion as follows.

Segmentation Criterion: For the current segment $\{(t^{0*}, x^{0*}), \dots, (t_q, x_q)\}$, let $up^* = \min_{\omega < p \leq q} up(p)$ and $low^* = \max_{\omega < p \leq q} low(p)$. The newly arriving data point (t_{q+1}, x_{q+1}) can be added into the current segment iff $low^* \leq f_{\omega,q+1}(p) \leq up^*$, where $f_{\omega,q+1}(p)$ denotes the value of function $f_{\omega,q+1}$ at time t_p .

The above segmentation criterion is derived from Property 1

$$\begin{aligned} low^* &\leq f_{\omega,q+1}(p) \leq up^* \\ &\Leftrightarrow \max_{\omega < p \leq q} low(p) \leq f_{\omega,q+1}(p) \leq \min_{\omega < p \leq q} up(p) \\ &\Leftrightarrow low(p) \leq f_{\omega,q+1}(p) \leq up(p) \quad (\omega < p \leq q) \\ &\Leftrightarrow |x_p - f_{\omega,q+1}(t_p)| \leq \delta \quad (\text{Property1}) \\ &\Leftrightarrow \text{add } (t_{q+1}, x_{q+1}) \text{ into the current segment.} \end{aligned} \quad (6)$$

If (t_{q+1}, x_{q+1}) satisfies the proposed segmentation criterion, we add it into the current segment, as Fig. 6(a) depicts. Otherwise, employ (t_q, x_q) as a segmentation point, and denote it as the forward segmentation point (t^{2*}, x^{2*}) , as depicted in Fig. 6(b). Meanwhile, one can calculate the backward segmentation point (t^{1*}, x^{1*}) in the same way.

After obtaining the backward segmentation point and forward segmentation point, the first set of segmentation points is obtained, namely $\{(t^{1*}, x^{1*}), (t^{0*}, x^{0*}), (t^{2*}, x^{2*})\}$. Let $s1 = \{(t^{1*}, x^{1*}), \dots, (t^{0*}, x^{0*}), \dots, (t^{2*}, x^{2*})\}$, where $s1 \in X$,

Algorithm 2 Pseudo-Codes for Calculating All the Segmentation Points in X

Input: A time series X and TTP_{list}

Output: All the segmentation points in X

Process:

```

1:  $SP = []$ ;
2: while  $\sim isempty(TTP_{list})$  do
3:    $(t^{0*}, x^{0*}) = TTP_{list}(1)$ ;
4:   Calculate  $(t^{1*}, x^{1*})$  and  $(t^{2*}, x^{2*})$  based on  $(t^{0*}, x^{0*})$  and the proposed segmentation criterion;
5:    $SP' = [(t^{0*}, x^{0*}), (t^{1*}, x^{1*}), (t^{2*}, x^{2*})]$ ;
6:    $SP = [SP, SP']$ ;
7:   for  $j = 1 : length(TTP_{list})$  do
8:     if  $TTP_{list}(j) \in s1$  then
9:        $TTP_{list}(j) = []$ ;
10:    end if
11:   end for
12: end while
13: return  $SP$ ;
```

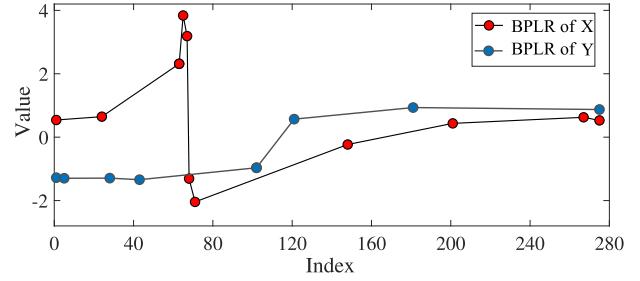


Fig. 7. Two LF time series.

remove the trend turning points within $s1$ from TTP_{list} . Then, select the trend turning point with the largest F -value in the updated TTP_{list} and employ this trend turning point as a new starting segmentation point. Next, calculate the second forward segmentation point and backward segmentation point. Then, repeat the above procedures till TTP_{list} is empty and connect all the segmentation points to get a BPLR of X . The specific calculation process for all the segmentation points in X is shown in Algorithm 2.

IV. HIERARCHICAL CLUSTERING FOR LF TIME SERIES BASED ON LF-DTW DISTANCE

This section elaborates on the clustering process for the obtained LF Time Series. More specifically, Section IV-A introduces the calculation process of the distance between two fitting functions. In Section IV-B, we propose a new distance, LF-DTW distance, and employ it to measure the similarity between any two representation time series. Section IV-C gives the clustering process of the obtained LF time series based on the proposed distance measure.

A. Distance Calculation of Linear Functions

Considering two time series, X and Y , we arrive at the two corresponding LF time series after data representation,

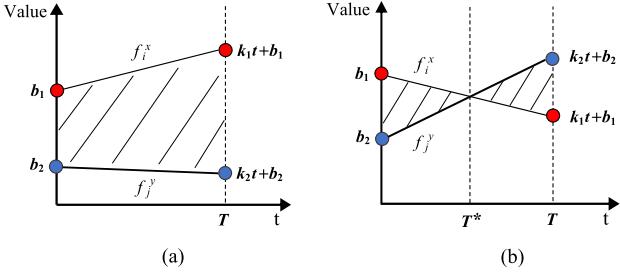


Fig. 8. Distance between two equal-size linear functions. (a) Two functions do not intersect. (b) Two functions intersect.

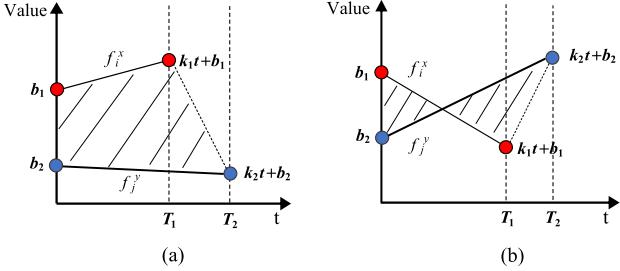


Fig. 9. Distance between two unequal-size linear functions. a) Two functions do not intersect. (b) Two functions intersect.

as shown in the following equations. Fig. 7 gives an example of the two LF time series, LF_X and LF_Y

$$\text{LF}_X = \{f_1^x, f_2^x, \dots, f_i^x, \dots, f_{n_x}^x\} \quad (7)$$

$$\text{LF}_Y = \{f_1^y, f_2^y, \dots, f_j^y, \dots, f_{n_y}^y\} \quad (8)$$

where n_x and n_y denote the number of linear functions in LF_X and LF_Y , respectively. Here, we propose a new distance to measure the similarity between f_i^x and f_j^y . Since the time interval of the linear functions in LF_X and LF_Y may be unequal, the distance between f_i^x and f_j^y is considered in the following two cases. Note that the linear functions in LF_X and LF_Y are considered left-aligned in the similarity measurement process.

1) Distance Measure of Two Equal-Size Linear Functions: As depicted in Fig. 8, the distance between two equal-size linear functions is denoted as the area of the shaded part, which is calculated as

$$d_1(f_i^x, f_j^y) = \int_0^T |f_i^x - f_j^y| dt. \quad (9)$$

2) Distance Measure of Two Unequal-Size Linear Functions: Considering two unequal-size linear functions, as shown in Fig. 9, (8) is no longer applicable. Here, we consider f_j^y as a joint of two functions $f_{j,1}^y$ and $f_{j,2}^y$. It is obvious that the region between f_i^x and f_j^y consists of two parts, as Fig. 10 illustrates.

Considering part one, we calculate the distance between f_i^x and $f_{j,1}^y$, where f_i^x and $f_{j,1}^y$ have the same size T_1 . Considering part two, we calculate the distance between f_i^x and $f_{j,2}^y$, where f_i^x and $f_{j,2}^y$ are with the same size $T_2 - T_1$. Thus, the distance between two unequal-size linear functions is defined as

$$\begin{aligned} d_2(f_i^x, f_j^y) &= d_1(f_i^x, f_{j,1}^y) + d_1(f_i^x, f_{j,2}^y) \\ &= \int_0^{T_1} |f_i^x - f_{j,1}^y| dt + \int_{T_1}^{T_2} |f_i^x - f_{j,2}^y| dt \end{aligned} \quad (10)$$

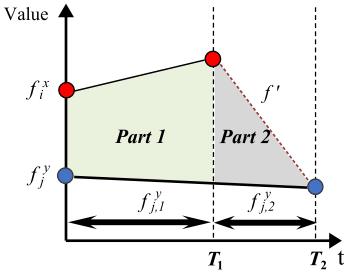


Fig. 10. Partition of the region.

where $d_1(\cdot, \cdot)$ denotes the distance between two equal-size linear functions.

B. Similarity Measurement for LF Time Series

After calculating the distance between two linear functions, we need to figure out a method to measure the similarity of two LF time series. Since the segmentation of the original time series is related to its dynamic characteristics, the number of linear functions in different LF time series may not be the same. Based on DTW distance, we propose LF-DTW distance in this article, which can measure the distance between two LF time series with different sizes.

To start, we briefly review the DTW distance [36], [37], [38], a common metric in time-series clustering. In the field of time-series analysis, DTW is a frequently utilized method for time-series alignment. Given two time series, $X = \{(t_1, x_1), (t_2, x_2), \dots, (t_N, x_{N_1})\}$ and $Y = \{(t_1, y_1), (t_2, y_2), \dots, (t_N, y_{N_2})\}$, DTW introduces a variable $w_{i,j}$ in the following equations and then uses recursion to match the data points in X and Y with the shortest distance:

$$w_{i,j} = \begin{cases} \infty, & (i = 0 \text{ or } j = 0) \text{ and } i \neq j \\ 0, & i = j = 0 \end{cases} \quad (11)$$

$$w_{i,j} = d((t_i, x_i), (t_j, y_j)) + \min \begin{cases} w_{i-1,j} \\ w_{i-1,j-1} \\ w_{i,j-1} \end{cases} \quad (12)$$

where $1 \leq i \leq N_1$, $1 \leq j \leq N_2$, and $d((t_i, x_i), (t_j, y_j))$ denotes the distance between data points (t_i, x_i) and (t_j, y_j) . Following the mentioned above process, an accumulated matrix $W = [w_{i,j}] \in R^{N_1 \times N_2}$ is computed, and the element in W , w_{N_1, N_2} , denotes the DTW distance between time series X and Y .

Next, we extend the DTW distance to measure the similarity for the LF time series. Given two LF time series, $\text{LF}_X = \{f_1^x, f_2^x, \dots, f_{n_x}^x\}$ and $\text{LF}_Y = \{f_1^y, f_2^y, \dots, f_{n_y}^y\}$, the corresponding accumulated distance matrices $W^{\text{LF}} = [w_{i,j}^{\text{LF}}] \in R^{n_x \times n_y}$ is computed as

$$w_{i,j}^{\text{LF}} = \begin{cases} \infty, & (i = 0 \text{ or } j = 0) \text{ and } i \neq j \\ 0, & i = j = 0 \end{cases} \quad (13)$$

$$w_{i,j}^{\text{LF}} = d(f_i^x, f_j^y) + \min \begin{cases} w_{i-1,j}^{\text{LF}} \\ w_{i-1,j-1}^{\text{LF}} \\ w_{i,j-1}^{\text{LF}} \end{cases} \quad (14)$$

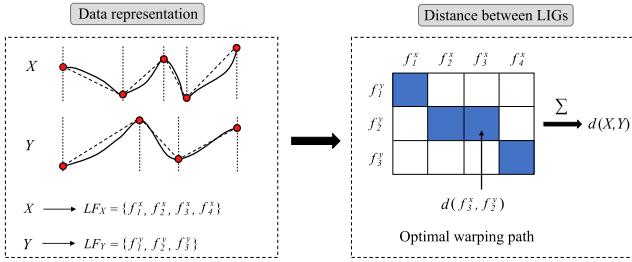


Fig. 11. Framework of the proposed similarity measurement method.

where $1 \leq i \leq n_x$, $1 \leq j \leq n_y$, and $d(f_i^x, f_j^y)$ is the distance between f_i^x and f_j^y . $w(n_x, n_y)$ in W^{LF} denotes the LF-DTW distance between LF_X and LF_Y , namely $d(X, Y)$. Fig. 11 presents a conceptual illustration of the proposed distance measure.

C. Hierarchical Clustering Based on LF-DTW Distance

With the distance measure, LF-DTW distance, we can complete the hierarchical clustering for the formed LF time series. Given a set of LF time series $\text{LFs} = \{\text{LF}_{X_1}, \text{LF}_{X_2}, \dots, \text{LF}_{X_n}\}$, we can obtain a distance matrix $D_{n \times n}$

$$D_{n \times n} = \begin{vmatrix} 0 & d(X_1, X_2) & \cdots & d(X_1, X_n) \\ d(X_2, X_1) & 0 & \cdots & d(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ d(X_n, X_1) & d(X_n, X_2) & \cdots & 0 \end{vmatrix} \quad (15)$$

where the elements in $D_{n \times n}$ denotes the LF-DTW distance between two elements in LFs.

Then, we can realize the hierarchical clustering according to $D_{n \times n}$. First, let each LF time series in LFs be a separate cluster, forming n initial clusters. Next, select and merge the two clusters with the highest similarity until there is only one cluster. Finally, a hierarchical structure of LF time series is formed, with the merging process mentioned above. The clustering results can be obtained when the number of clusters is specified as ranging from 2 to C .

V. EXPERIMENTS, RESULTS, AND DISCUSSION

In this section, based on the proposed BPLR data representation method and LF-DTW distance-based similarity measurement method, we carry out hierarchical clustering on real-world data, which is obtained from the UCR time-series database [40]. The clustering results of major existing methods are also depicted to show the superiority of the proposed clustering method.

A. Parameter Discussion

To start, the parameter in the BPLR method needs to be discussed, which is the error tolerance factor β in (4). From Definition 4, we can infer that the larger the value of β , the better the dimensionality reduction effect of the BPLR method. However, when the dimensionality reduction is too aggressive, the fitting error between the original data and its BPLR may increase. Therefore, we need to find

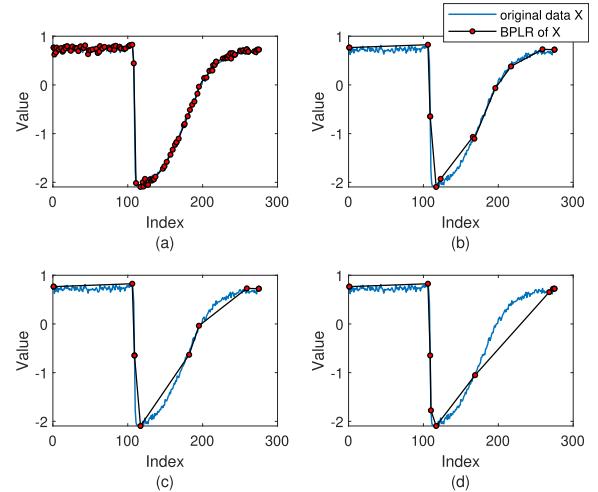


Fig. 12. BPLR of a time series from Trace Dataset with different β . (a) $\beta = 1\%$. (b) $\beta = 5\%$. (c) $\beta = 10\%$. (d) $\beta = 15\%$.

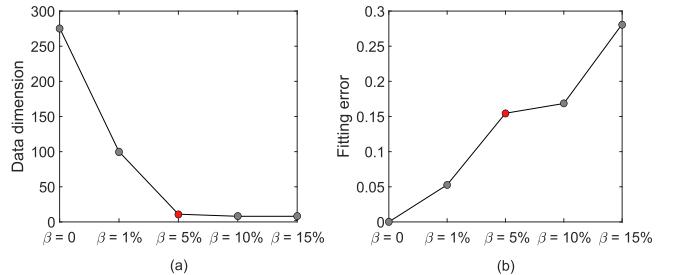


Fig. 13. Dimensionality reduction effect and fitting error of the selected time series. (a) Dimensionality reduction effect. (b) Fitting error.

a balance between enhancing the dimensionality reduction effect and controlling the reduction representation and fitting error of the original time series, to ensure that the fitting error does not become excessive. Given a time series $X = \{(t_1, x_1), (t_2, x_2), \dots, (t_N, x_N)\}$, the fitting error ϵ [17] between the original data and its BPLR can be computed as

$$\epsilon = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - x'_i)^2} \quad (16)$$

where x_i and x'_i refer to the original sequence value and fitting sequence value at the same time t_i , respectively.

A time series from Trace Dataset with different values of β is shown in Fig. 12. Fig. 13 presents the dimensionality reduction effect and fitting error of the selected time series. It can be observed that when $\beta = 5\%$, a good dimensionality reduction effect can be achieved, and at the same time, the fitting error between the BPLR after dimensionality reduction and the original data is also relatively small.

Table I compares the results of dimensionality reduction effects and fitting errors on four datasets, using different values of β . It can be observed that when $\beta = 5\%$, good dimensionality reduction effects were achieved on all four datasets, while maintaining relatively low fitting errors. Therefore, we chose $\beta = 5\%$ for the subsequent clustering experiments. It is worth noting that the value of β can be determined based on user requirements. If the user desires a better dimensionality

TABLE I
DIMENSIONALITY REDUCTION EFFECTS AND FITTING ERRORS
BASED ON DIFFERENT β

Dataset	value of β	data dimension	fitting error
Trace	$\beta = 0$	276	0
	$\beta = 1\%$	61	0.1022
	$\beta = 5\%$	12	0.1024
	$\beta = 10\%$	10	0.1768
	$\beta = 15\%$	9	0.2026
Gesture	$\beta = 0$	129	0
	$\beta = 1\%$	29	1.8549
	$\beta = 5\%$	15	2.2089
	$\beta = 10\%$	9	3.7826
	$\beta = 15\%$	9	4.0345
Mixed Shapes	$\beta = 0$	1025	0
	$\beta = 1\%$	161	0.0502
	$\beta = 5\%$	21	0.1230
	$\beta = 10\%$	14	0.2776
	$\beta = 15\%$	12	0.3349
ECG	$\beta = 0$	141	0
	$\beta = 1\%$	40	0.1473
	$\beta = 5\%$	13	0.2346
	$\beta = 10\%$	12	0.3518
	$\beta = 15\%$	12	0.4550

reduction effect and can tolerate a certain amount of error, the value of β can also be increased.

B. Experiment on the Trace Dataset

The first experiment operates on the Trace Dataset. Fig. 14 shows a dataset containing 12 times series of trace, which are clustered into four classes, namely $\{1, 2, 3\}$, $\{4, 5, 6\}$, $\{7, 8, 9\}$, $\{10, 11, 12\}$. In Fig. 14, blue lines refer to the original time series, black lines refer to its BPLR, and the red markers refer to segmentation points. It can be seen that the proposed BPLR method can capture the main essence of original data and provide a concise description of it. Moreover, the dimensionality reduction of the original time series is realized.

The clustering results of the proposed method and other existing methods are shown in Fig. 15, where the horizontal axis represents the numbering of time series, and the vertical axis represents the distance value. More specifically, Fig. 15(a) presents the clustering result based on the proposed method. In the Trace Dataset, time series from the same class exhibit similar structures. However, their peak positions can vary, making direct point-to-point distance calculations on the

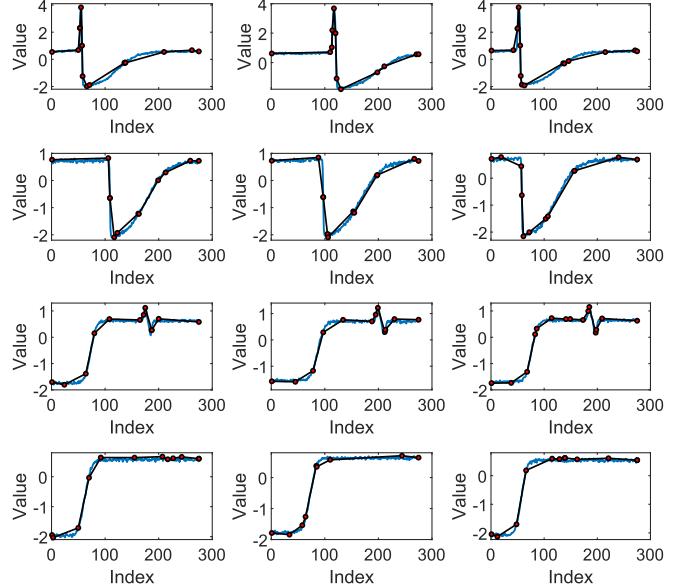


Fig. 14. Trace time series and its BPLR. Blue lines indicate the original time series, black lines indicate its BPLR, and red markers indicate the segmentation points.

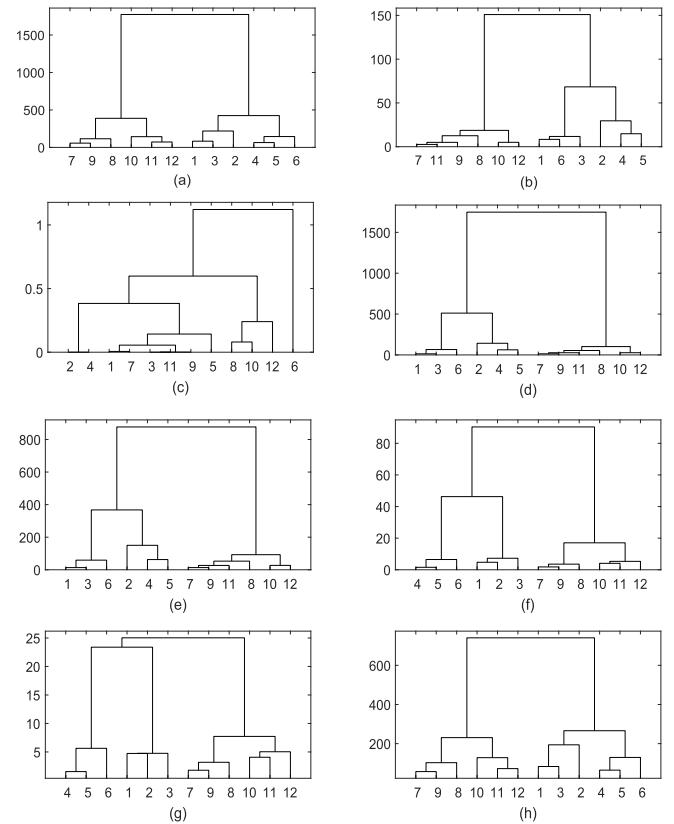


Fig. 15. Clustering results of the Trace Dataset. (a) Proposed method. (b) Euclidean distance. (c) Hausdorff distance. (d) DTW distance [36]. (e) Clustering method in [18]. (f) Clustering method in [19]. (g) Clustering method in [20]. (h) Clustering method in [39].

original series unreliable for accurate clustering. Fig. 15(b)–(d) showcase the clustering results using the Euclidean distance, Hausdorff distance, and DTW distance [36], respectively.

TABLE II
CLUSTERING RESULTS OF THE REAL-WORLD DATASET

Dataset	Correct clusters	Method	Clustering result
Trace	$\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}, \{10, 11, 12\}\}$	Proposed method	$\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}, \{10, 11, 12\}\}$
		Euclidean distance	$\{\{1, 3, 6\}, \{2\}, \{4, 5\}, \{7, 8, 9, 10, 11, 12\}\}$
		Hausdorff distance	$\{\{1, 3, 4, 5, 7, 8\}, \{2\}, \{6\}, \{9, 10, 11, 12\}\}$
		DTW distance [36]	$\{\{1, 3, 6\}, \{2, 4, 5\}, \{7, 9, 11\}, \{10, 12\}\}$
		Method in [18]	$\{\{1, 3, 6\}, \{2\}, \{4, 5\}, \{7, 8, 9, 10, 11, 12\}\}$
		Method in [19]	$\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}, \{10, 11, 12\}\}$
		Method in [20]	$\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}, \{10, 11, 12\}\}$
Gesture	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}, \{11, 12\}\}$	Method in [39]	$\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}, \{10, 11, 12\}\}$
		Proposed method	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}, \{11, 12\}\}$
		Euclidean distance	$\{\{1, 2, 10\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9\}, \{11, 12\}\}$
		Hausdorff distance	$\{\{1, 5\}, \{2\}, \{3, 8\}, \{4, 6\}, \{7\}, \{9, 10, 11, 12\}\}$
		DTW distance [36]	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}, \{11, 12\}\}$
		Method in [18]	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}, \{11, 12\}\}$
		Method in [19]	$\{\{1, 2, 10\}, \{3, 4\}, \{5\}, \{6\}, \{7, 8, 11, 12\}, \{9\}\}$
Mixed Shapes	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}\}$	Method in [20]	$\{\{1, 2, 10\}, \{3, 4\}, \{5\}, \{6\}, \{7, 8, 11, 12\}, \{9\}\}$
		Method in [39]	$\{\{1, 2, 9\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{10\}, \{11, 12\}\}$
		Proposed method	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}\}$
		Euclidean distance	$\{\{1, 2\}, \{3, 4\}, \{5, 6, 8\}, \{7\}, \{9, 10\}\}$
		Hausdorff distance	$\{\{1\}, \{2, 4, 7\}, \{3, 6, 8\}, \{5\}, \{9, 10\}\}$
		DTW distance [36]	$\{\{1, 2\}, \{3, 4\}, \{5, 6, 7\}, \{8\}, \{9, 10\}\}$
		Method in [18]	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}\}$
ECG	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$	Method in [19]	$\{\{1, 2, 8\}, \{3, 4\}, \{5, 6\}, \{7\}, \{9, 10\}\}$
		Method in [20]	$\{\{1\}, \{2, 8\}, \{3, 4\}, \{5, 6, 7\}, \{9, 10\}\}$
		Method in [39]	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}\}$
		Proposed method	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$
		Euclidean distance	$\{\{1\}, \{2, 7\}, \{3, 4, 8\}, \{5, 6\}\}$
		Hausdorff distance	$\{\{1, 7, 8\}, \{2\}, \{3, 5\}, \{4, 6\}\}$
		DTW distance [36]	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$

Meanwhile, the results based on different time-series clustering methodologies proposed in [18], [19], [20], and [39] are illustrated in Fig. 15(e)–(h). Among these, accurate clustering results are observed in Fig. 15(a) and (f)–(h).

C. Experiment on the Gesture Dataset

The Gesture Dataset is collected with the smartwatch, which is used to experiment in this section. We can see from Fig. 16 that there are six classes of time series, and BPLR still achieves a good effect in data representation. The structural differences between the 12 time series in Fig. 16 are not significant, making it more difficult to distinguish them, and the correct clustering result should be $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}, \{11, 12\}\}$. The clustering results of the proposed method and other existing methods are presented in Fig. 17. It is observed that the proposed method, the DTW distance-based method, and the clustering

method in [18] achieve accurate clustering results, as shown in Fig. 17(a), (d), and (e), verifying the effectiveness of our method. Moreover, the DTW distance-based method and the method in [18] are realized by using the original time series, and, therefore, compared to them, our method has lower computational complexity.

D. Experiment on the Mixed Shapes Dataset

In this experiment, we use the Mixed Shapes Dataset to evaluate the performance of the proposed method. The original dataset and its BPLR are depicted in Fig. 18, and it can be seen that the BPLR preserves the dynamic characteristics of the original data. At the same time, the dimensionality of the original time series has significantly decreased. According to the structure of the dataset, the correct clustering result is $\{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}\}$. Fig. 19 presents the clustering results of the proposed method and other existing

TABLE III
RESULTS OF COMPARATIVE EXPERIMENTS

Dataset	Correct clusters	Method	Clustering result
Synthetic Control	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}	Proposed method	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Euclidean distance	{\{1, 2, 3\}, \{4, 5, 6, 7, 8\}, \{9\}}
		Hausdorff distance	{\{1, 3, 4, 5, 7, 8\}, \{2\}, \{6\}, \{9\}}
		DTW distance [36]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [18]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [19]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [20]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [39]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
BME	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}	Proposed method	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Euclidean distance	{\{1, 2, 3, 9\}, \{4, 5, 6\}, \{7, 8\}}
		Hausdorff distance	{\{1, 2, 3\}, \{4, 5, 6, 7, 8\}, \{9\}}
		DTW distance [36]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [18]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [19]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [20]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [39]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
Mixed Shapes Regular	{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}}	Proposed method	{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{8\}, \{7, 9, 10\}}
		Euclidean distance	{\{1, 2\}, \{3, 4\}, \{5, 6, 8\}, \{7\}, \{9, 10\}}
		Hausdorff distance	{\{1\}, \{2, 4\}, \{3, 5, 6, 8\}, \{7\}, \{9, 10\}}
		DTW distance [36]	{\{1, 2\}, \{3, 4\}, \{5, 6, 7\}, \{8\}, \{9, 10\}}
		Method in [18]	{\{1, 2\}, \{3, 4, 6\}, \{5, 7\}, \{8\}, \{9, 10\}}
		Method in [19]	{\{1, 2\}, \{3, 4, 6\}, \{5, 8\}, \{7\}, \{9, 10\}}
		Method in [20]	{\{1, 2\}, \{3, 4\}, \{5, 6, 7\}, \{8\}, \{9, 10\}}
		Method in [39]	{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}, \{9, 10\}}
Insect EPG	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}	Proposed method	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Euclidean distance	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Hausdorff distance	{\{1, 2, 3, 4, 5, 6\}, \{7, 8\}, \{9\}}
		DTW distance [36]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [18]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [19]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [20]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [39]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
UMD	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}	Proposed method	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Euclidean distance	{\{1, 3, 7, 8, 9\}, \{4, 5, 6\}, \{2\}}
		Hausdorff distance	{\{1, 3, 5, 6, 7, 8\}, \{2, 4\}, \{9\}}
		DTW distance [36]	{\{1, 3\}, \{2, 7, 8, 9\}, \{4, 5, 6\}}
		Method in [18]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [19]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}
		Method in [20]	{\{1, 2, 3, 8\}, \{4, 5, 6\}, \{7, 9\}}
		Method in [39]	{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}}

methods. As shown in Fig. 18, the third class and the fourth class of time series are quite similar. One can see that the proposed method, the clustering method in [18], and the clustering method in [39] achieve accurate clustering results, as shown in Fig. 17(a), (e), and (h), and this validates the effectiveness and robustness of the proposed method.

E. Experiment on the Electrocardiograph Dataset

In this section, we use the Electrocardiograph (ECG) Dataset to conduct the comparative experiment. The original ECG Dataset and its BPLR are shown in Fig. 20, and we can see that there are four classes of time series in the dataset. The correct clustering result is {\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}}, according to the structure of the eight time series in Fig. 20.

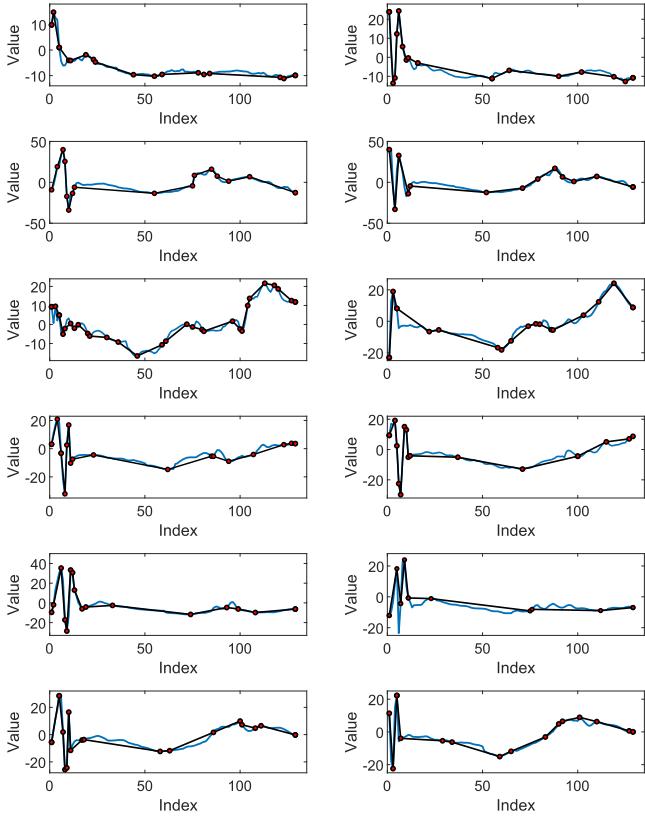


Fig. 16. Gesture time series and its BPLR. Blue lines indicate the original time series, black lines indicate its BPLR, and red markers indicate the segmentation points.

We can also see that BPLR still has a good performance in capturing the essential characteristics of the original time series.

The eight time series are very similar in the first half, and peaks and valleys appear at different positions in the second half, making them difficult to distinguish. Fig. 21 presents the clustering results of the proposed method and other existing methods. We can see that the proposed method, the clustering method in [18], [19], and [20] achieve accurate clustering results.

F. Summary of Experimental Studies

The comparative experimental results from these four groups of real-world data are collected in Table II. Compared with the other three distance measures and four clustering methods mentioned in the above experiments, the proposed method achieves a better clustering accuracy. It is found that the proposed method consistently produces clustering results that match the natural structure of the given dataset, while other methods fail in at least one experiment. It can be seen that the proposed BPLR method can accurately preserve the core features of the original data while reducing dimensionality. When evaluated against classic distance metrics such as Euclidean distance, Hausdorff distance, and DTW distance, the LF-DTW distance stands out as being more stable and effective. Additionally, our method, which integrates BPLR with LF-DTW distance, demonstrates superior clustering accuracy when contrasted with the clustering methods detailed in [18], [19], [20], and [39].

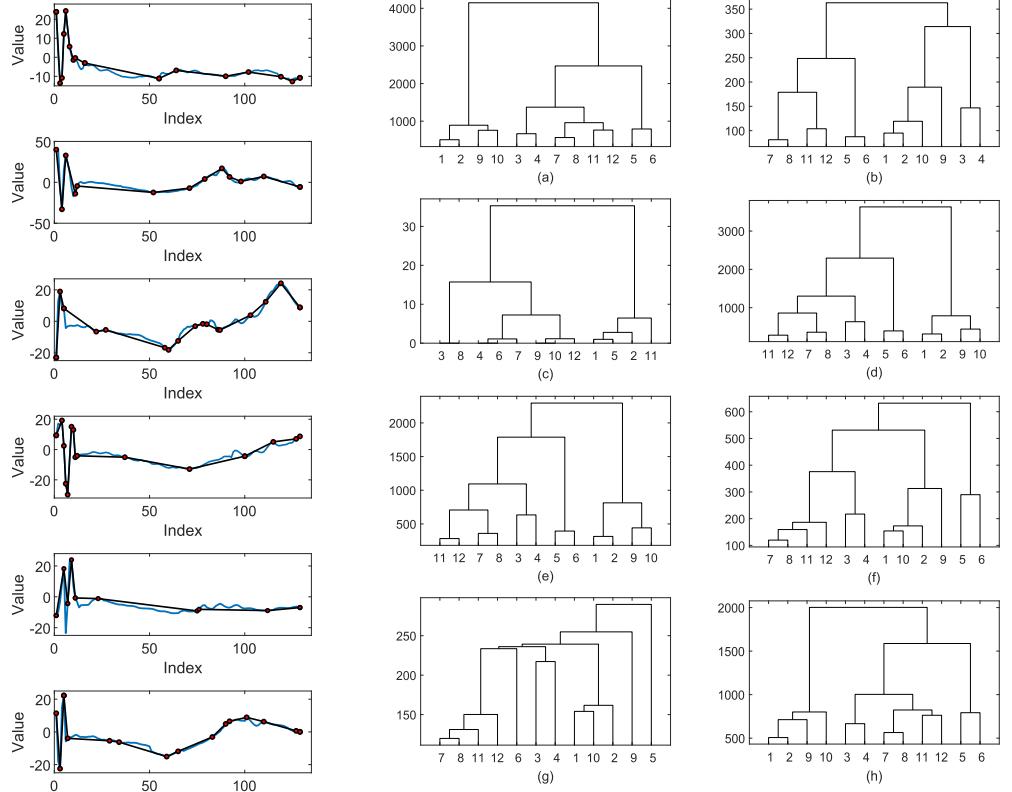


Fig. 17. Clustering results of the Gesture Dataset. (a) Proposed method. (b) Euclidean distance. (c) Hausdorff distance. (d) DTW distance [36]. (e) Clustering method in [18]. (f) Clustering method in [19]. (g) Clustering method in [20]. (h) Clustering method in [39].

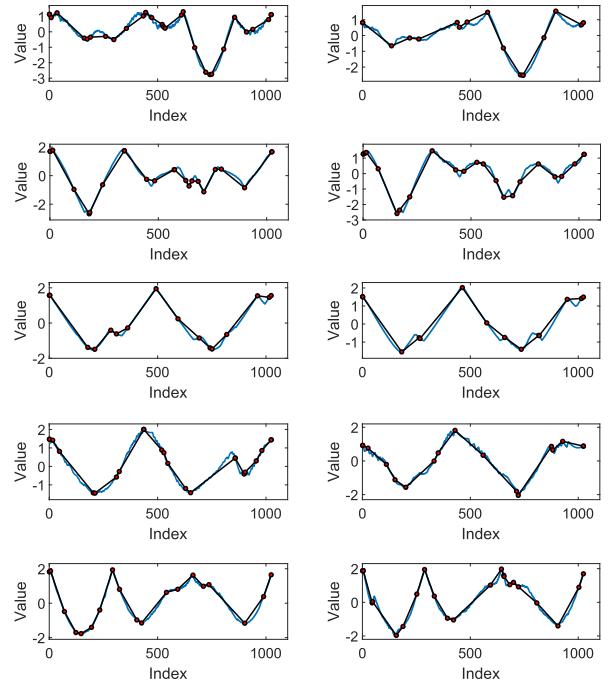


Fig. 18. Mixed shape time series and its BPLR. Blue lines indicate the original time series, black lines indicate its BPLR, and red markers indicate the segmentation points.

Five more real-world datasets are used for comparative experiments to support the above conclusions, and the corresponding experimental results are summarized in Table III.

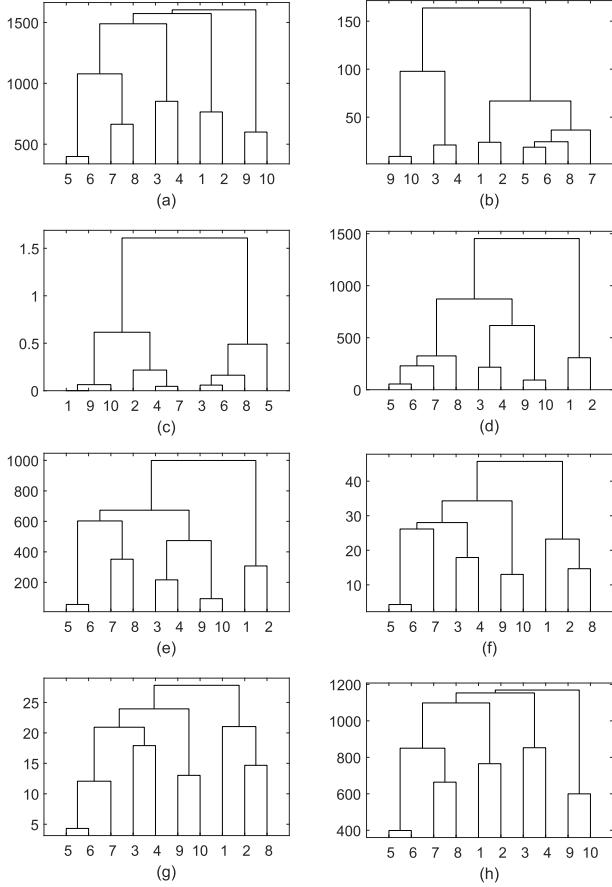


Fig. 19. Clustering results of the Mixed Shapes Dataset. (a) Proposed method. (b) Euclidean distance. (c) Hausdorff distance. (d) DTW distance [36]. (e) Clustering method in [18]. (f) Clustering method in [19]. (g) Clustering method in [20]. (h) Clustering method in [39].

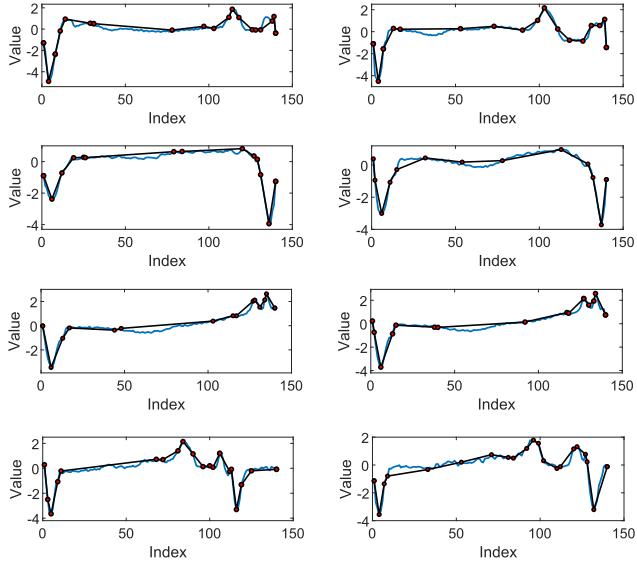


Fig. 20. ECG time series and its BPLR. Blue lines indicate the original time series, black lines indicate its BPLR, and red markers indicate the segmentation points.

We consolidate the experimental results from Tables II and III into Fig. 22. As seen in Fig. 22, the method introduced in this article consistently stands out, demonstrating the least number of clustering errors across the tested datasets, with

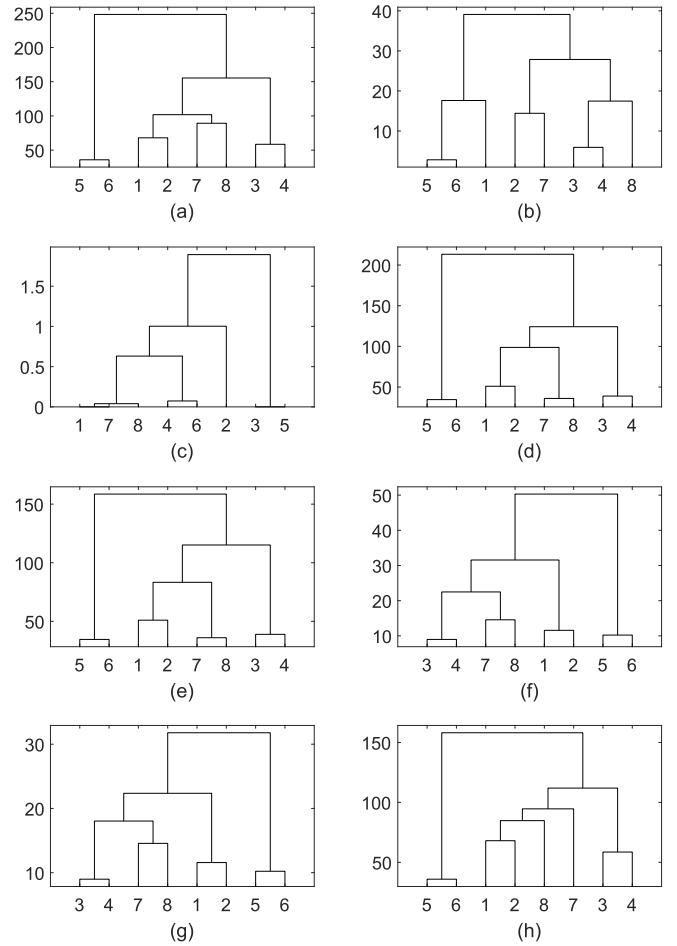


Fig. 21. Clustering results of the ECG Dataset. (a) Proposed method. (b) Euclidean distance. (c) Hausdorff distance. (d) DTW distance [36]. (e) Clustering method in [18]. (f) Clustering method in [19]. (g) Clustering method in [20]. (h) Clustering method in [39].

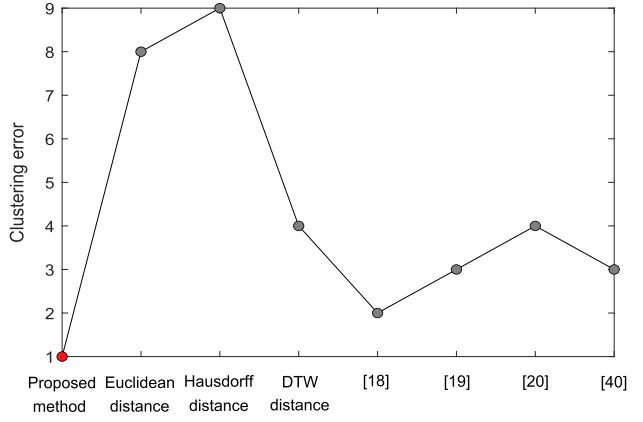


Fig. 22. Clustering errors of different methods.

only one error. Conversely, other methods register clustering errors in at least two datasets. Specifically, in the Mixed Shapes Regular Dataset, the proposed method registers a single error, yet it achieves flawless results in the other datasets. A comprehensive comparison in Fig. 22 clearly indicates the superior performance of the proposed method over the other seven techniques.

VI. CONCLUSION

In this article, we propose a novel data representation method of time series and validate it in a hierarchical clustering task. First, using the proposed BPLR method, the original time series is represented by a collection of LF functions. The achieved LF time series with much lower dimensionality carries a concise description of the original time series. Then, a new distance measure, LF-DTW distance, is proposed to calculate the distance between any two LF time series, which provides good matching results even if the two time series are distorted in the time scale. Finally, hierarchical clustering of the obtained LF time series is realized based on the proposed LF-DTW distance. The proposed BPLR method can fully maintain the dynamic characteristics of time series and help to reduce the computing overhead. Experimental results using real-world data validate the effectiveness and advantages of the proposed approach. Compared to other existing methods, the proposed approach shows consistent superior effectiveness.

The proposed BPLR method provides a way of depicting the inherent structure of the original time series. In this study, we focus on realizing the hierarchical clustering of the LF time series. It is a promising direction to generalize the K-means algorithm and the fuzzy C-means algorithm to make them suitable for the LF time series. Meanwhile, we plan to extend our method to analyze multivariate time series, thereby broadening its applicability in various fields.

REFERENCES

- [1] P. Ferrari, A. Flammini, E. Sisinni, S. Rinaldi, D. Brandão, and M. S. Rocha, "Delay estimation of industrial IoT applications based on messaging protocols," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 9, pp. 2188–2199, Sep. 2018.
- [2] S. Wang, C. Li, and A. Lim, "A model for non-stationary time series and its applications in filtering and anomaly detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
- [3] S. Takahashi, Y. Chen, and K. Tanaka-Ishii, "Modeling financial time-series with generative adversarial networks," *Phys. A, Stat. Mech. Appl.*, vol. 527, Aug. 2019, Art. no. 121261.
- [4] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," *Appl. Soft Comput.*, vol. 90, May 2020, Art. no. 106181.
- [5] L. Bai, L. Cui, Z. Zhang, L. Xu, Y. Wang, and E. R. Hancock, "Entropic dynamic time warping kernels for co-evolving financial time series analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 4, pp. 1808–1822, Apr. 2023.
- [6] Y. Ji, G. Zhou, L. Wang, S. Wang, and Z. Li, "Identifying climate risk causing maize (*Zea mays* L.) yield fluctuation by time-series data," *Natural Hazards*, vol. 96, no. 3, pp. 1213–1222, Apr. 2019.
- [7] S. Mehdizadeh, F. Ahmadi, A. Danandeh Mehr, and M. J. S. Safari, "Drought modeling using classic time series and hybrid wavelet-gene expression programming models," *J. Hydrol.*, vol. 587, Aug. 2020, Art. no. 125017.
- [8] Z. Şen, "Jump point identification in hydro-meteorological time series by crossing methodology," *Theor. Appl. Climatol.*, vol. 144, nos. 1–2, pp. 769–777, Apr. 2021.
- [9] J. P. Karmy and S. Maldonado, "Hierarchical time series forecasting via support vector regression in the European travel retail industry," *Expert Syst. Appl.*, vol. 137, pp. 59–73, Dec. 2019.
- [10] T. Wang, H. Leung, J. Zhao, and W. Wang, "Multiseries featural LSTM for partial periodic time-series prediction: A case study for steel industry," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 5994–6003, Sep. 2020.
- [11] F. Nilsson, M.-R. Bouguila, and T. Rögnvaldsson, "Practical joint human-machine exploration of industrial time series using the matrix profile," *Data Mining Knowl. Discovery*, vol. 37, no. 1, pp. 1–38, Jan. 2023.
- [12] W. Mao, J. Liu, J. Chen, and X. Liang, "An interpretable deep transfer learning-based remaining useful life prediction approach for bearings with selective degradation knowledge fusion," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–16, 2022.
- [13] Y. Wu et al., "OPR-miner: Order-preserving rule mining for time series," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 6, 2023, doi: 10.1109/TKDE.2022.3224963.
- [14] H. Li and Z. Liu, "Multivariate time series clustering based on complex network," *Pattern Recognit.*, vol. 115, Jul. 2021, Art. no. 107919.
- [15] R. R. Sharma, M. Kumar, S. Maheshwari, and K. P. Ray, "EVDHM-ARIMA-based time series forecasting model and its application for COVID-19 cases," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.
- [16] W. Li, X. Wang, H. Han, and J. Qiao, "A PLS-based pruning algorithm for simplified long-short term memory neural network in time series prediction," *Knowl.-Based Syst.*, vol. 254, Oct. 2022, Art. no. 109608.
- [17] X. Kong, Y. Bi, and D. H. Glass, "Detecting anomalies in sequential data augmented with new features," *Artif. Intell. Rev.*, vol. 53, no. 1, pp. 625–652, Jan. 2020.
- [18] Z. Yang, S. Jiang, F. Yu, W. Pedrycz, H. Yang, and Y. Hao, "Linear fuzzy information-granule-based fuzzy C-means algorithm for clustering time series," *IEEE Trans. Cybern.*, early access, Jul. 13, 2022, doi: 10.1109/TCYB.2022.3184999.
- [19] H. Guo, L. Wang, X. Liu, and W. Pedrycz, "Information granulation-based fuzzy clustering of time series," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 6253–6261, Dec. 2021.
- [20] Q. Zhang et al., "A method for measuring similarity of time series based on series decomposition and dynamic time warping," *Appl. Intell.*, vol. 53, pp. 6448–6463, Mar. 2023.
- [21] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowl. Inf. Syst.*, vol. 3, no. 3, pp. 915–937, 2001.
- [22] E. Keogh, J. Lin, and A. Fu, "HOT SAX: Efficiently finding the most unusual time series subsequence," in *Proc. 5th IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2005, pp. 226–233.
- [23] Y. Wan, X. Gong, and Y.-W. Si, "Effect of segmentation on financial time series pattern matching," *Appl. Soft Comput.*, vol. 38, pp. 346–359, Jan. 2016.
- [24] W. Pedrycz and W. Homenda, "Building the fundamentals of granular computing: A principle of justifiable granularity," *Appl. Soft Comput.*, vol. 13, no. 10, pp. 4209–4218, Oct. 2013.
- [25] E. Keogh and M. Pazzani, "A simple dimensionality reduction technique for fast similarity search in large time series databases," *Knowl. Inf. Syst.*, vol. 3, no. 2, pp. 263–286, 2000.
- [26] D. Yankov, E. Keogh, and U. Rebbapragada, "Disk aware discord discovery: Finding unusual time series in terabyte sized datasets," *Knowl. Inf. Syst.*, vol. 17, no. 2, pp. 241–262, Nov. 2008.
- [27] Q. Xie, C. Pang, X. Zhou, X. Zhang, and K. Deng, "Maximum error-bounded piecewise linear representation for online stream approximation," *VLDB J.*, vol. 23, no. 6, pp. 915–937, Dec. 2014.
- [28] X. Liu, Z. Lin, and H. Wang, "Novel online methods for time series segmentation," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 12, pp. 1616–1626, Dec. 2008.
- [29] E. Keogh and S. Kasetty, "On the need for time series data mining benchmarks: A survey and empirical demonstration," *Data Mining Knowl. Discovery*, vol. 7, no. 4, pp. 349–371, Oct. 2003.
- [30] T. Rakthanmanon and E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in *Proc. SIAM Int. Conf. Data Mining*, May 2013, pp. 668–676.
- [31] Y. Wu, D. Shah, and P. R. Childs, "Pseudo non-linear multiresolution analysis of non-stationary time series," *Signal Process.*, vol. 112, pp. 52–65, Jan. 2015.
- [32] B. Fuchs, L. Gruenwald, and M. Mirza, "Piecewise linear approximation of streaming time series data with max-error guarantees," in *Proc. SIAM Int. Conf. Data Mining*, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2014, pp. 358–366.
- [33] R. Agrawal, C. Faloutsos, and A. N. Swami, "Efficient similarity search in sequence databases," in *Proc. 4th Int. Conf. Found. Data Org. Algorithms (FODO)*, vol. 730. Cham, Switzerland: Springer, 1993, pp. 69–84.
- [34] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering—A decade review," *Inf. Syst.*, vol. 53, pp. 16–38, Oct. 2015.
- [35] A. Mueen, N. Chavoshi, N. Abu-El-Rub, H. Hamooni, and A. Minnich, "AWarp: Fast warping distance for sparse time series," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, Dec. 2016, pp. 101–110.

- [36] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.
- [37] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, Mar. 2005.
- [38] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, Oct. 2007.
- [39] H. Li, Z. Liu, and X. Wan, "Time series clustering based on complex network with synchronous matching states," *Expert Syst. Appl.*, vol. 211, Jan. 2023, Art. no. 118543.
- [40] H. A. Dau et al., "The UCR time series archive," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 6, pp. 1293–1305, Nov. 2019.



Yongming Huang received the Ph.D. degree from Southeast University, Nanjing, China, in 2012.

He is currently a Professor with the School of Automation Engineering, Southeast University. His main research interests include data mining, fault diagnosis, and health management.

Dr. Huang is a member of the China Association of Automation and the Seismological Society of Jiangsu Province, China.



Wen Shi received the M.Sc. degree from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2020. She is currently pursuing the joint Ph.D. degree with the School of Automation Engineering, Southeast University, Nanjing, and the Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Groningen, The Netherlands.

Her main research interests include time-series analysis and mining, machine learning and its applications, and anomaly detection.



Dimka Karastoyanova (Member, IEEE) received the Ph.D. degree in computer science from Technische Universität Darmstadt, Darmstadt, Germany, in 2006.

She is currently a Professor of information systems with the Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Groningen, The Netherlands. Her main research interests include adaptive, service-based, process-aware information systems, data-driven, autonomic

process performance improvement for business applications, scientific computing applications, and data processing pipelines.

Dr. Karastoyanova is an Active Member of the Business Process Management (BPM), Service-Oriented Computing, and Enterprise Computing Communities and a member of the Board of Informatics Europe.



Guobao Zhang received the Ph.D. degree from Southeast University, Nanjing, China, in 1998.

He is currently a Professor with the School of Automation Engineering, Southeast University. His main research interests include pattern recognition and machine learning.