A novel computational approach for discord search with local recurrence rates in multivariate time series

Min Hu\textsuperscript{a,d}, Xiaowei Feng\textsuperscript{a,d}, Zhiwei Ji\textsuperscript{b,*}, Ke Yan\textsuperscript{c,*}, Shengchen Zhou\textsuperscript{a,d}

\textsuperscript{a}SHU-UTS SILC Business School, Shanghai University, Shanghai 201800, China
\textsuperscript{b}School of Information & Electronic Engineering, Zhejiang Gongshang University, 18 Xuezheng Road, Hangzhou 310018, China
\textsuperscript{c}College of Information Engineering, China Jiliang University, 258 Xueyuan Street, Hangzhou 310018, China
\textsuperscript{d}SHU-SUCG Research Center of Building Industrialization, Shanghai University, Shanghai 200072, China

\textbf{Article info}

Article history:
Received 17 December 2016
Revised 17 October 2018
Accepted 25 October 2018
Available online 26 October 2018

Keywords:
Multivariate time series
Discord search
Recurrence structure
Time series segment
Outlier detection

\textbf{Abstract}

Discord search is an important technique for time series analysis, especially for anomaly detection. In recent years, many computational approaches of discord search were studied; however, limitation exists while only the problems with univariate time series data can be well addressed. In this study, we proposed a novel computational framework to identify discords from multivariate time series (MTS) data, namely, LRRDS (Local Recurrence Rate based Discord Search). LRRDS accurately identifies the discords by analyzing a recurrence plot, which is transformed from the original time series data. An innovative strategy was employed to improve the efficiency for pair-wise distance comparison of two subsequences. In the experimental simulations, LRRDS was applied to an extensive number of MTS datasets. Results show that the proposed approach is more efficient than existing methods, such as GDS. In conclusion, the LRRDS approach solves the adaptability problem of discord sequences in multi-dimensional space and guarantees the computational effectiveness and efficiency.

© 2018 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)

1. Introduction

In the field of time-series data mining, discord search (DS) is an important technique to identify anomalous subsequences of time series data [21]. Time series DS is proven to be useful in a diverse range of applications, such as abnormality diagnosis [40], health monitoring [13], and financial data analysis [43]. The limitation of the most existing approaches of DS is that they are usually limited to univariate time series (UTS) data [18]. However, time-series datasets for DS in many practical cases are multi-dimensional, which makes them difficult to be completely addressed by existing approaches [41]. Multivariate time series DS is challenging from two perspectives: (1) difficulties exist to precisely define “abnormality” in multivariate time series. Unlike the unusual values (significant high/low), which can be easily found in single-dimensional time series. An abnormality in high-dimensional time series may appear in a feature sub-space. (2) Irrelevant features may produce noise in the input data and mask the true anomalies. The performance of discord search algorithms in high-dimensional time series is also found to be affected by noisy information [4].

\footnotesize
\textsuperscript{*} Corresponding authors.
\textsuperscript{E-mail addresses:} jzw18@hotmail.com (Z. Ji), yanke@cjlu.edu.cn (K. Yan).

https://doi.org/10.1016/j.ins.2018.10.047
0020-0255/© 2018 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license. (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Over the past few years, some studies of DS in MTS were reported in the literatures. Most existing techniques for DS can be generally divided into two categories, namely, distance-based anomaly detection and model-based anomaly detection [9,12,36]. The distance-based anomaly detection includes KNN [28,38], K-means [32], DBSCAN [3], etc. The main limitation of the distance-based methods is that the choice of the distance measurements directly affects their performances. Model-based anomaly detection techniques include Bayesian networks [30], Markov models [35], neural networks [33], rough set theory [44] and support vector machines [10]. However, these intelligent anomaly detection approaches also have problems, such as: (1) lacking fault samples; (2) the training stage and testing stage of these algorithms are mutually independent, which lacks continuous learning ability. (3) For high-dimensional datasets, the learning time can be too long and may defeat the original purpose of learning [6]. Currently, there are a lot of publicly available time series datasets, e.g., UCI [49] and ODDS [50], which can be used to test anomaly detection algorithms. However, the efficiency of the existing methods is limited for real-world applications. Therefore, developing a novel anomaly detection algorithm with high accuracy, efficiency and generalization is still demanded.

As a powerful tool for visualization of the recurrences in two dimensional plots, recurrence plot (RP), can be used to identify recurrent or periodic behaviors from dynamic systems [8]. RPs were designed to detected non-stationarity in time series and are therefore candidates for anomaly detection [34]. However, RPs are rather difficult to interpret, they have not gained popularity beyond the admiration of the interesting patterns that they reveal. To solve this problem, Zbilut and Charles developed recurrence quantification analysis (RQA) to quantify the dynamical global properties from RPs, including recurrence rate, determinism, divergence, laminarity, etc. [45]. However, RQA cannot be directly used for DS. local recurrence rate, a new measurement metric of the complexity of RPs, reveals the potential to identify the changes and transitions of the dynamics in time series [42]. According to RP representation, a multivariate time series can be mapped into a new space without losing important information. The discord search of original time series can be implemented by identifying the abnormal local-structure from recurrence plot.

In this study, we proposed a novel computational approach LRRDS (Local Recurrence Rate based Discord Search), to identify the discords in multivariate time series. Firstly, a recurrence plot of original time series was generated. Secondly, we identified the time points with abrupt changes according to the local recurrence rate in RP. After segmenting the whole observed time axis with above change points, discords were detected by calculating the dissimilarity of any two subsequences. The LRRDS approach enables us to efficiently and precisely capture the discords that reveal in a long term and high-dimensional time series. In the experimental simulations, a single dimensional dataset [21], two well-known electrocardiograms (ECGs) datasets [21,24], a video surveillance dataset [19,21] with two dimensions and a special dataset (5 dimensions) from a real river tunnel project in China, were selected to test the performance of LRRDS. Our simulation results show that the proposed approach is more efficient to find the discords than existing methods, such as GDS [27]. Furthermore, the LRRDS method automatically searches the optimal parameters to ensure the accuracy of experiment results.

The rest of this work is organized as follows. In Section 2, we summarized some related works about the studies of discord search. The details of the proposed approach LRRDS were described in Section 3. In Section 4, we introduced the simulation experiments, including the validating datasets and parameters setting. Finally, we discussed the result of the experiment and conclude our work in Sections 5 and 6.

2. Related work

Recurrence plot (RP) is an important tool for visualization of time series data [48]. It is considered as a global, probabilistic autocorrelation function that reveals the relative frequencies of a system at given dynamical states [34]. The texture of the RP gives the impression about the time series from which it has been generated like for periodic signals the RP may have very long diagonal lines [31]. Based on its excellent performance of capturing nonlinear features in time-series data, RP has been applied in many fields, including fault diagnosis, and pattern recognition [11,47]. However, RPs are rather difficult to interpret; recurrence quantification analysis (RQA) was thus developed to measure many dynamic properties of RP, which is also an efficient way to identify recurrence features in time series [2]. Therefore, discord search with RQA appears to be an effective way to identify abnormal states from time series data. Nevertheless, the challenge of previous studies for discord search is the time complexity of RPs generation, which is \(O(n^2)\). Due to the time-consuming extraction of measure variables (such as determinism (DET), and laminarity (LAM)), RQA is more suitable for being used in offline time series analysis [34]. Moreover, RPs and RQAs were mainly used to visualize and quantify hidden structures, but it is difficult to automatically identify discords (anomalies) in time series. In this study, local recurrence rate was employed as a line-based RQA measure to analyze the local dynamics in multivariate time-series data so that the abnormal subsequences can be identified.

The existing RP-based discord discovery algorithms can be categorized into two types:

- **RP similarity-based methods.** Silva and colleagues proposed the use of RPs as representation domain for time series classification [37]. Their approach used the Campana–Keogh (CK-1) distance to measure the similarity between recurrent plots and proved that their distance measurement outperforms the Euclidean and dynamic time wrapping distance in most of testing datasets. Marwan et al. also reported that cross recurrence plots (CRP) can reveal fundamental relationship between two studied systems [29].

- **Local feature-based methods.** In Luo’s work [27], they proposed a general discord search (GDS) algorithm that is free of parameters. It searches for the top-K discords by computing the distance of each subsequence to its nearest neighbor
based on RPs. Koebbe et al. also found that qualitative information about local recurrence rates are obvious enough for anomaly detection with advantages of rapid computation and less data input [23].

The above methods are capable of providing fast and parameter-free solutions to overcome the general disadvantages in RP-based method: huge computational cost and unintuitive set of threshold value. However, the above two categories of method are mainly designed for univariate time series, and multivariate-based RQA analysis for anomaly detection is rare reported in the literatures.

To overcome the limitations listed above, we proposed a novel computational framework to identify discords in multivariate time series (MTS) based on local recurrence rate. By employing the adaptive segmentation of raw MTS, the orientation of the discord subsequence tends to be more accurate.

3. Local Recurrence Rate based Discord Search (LRRDS)

The proposed LRRDS approach includes three steps (see Fig. 1) and will be explained in detail in the following subsections.

3.1. Recurrence plot (RP) representation

In this section, we introduce the way of generating a recurrence plot (RP) from the original time series data. Given a $d$-dimensional time series dataset $X^O = [X_1, X_2, \ldots, X_i, \ldots, X_d]$, where vector $X_i$ corresponds to the $i$th observed variable in this system ($i \in [1, d]$), each column in $X^O$ is represented as an univariate time series: $X_i = [v_{1i}, v_{2i}, \ldots, v_{ki}, \ldots, v_{ti}]^T$, which is
sampled at $t$ time points ($k \in [1, t]$). The original dataset $X^0$ is firstly normalized as:

$$X'_i = 1 + \frac{X_i - \min(X_i)}{\max(X_i) - \min(X_i)}$$

(1)

In Eq. (1), $X_i$ is the time series of $i$th variable in original dataset $X^0$, and $X'_i$ is its normalized vector. Obviously, each element in $X'_i$ is in the range from 1 to 2. By performing the same normalization for each column in $X^0$, we obtain the normalized dataset $X^E = [X'_1, \ldots, X'_i, \ldots, X'_n]$.

Secondly, we adopt time-delay embedding technique [18] to implement phase space reconstruction on the normalized dataset $X^E$ as Eq. (2):

$$X^E(j, :) = [X^E(j, h), X^E(j + \tau, h), \ldots, X^E(j + (m - 1)\tau, h)]$$

(2)

In Eq. (2), $X^E(j, :)$ is the $j$th row in $X^E$, which represents the $j$th timestamp for the $h$–th variable $(1 \leq j \leq t - (m - 1)\tau, 1 \leq h \leq d)$; $m$ is the embedding dimension; and $\tau$ is time delay. Consequently, $X^E$ is a $(t - (m - 1)\tau) \times m$ array where every element is a vector with length $d$. In our experiment, we set embedding dimension $m$ to be 1, since it was proved sufficient to detect dynamics on RPs [17]. Delay $\tau = 1$ is also a constant in our experiments.

Next, we construct a recurrence matrix $R(t \times t)$ in order to draw the corresponding recurrence plot (RP). The binary element $R_{ij} (i,j \in [1, t])$ in the matrix $R$ is determined by two binary variables $ED'_{ij}$ and $BD'_{ij}$ through a logical "and" operation:

$$R_{ij} = ED'_{ij} \cap BD'_{ij}$$

(3)

In Eq. (3), the binary variable $ED'_{ij}$ and $BD'_{ij}$ are the elements in matrix $ED'$ and $BD'$, respectively. $R_{ij}$ equals 1 if and only if $ED'_{ij} = BD'_{ij} = 1(1 \leq i, j \leq t)$; otherwise, equals 0. Moreover, both $ED'_{ij}$ and $BD'_{ij}$ are calculated from Eq. (4):

$$D'_{ij} = f(D_{ij}) = \begin{cases} 1, & D_{ij} \leq \varepsilon \\ 0, & D_{ij} > \varepsilon \end{cases}$$

(4)

In Eq. (4), $D_{ij}$ is the distance between timestamp $i$ and timestamp $j$ of $X^E$, and parameter $\varepsilon$ is the threshold of distance. The purpose of the function in Eq. (4), namely Heaviside function [39], is to decide the value of $D'_{ij}$ depending on the values of $D_{ij}$ and $\varepsilon$. In this study, $\varepsilon$ is calculated as 25% of the maximal distance value of $D_{ij}$ [48].

Based on formula (4), we have $ED'_{ij} = f(ED_{ij})$, $BD'_{ij} = f(BD_{ij})$, where $ED_{ij}$ and $BD_{ij}$ are the Euclidean distance [5] and the Bhattacharyya distance [1] between timestamp $i$ $(1 \leq i \leq t)$ and $j$ $(1 \leq j \leq t)$ in $X^E$. The definition of $ED_{ij}$ and $BD_{ij}$ are described in Eqs. (5) and (6):

$$ED_{ij} = \sqrt{\sum_{h=1}^{d} (e_{ih} - e_{jh})^2}$$

(5)

$$BD_{ij} = \sqrt{1 - \frac{\sum_{h=1}^{d} (e_{ih}e_{jh})}{(\sum_{h=1}^{d} e_{ih}) * (\sum_{h=1}^{d} e_{jh})}}$$

(6)

In Eqs. (5) and (6), $e_{ih}$ and $e_{jh}$ represent the $i$–th and $j$–th timestamp of the $h$th variable in $X^E(1 \leq h \leq d)$. If $d$ is equal to 1, $e_{ih}$ and $e_{jh}$ are 1-dimensional vector. Once recurrence matrix $R$ is constructed following the above steps, the corresponding recurrence plot can be generated.

3.2. Change point detection

After we obtained the RP of the original time series dataset, the next step is to detect the abrupt change points [26]. Fig. 2 shows the workflow of detecting the change points from RP.

Under certain situations, it is more useful to focus on the local distribution of recurrence points rather than the global distribution [7]. We use the local recurrence rate (LREC) (Eq. 7), to represent the density of a local region in the RP (see Fig. 2A).

$$\text{LREC}(p) = \left(\frac{1}{w^2}\right) \sum_{i,j=p-w+1}^{p} R_{i,j}, \quad w \ll t$$

(7)

where the value of LREC is calculated within a square region $(w \times w)$ at timestamp $p (1 \leq p \leq t - w + 1)$.

According to the window-based strategy [20], we use a sliding window $(w \times w)$ to scan the local recurrence on the RP graph along the diagonal direction, and produce a time varying vector of local recurrence rate $LREC_t = \{LREC(1), \ldots, LREC(t - w + 1)\}$. Fig. 2B shows the curve of local recurrence rates at different time points.
For easy identification of the change points from the LREC curve, each value of \( LREC(p) \) was transformed to Boolean signal using Eq. (8).

\[
LREC'(p) = \begin{cases} 1, & LREC(p) > 0 \\ 0, & LREC(p) \leq 0 \end{cases}
\]  

(8)

Based on Eq. (8), we get a 0–1 vector \( LREC_{v} = [LREC'(1), \ldots, LREC(t - w + 1)] \) from \( LREC_{v} \) (Fig. 2C). A vector \( CP = [t_{q_1}, t_{q_2}, \ldots, t_{q_l}, \ldots, t_{q_s}] \) was defined as a sequence, which includes all change points. The element \( t_{q_l}(l \in [1, s]) \) satisfies the condition \( [LREC'(t_{q_l} + 1) - LREC'(t_{q_l})] = 1 \). By binarizing the \( LREC_{v} \) vector, the change points can be easily identified when the adjacent values in the \( LREC_{v} \) change suddenly.

Finally, we extract subsequence sets \( S \) from LREC curve by separating it with the change points included in \( CP \) (see Fig. 2D). Apparently, all obtained subsequences are non-overlapping. In the next section, we determine the discords from all subsequences.

3.3. Discord detection

In this study, a subsequence in LREC curve was considered as a discord if it has the largest distance to its nearest non-overlapping neighbor (subsequence) [21]. The term \textit{global discord degree} is used to denote the dissimilarity between any two subsequences in LREC curve. Since the shape of subsequence curve directly reflects the dissimilarity between subsequences, we choose two statistical features: “Length” and “Mean” (with two variables \( V_L \) and \( V_M \) in formula (9-10)) [22], to simply represent a subsequence in LREC curve (Fig. 3).

\[
V_L = q_j - q_i
\]  

(9)

\[
V_M = \frac{1}{q_j - q_i} \sum_{m=q_i}^{q_j} t_m
\]  

(10)

where \( t_{q_i} \) and \( t_{q_j} \) are two time points from the CP vector.

Based on all subsequences obtained in Section 3.2, we further construct a matrix \( Q \) (\( |S| \times 2 \)), which consists of all the values of “Length” and “Mean” for these subsequences. The matrix \( Q \) was substituted into Eqs. (5) and (6) to obtain two \( |S| \times |S| \) distance matrices \( ED' \) and \( BD' \), respectively, where \( |S| \) is the total number of subsequences in the original LREC curve.
Next, matrix $ED^i$ and $BD^i$ were binarized to matrices $ED^i'$ and $BD^i'$ using Eq. (11):

$$D^i_{ij}' = f(D^i_{ij}) = \begin{cases} 0, & D^i_{ij} \leq \varepsilon \\ 1, & D^i_{ij} > \varepsilon \end{cases}$$

(11)

where the threshold $\varepsilon$ differentiates whether the $i$-th subsequence belongs to the neighborhood of the $j$-th subsequence.

Thirdly, the discord degree matrix $OD$ is calculated from $ED^i'$ and $BD^i'$ using Eq. (3). The global discord degree (gdd) of subsequence $i$ from LREC curve is defined as shown in Eq. (12)

$$\text{gdd}_i = \left( \frac{\sum_{j=1}^{|S|} OD_{ij}}{|S| - 1} \right)$$

(12)

where $OD_{ij}$ represents whether or not subsequence $i$ and $j$ are dissimilar, and $|S|$ is the total number of subsequences ($1 \leq i, j \leq |S|$). The larger value of gdd, indicates that the corresponding subsequence is more different from the others. Given the global discord degree vector $\text{GDD} = \{\text{gdd}_1, \text{gdd}_2, \ldots, \text{gdd}_{|S|}\}$, all the elements in this GDD vector are classified as two clusters using $K$-means clustering ($K$ is pre-determined as 2). These two clusters represent abnormal and normal subsequences, respectively. The cluster includes higher values of gdd, indicates a set of potential discords.

### 3.4. Acceleration and parameter optimization

Due to the fact that the computational cost increases sharply with the size increment of the time series datasets, a compressive strategy PAA (Piecewise Aggregation Approximation) [25] is employed to accelerate the process of our proposing approach. Given a normalized time series datamatrix $X^0$ with $n$ time points ($X^0 = [t_{s1}; t_{s2}; \ldots; t_{sn}]$) and the compressed ratio $c$, it is compressed to $X^\beta$ with $h$ time points ($\hat{T} = [\hat{t}_{s1}; \hat{t}_{s2}; \ldots; \hat{t}_{sh}]$). The element at $i$th time point in $X^\beta$ is calculated by the following equation:

$$\hat{t}_{si} = \frac{h}{n} \sum_{j=c(i-1)+1}^{c(i-1)+1} t_{sj}$$

(13)

where $c = h/n$. The Eq. (13) divides the original time series data into $h$ segments with equal size, and extracts the mean value of each segment.

Furthermore, the time complexity of generating a globle recurrence matrix is $O(N^2)$, which is a serious problem if the size of dataset is large (Fig. 4A). According to the description in Sections 3.1–3.3, scanning the local recurrence matrix along the diagonal is adequate to obtain the LREC curve for discord search, which reduces the time complexity to $O(kN)$. Fig. 4 shows the idea of quickly scanning the local recurrence structure while the LREC curve is generated. We defined a parameter $k$ to represent the range of the local recurrence matrix, which is shown in Fig. 4B.
Next, we choose the optimal values of three parameters: c, k, and w. The optimization of parameter c was reported in the literature [21]. The optimal values of k and w to can be solved consequently using Eq. (14).

\[
\begin{align*}
\min_{k, w \in \mathbb{R}^d} & \sqrt{\frac{1}{n} \sum_{i=1}^{n} [s_i - \bar{s}]^2} \\
\text{s.t.} & \quad k - w > 0, \quad w_1, ..., w_d \in \mathbb{N}^+ 
\end{align*}
\]

where s is the vector of the length values of all the subsequences, and \(\bar{s}\) is the mean value of s. \(k - w\) denotes the slack margin (Fig. 4B).

**4. Experiment design**

In this section, we describe the simulation experiments on extensive datasets and evaluate the performance of our proposed LRRDS approach.

**4.1. Datasets**

In the simulation experiment, five time series datasets with different dimensions were used to validate the effectiveness of our proposed approach. The details of these datasets were described in Table 1. The first dataset TEK17 is a single-dimensional time series of space shuttle marotta valve, which describes the energized and de-energized cycle of an engine [21,28]. The next two datasets are electrocardiograms (ECGs) datasets, whose dimensions are two. Electrocardiograms are time series datasets of electrical potentials between two points on the body surface caused by heart [21]. As shown in Table 1, ECG1 is collected from BIDMC Congestive Heart Failure database [21], and ECG2 is downloaded from MIT-BIH Arrhythmia database [21]. The fourth dataset is a video surveillance dataset with two dimensions [19]. This 2D time series dataset was extracted from a video of an actor performing various actions with and without a replica gun. This time series dataset from a river tunnel project [14].

**Table 1**
The description of datasets in experiments.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Dimensionality</th>
<th>Ground truth (Real discords)</th>
<th>Data sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space shuttle (name: TEK17)</td>
<td>1</td>
<td>1100–1200</td>
<td>[21,24]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1450–1550</td>
<td></td>
</tr>
<tr>
<td>ECG1 (name: chfdbchf15)</td>
<td>2</td>
<td>2400–2500</td>
<td>[21,27]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4000–4400</td>
<td></td>
</tr>
<tr>
<td>ECG2 (name: Mitdb/x_mitdb/x_108)</td>
<td>2</td>
<td>9950–10,600</td>
<td>[21,24]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10,500–11,600</td>
<td></td>
</tr>
<tr>
<td>Video surveillance</td>
<td>2</td>
<td>300–430</td>
<td>[19,21,52]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1465–1590</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>1913–2964</td>
<td></td>
</tr>
<tr>
<td>Dataset from a river tunnel project</td>
<td>5</td>
<td>8500</td>
<td>Hu et al. [14].</td>
</tr>
</tbody>
</table>
dataset measures the X and Y coordinates of the actors right hand. The actor draws a replica gun from a hip mounted hol-ster, aims it at a target, and returns it to the holster. These above four datasets can be directly download from the website: http://www.cs.ucr.edu/~eamonn/discords. The last one was collected from a real project of a certain river tunnel in China. The length and diameter of this tunnel is 8.95 km and 13.7 m, respectively. Due to its long excavation, large diameter and complicated geologic factors, there is a high probability for disastrous risk in the processes of construction. For example, the ground surface was seriously collapsed during the construction of tunnel around 14:00 on the day 19th May in 2008. This special dataset includes 10,000 time points during the time period from 05-01-2008 to 05-21-2008 when the river tunnel up-line was constructed. The interval of data sampling is three minutes. Every detected time points corresponds to five variables, which are related to sealed cabin pressure in the shield [14].

4.2. Experiment design

In this study, the simulation experiment was performed to validate the efficiency and effectiveness of the proposed approach (LRRDS). First, the single-dimensional space shuttle dataset was processed. Second, two ECG datasets were sequentially used in LRRDS. Third, the video surveillance dataset was adopted to further test the performance of LRRDS. The preprocessing of the datasets is already done before they were published online. Last, we validated the LRRDS on the special dataset (five dimensional) of river tunnel in the real-world application. For each dataset, we evaluated the results from two aspects, which were shown in the following section.

4.3. Parameter setting

All the simulations were performed using the R programming language with Rstudio V7.8 on Ubuntu 14.04 platform. The hardware setups are 2.30 GHz CPU and 4.00 GB RAM. We set embedding dimension as m = 1, and slack margin k – w = 30 for all the simulated datasets. The performance of the proposed approach was evaluated on two aspects, namely, effectiveness and efficiency. In this study, F1 score [2] was applied to evaluate the effectiveness of the proposed algorithm. The definition of F1 score is defined as shown in formula (15):

$$F_1 = \frac{2 \times PR}{P + R} \times 100\%$$  \hspace{1cm} (15)

Where P is precision and R is recall and they can be shown as $P = \frac{TP}{TP+FP}$, $R = \frac{TP}{TP+FN}$. TP represents the true positive results, while FP and FN represent the false positive and false negative results, respectively. In addition, an F1 score reaches its best value at 1 and worst at 0.

In addition, the efficiency was evaluated with the time complexity of LRRDS, such as:

$$T_{eff} = O(kN)$$  \hspace{1cm} (16)

Where k and N denote the range of local recurrence matrix and the size of the original time series, respectively. Since the distance calculation in LRRDS model is time consuming [21], particularly for the high-dimensional and long time series data, two indicators are chosen to represent the time complexity, including: (1) the number of times that two distance function is called; (2) end-to-end running time.

5. Results

5.1. Anomaly detections on five datasets

As described in Section 4, the proposed approach was applied to all the five datasets. The results of anomaly detections were analyzed in detail in this section. The values of global discord degree (gdd) for the first eight discords in each dataset are listed in Table 2, in which the blue ones are identified as abnormal ones by K-means clustering (K=2).

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Eight potential discords</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st</td>
</tr>
<tr>
<td>Space shuttle</td>
<td>0.92</td>
</tr>
<tr>
<td>ECG_1</td>
<td>0.99</td>
</tr>
<tr>
<td>ECG_2</td>
<td>0.96</td>
</tr>
<tr>
<td>Video surveillance</td>
<td>0.99</td>
</tr>
<tr>
<td>Engineering</td>
<td>0.92</td>
</tr>
</tbody>
</table>
(I) Anomaly detection in single-dimensional space shuttle dataset

Consistent with the first row in Table 2, Fig. 5A also reveals one discord for the space shuttle dataset predicted by LRRDS. The discord was marked to each variable with red color (Fig. 5B–C). In Fig. 5B–C, it is clear that there is a globale discord (marked with red color) in the early stage of time series, which can be easily detected from the LREC curve of space shuttle data. Fig. 5B–C show the dynamic changes on each variable of the dataset during the time slot when the discord occurred. These two variables are exactly the same shape, because they were produced by phase space reconstruction. It is the same case for the two vertical lines. It can be concluded that the discord corresponds to a missing peak before the large plateau. The predicted discord covers the real discords shown in Table 1.

(II) Anomaly detection in Electrocardiograms

In this subsection, we particularly discuss about the results of LRRDS on the dataset ECG1 and ECG2 (see Figs. 6–7). For ECG1, we identified an obvious discord from this dataset using the K-means clustering method, which is consistent with the ground fact (Table 2). The LREC curve of ECG1 reveals that the region from 2190 to 2500 is significantly different from other regions (Fig. 6A). Comparing Fig. 6B with C, we finally found that the second variable in the dataset made great contribution to this discord in the whole series. This prediction is consistent with [48]. Similarly, we presented three subtle discords from ECG2 (Fig. 7). Actually, our approach reported 5 discords (Table 2). However, we found that some detected discords can be merged because their regions on the timeline are adjacent. Fig. 7 shows three labeled discords after merging. The results shown in Fig. 7A–C were consistent with the conclusions in literature [21].

In conclusion, the above results indicate that our computational framework for MTC analysis is capable of detecting the anomalies from the target time series datasets.

(III) Abnormal human behavior identification

As we see in Table 2, five discords were detected from the video surveillance dataset and we found that they can be merged into four due to adjacency of timeline regions. In Fig. 8A, four discords are identified by LRRDS from the video surveillance dataset. These discords located mainly in the first half regions and each discord reflects an abnormal human behavior (see Fig. 8B–C).

(IV) Risk forecasts in engineering construction

All above experiments were performed on low-dimensional datasets, and the discords can be easily recognized. We present the results when LRRDS was applied on a 5-dimensional dataset, which comes from a real-world project of river tunnel project (see Fig. 9). From the LREC curve shown in Fig. 9A, there was one discord in the later stage of this time series. Our predicted results (Fig. 9B–F) show that the discord occurred at the timestamp between 6690 (22:30 on 13th May 2008) and 7730 (02:30 on 16th May 2008), which is 2–5 days earlier than the time of accident (18th May 2008). From Fig. 9, we found that the 5D time series includes a dramatic changes around 8000. The prediction is consistent with our fault log of this project.
Fig. 6. LREC curve and identified discord on the ECG1 dataset \((w = 2, c = 10\%)\). (A) LREC curve; (B–C) The identified discords in variable 1 and variable 2.

Fig. 7. LREC curve and identified discords on the ECG2 dataset \((w = 2, c = 10\%)\). (A) LREC curve; (B–C) The identified discords in variable 1 and variable 2.
Fig. 8. (A) LREC curve; (B) The identified discords from X and Y coordinate on the video surveillance dataset ($w = 2$, $c = 10\%$).

Fig. 9. LREC curve and identified discord on the river tunnel dataset ($w = 4$, $c = 10\%$). (A) LREC curve; (B–F) The identified discords in variable 1 to variable 5.
5.2. Performance evaluation

In this part, we evaluated the effectiveness of the LRRDS method using $F_1$ score, as shown in formula (15). We assumed that the results reported in [48] were true, since the abnormal sequences were labelled by domain experts. Our first three datasets come from the same sources as literature [48]. Therefore, we directly compared our predicted results with the results in [48]. We concluded that our detected results were completely consistent with the experts labelled discords. Therefore, the $F_1$ score of all the first three experiments were 1 which signified the effectiveness of our algorithm.

Finally, we verified the efficiency of the proposed approach through additional two experiments. Firstly, we applied both LRRDS and GDS [27] on four sub-sets (generated from a long time series ECG qtdb/se102 [52]), and presented the comparison results in Table 3. It is clear that LRRDS requires fewer distance calculations and spends less time than GDS. The LRRDS is more efficient for large-scale time series datasets. However, GDS can only address a portion of dataset with one dimensionality each time, while LRRDS is able to process multi-dimensional data at one time. Moreover, we compared the effects of different compressed ratios on four different time series [51,52]. For each time series, the value of compressed ratio $c$ is 1/5, 1/10, and 1/15, respectively. Other parameters were fixed as: $w=2$, $k=32$. As shown in Table 4, the efficiency of algorithm decreases when the value of compressed ratio is increased.

5.3. The effects of parameter settings on computational results

In LRRDS, there are two key parameters, which play key roles in the final results. The most important one is the size of sliding window $w$, which affects the detection of change points. Additionally, the slack margin is also essential. When $w$ is fixed, the slack margin supplies visual information to determine the recurrence point. The larger the slack margin is, the more significant the discord will be; while the smaller the slack margin is, the lower the time complexity will be. Therefore, there is a tradeoff between detection accuracy and efficiency.

6. Discussion and conclusions

The multivariate time series (MTS) analysis is a very difficult problem because of the complexity of the MTS data type. The main challenges of processing time series data for DS involve the high dimensionality [46], noise and redundancy in the data [16]. In this study, we introduced a novel computational approach, LRRDS, for multivariate time series data DS. LRRDS was designed using the concept of RQA and outperformed existing typical methods. Numerous experiments were conducted with different datasets to evaluate the performance of the proposed method. The simulation results demonstrate the better performance, as well as the higher detection accuracy of the proposed approach, comparing with existing methods. In summary, LRRDS is suitable for anomaly detection on both multivariate and univariate time series.

The existing limitations of the proposed LRRDS algorithm are: (1) only two statistical features (mean, and length) were used to represent the subsequence of LREC curve; (2) the dimensionality of the testing datasets in the experiments is not high. There are mainly two routines to be considered for the future works: (1) more features can be introduced to represent the local dynamics of LREC curve for improving the clustering results of abnormal and normal subsequences; (2) more high-dimensional time series datasets will be tested, including benchmark and real datasets. In addition, we plan to explore the parameter optimization strategy [15] and discord detection applications under real-world data context.
Acknowledgment

This work is supported by the Foundation of Zhejiang Provincial Department of Education (No. 1120KJ0416255), the Foundation of talent's start-up project in Zhejiang Gongshang University (1120XJ2116016), and the Foundation of Shanghai Science and Technology Committee (Nos. 18DZ1205502, 17DZ1204203, 18DZ1201204). This study was partially supported by the National Science Foundation of China (Nos. 61472282, 61850410531). We also thank valuable discussion from Dr. Lei Shi at Zhejiang Gongshang University.

Conflict of interest

The authors declare that they have no conflict of interest.

Author Contributions


Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2018.10.047.

References


