Journal of Intelligent Manufacturing (2021) 32:1669–1678
https://doi.org/10.1007/s10845-021-01768-1

Temporal anomaly detection on IIoT-enabled manufacturing

Peng Zhan[1] · Shaokun Wang[2] · Jun Wang[2] · Leigang Qu[2] · Kun Wang[2] · Yupeng Hu[2] · Xueqing Li[1]

Received: 16 November 2020 / Accepted: 25 March 2021 / Published online: 15 April 2021

Abstract

Along with the coming of industry 4.0 era, industrial internet of things (IIoT) plays a vital role in advanced manufacturing. It can not only connect all equipment and applications in manufacturing processes closely, but also provide oceans of sensor data for real-time work-in-process monitoring. Considering the corresponding abnormalities existing in these sensor data sequences, how to effectively implement temporal anomaly detection is of great significance for smart manufacturing. Therefore, in this paper, we proposed a novel time series anomaly detection method, which can effectively recognize corresponding abnormalities within the given time series sequences by standing on the hierarchical temporal representation. Extensive comparison experiments on the benchmark datasets have been conducted to demonstrate the superiority of our method in term of detection accuracy and efficiency on IIOT-enabled manufacturing.

Keywords Advanced manufacturing · Temporal representation · Anomaly detection

Introduction

Along with the coming of industry 4.0 era, the modern manufacturing goes beyond the traditional assembly but rather strives to adopt some new enabling technologies (e.g., Internet of Things, Cyber-Physical System, Big Data Analytics, and Cloud Computing) to realize smart manufacturing. Specially, one critical enabling technology for smart manufacturing is the Industrial Internet of Things (IIoT). It can not only connect all equipment and applications in manufacturing processes closely for dynamic cooperative controlling, but also provide vital data-driven innovations for promoting manufacturing performance (Yang et al. 2019; Cheng et al. 2020).

Taking the assembly integration test, dubbed as AIT, on engine manufacturing in Fig. 1 as an example, AIT contains different processing stages. Each stage performs independent assembly and testing, and then these assembled components in the current stage are transferred to the next stage for corresponding processing. In other words, AIT closely connects the production processes of different stages to achieve overall assembly and performance controlling. To ensure the processing stability at different stages and the reliability of overall assembly, AIT utilizes some pervasive sensing technologies and devices, such as radio frequency identification (RFID), sensor network, ultra wide band, and etc, to produce oceans of sensor data for real-time work-in-process monitoring and controlling. As illustrated in Fig. 1, these sensor data are collected in the temporal order, expressed as $T = \{< v_1, t_1 >, \ldots, < v_i, t_i > \ldots, < v_n, t_n >\}$, where element $< v_i, t_i >$ indicates that the received value v_i at the time t_i, with the time strictly increasing. Obviously, these time series sensor data reflect the real-time processing status of specific components in different stages. More importantly, there are corresponding abnormalities in these sequences, revealing specific defects in the components. Therefore, accurate and efficient anomaly detection is not only the basis for avoiding assembly risks, but also of great significance for optimizing manufacturing process (Sellami et al. 2020).

Towards this end, in this paper, we concentrate on anomaly detection on time series, which generally have large volume, high dimensionality, real-time generation/updating features, thus dubbed as time series in what follows. Time series anomaly detection, aiming for identifying unexpected observations within the given time series, attracting great interest from both industry and academia over the last decade (Huang

✉ Yupeng Hu
huyupeng@sdu.edu.cn

[1] School of Software, Shandong University, Jinan 250101, Shandong, China

[2] School of Computer Science and Technology, Shandong University, Qingdao 266237, Shandong, China

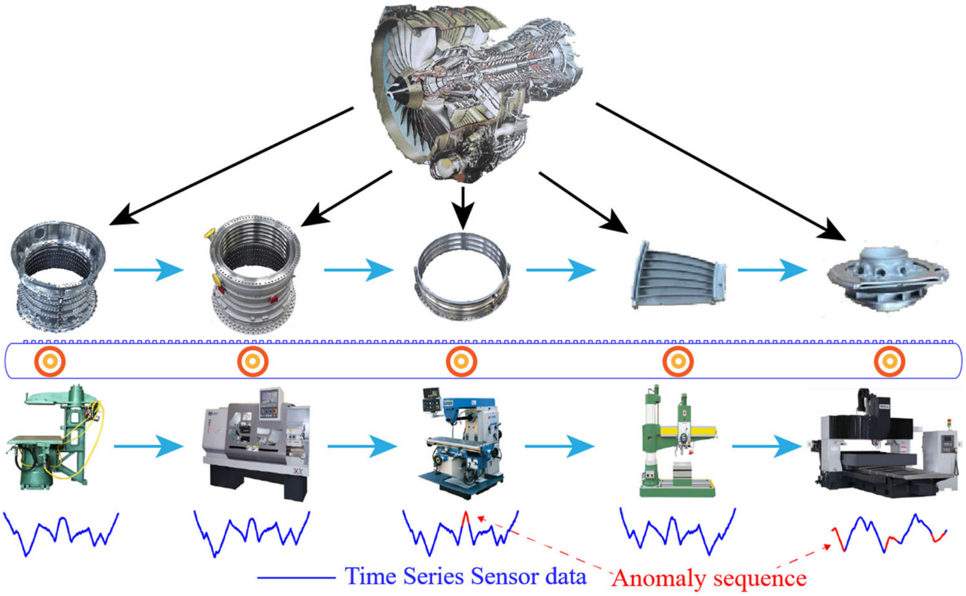
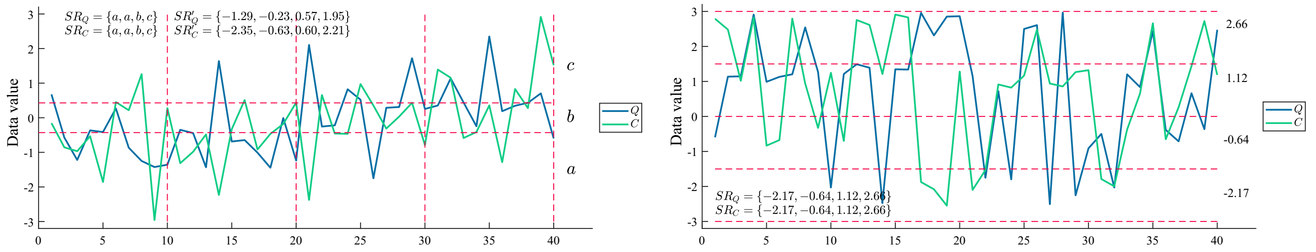


Fig. 1 Schematic illustration of time series sensor data involving anomalies in AIT



(a) SAX-AD

(b) APAA-AD

Fig. 2 Illustration on the defects of two existing methods

et al. 2020; Hsu et al. 2021; Dong et al. 2019; Pang et al. 2018; Abdulla and Hashimy 2018; Xue et al. 2018).

Considering the large amount, high dimensional and continuous accumulating features of time series, it is incapable for anomaly detection directly on the raw time series (Hu et al. 2019, 2020a, b). Therefore, Keogh et al. (2006) proposed a symbolic aggregate approximation (SAX) representation based method for anomaly detection, dubbed as SAX–AD in this paper. This method first divides the given time series into a series of sequences (segments) with equal length, and then it transforms the mean values into the corresponding SAX representation set for all the sequences to further implement anomaly sequence detection. Considering that only adopting mean value based symbolic representation may cause the loss of key information to affect the final detection accuracy, Ren et al. (2018) proposed an amplitude domain division based Piecewise Aggregate Approximation method for anomaly detection, dubbed as APAA–AD in this paper. Compared to SAX–AD, on the one hand APAA–AD directly uses the real mean values rather than value-transformed symbols to complete representation, thus reducing the quantization error; on the other hand, APAA–AD further subdivides all the sequences into equal size subsections according to the oscillation of amplitude domain to implement fine-grained representation, while improving the detection accuracy.

To make our point more clear, we take parts of momentum wheel (MW) time series data from aircraft industry, as an example. In Fig. 2a, two time series Q and C are divided into 4 equal-length sequences and further represented into the exact same results by SAX–AD, i.e. $SR_Q = SQ_C = \{a, a, b, c\}$, thus fails to effectively identify these two different sequences. Conversely, with the amplitude domain division strategy, APAA–AD can represent these two sequences as $SR'_Q = \{-1.29, -0.23, 0.57, 1.95\}$ and $SR'_C = \{-2.35, -0.63, 0.60, 2.21\}$ respectively, and distinguish them from each other effectively. Although APAA–AD does improve the detection accuracy to some extent, it only focuses on amplitude domain oscillation based representation, which cannot comprehensively identify abnormal sequences from their amplitude and temporal features, and thus fails to further improve detection performance. As shown in Fig. 2b, two time series are represented into the exact same results by APAA–AD, i.e. $SR_Q = SQ_C = \{-2.17, -0.64, 1.12, 2.66\}$, in other words, only rely on the amplitude oscillation trend for representation may result in missing temporal information, and hence fails to detect the corresponding anomaly sequences.

Motivated by the above analyses, we presented a novel hierarchical representation for anomaly detection, dubbed as HR–AD. Compared with other anomaly sequence detection methods, HR–AD not only pays close attention to the significant changes of time series in the amplitude domain, but also keeps a watchful eye on the corresponding variations in the hierarchical representation domain. Therefore, all anomaly sequences can be detected effectively through multi-domain representation based anomaly score calculation. The main contributions of this work are three-fold:

1. We proposed a novel time series hierarchical representation, which can present the significant amplitude and temporal features of time series, by standing on a multi-domain view, to provide more differentiable representation.
2. We developed an effective anomaly detection method, which comprehensively measures the corresponding features from multi-domain space, while enhancing anomaly recognition capability.
3. We performed extensive experiments on benchmark datasets, to justify the superiority of our method on IIOT-enabled manufacturing.

Related work

For a given time series, anomaly detection focuses on finding some observations, which are maximally different to the rest part of time series (Keogh et al. 2006). According to the detailed introduction on outlier detection by Gupta et al. (2014), anomaly detection within a given time series can be categorized as follows.

Anomaly point detection

Anomaly point detection aims to identify outlier data points within a given time series. Helman and Bhangoo (1997) utilized a normal data model to build histograms for anomaly detection, which can be applied to many applications, such as computer security, biomedical testing, networks. Yamanishi et al. (2004) proposed a smartsifter from the viewport of statistical learning theory to detect network intrusion. Tsay et al. (2000) proposed a vector autoregressive integrated moving-average model to detect anomalies. Although the above statistical model based methods do have certain effectiveness. The main obstacle of them are how to set corresponding appropriate models for different types of time series. In other words, for different data types, the corresponding data distribution is diverse, especially facing with the high-dimensional time series, the appropriate data distribution is difficult to estimate. Considering this defect, Breunig et al. (2000) introduced a density-based anomaly detection method, which is called the Local Outlier Factor (LOF). This method detects anomalies by comparing the local density of each data to its neighbor local density, which captures the relative degree of isolation. Compared to other statistical model based methods, LOF does not need to be restricted by the data distribution. Nevertheless, it has to calculate the distance between any two objects of object set for anomaly detection, which is not suitable for detecting anomalies on large amount, high-dimensional time series.

Anomaly sequence detection

Anomaly sequence detection dedicates to find unusual sequences within the given time series. The brute force solution iteratively calculates the distance of each two sequences in the given time series, so as to find abnormal sequences that are far away from normal sequences. This method is simple and intuitive. However, due to the large overhead of traversal calculation, the processing efficiency of brute force method is not ideal. Considering the above defects, Several faster variations have been proposed to improve the efficiency of detecting anomalies. Pruning based detection methods are proposed to improve the corresponding processing efficiency. Keogh et al. (2005) proposed a heuristic reordering of candidate subsequences to accelerate the process of detection, called HOT SAX. HOT SAX first transforms the given time series into symbolic words, and then embeds all the symbols into an augmented tree, whose leaves contain a linked list index for corresponding symbolic words. With the help of this index structure, heuristic sorting and pruning can be applied in both outer and inner loops, which accomplished three to four order of magnitude of speed-up compared with the brute force approach. Wei et al. (2006) utilized locality-sensitive hashing to estimate similarity between sequences more efficiently. Considering that Piecewise Linear Representation (PLR) (Keogh and Smyth 1997; Zhan et al. 2020) can preserves the trend features of time series. Leng et al. (2013) proposed an anomaly detection algorithm based on PLR representation and density-based function (LOF). However, finding appropriate segmenting points for PLR is not a trivial task (Guerrero et al. 2010) and the corresponding search efficiency is relatively low. Accordingly, Kha and Anh (2015) proposed a novel LOF based anomaly detection method, which utilized cluster-based LOF to improve detection accuracy. However, the corresponding clustering operation is also affect the efficiency, especially when facing large volume of time series, the performance of this method decreases sharply. To address this defect, Ren et al. (2018) proposed aamplitude domain division based Piecewise Aggregate Approximation for anomaly detection

(APAA–AD). It directly uses the real mean values rather than value-transformed symbols to complete representation, thus reducing the quantization error. Besides, the processing efficiency of APAA–AD is also improved while reducing the data dimension. Although APAA–AD does improve the detection performance, it only focuses on amplitude domain oscillation based representation, which cannot comprehensively identify abnormal sequences to further improve detection accuracy.

After reviewing the previous work on time series anomaly detection, we got two inspirations as follows. On the one hand, time series representation, which can reduce the original dimensionality while retaining the important features of the raw time series, is the basis of anomaly detection. On the other hand, an effective abnormal measurement is the key to anomaly detection. For such reasons, in this paper, we aim to not only comprehensively represent the corresponding sequences within the given time series by standing on a multi-domain view, but also propose an efficient abnormal evaluation strategy for anomaly detection.

Preliminaries

Definition 1 (Time Series) For a given time series T with n data points could be expressed as

$$T = \{< v_1, t_1 >, \ldots, < v_i, t_i > \ldots, < v_n, t_n >\}, \tag{1}$$

Thereafter, we utilize the traditional sliding window (SW) approach (Keogh et al. 2001; Luo et al. 2020) to produce temporal continuous sequences from T.

Definition 2 (Sequence Set) Supposing the length of SW is k $(k < n)$, T is divided into $L(\lceil n/k \rceil)$ sequences to form the sequence set $(\widetilde{S})$, expressed as

$$\widetilde{S} = \{S_1, S_2, \cdots, S_l, \cdots, S_L\}, \tag{2}$$

where the l-th sequence in Set, can be indicated as

$$S_l = \{v_{l\times k}, v_{l\times k+1}, \cdots, v_{l\times k+k-1}\}, \tag{3}$$

To detect all the anomaly sequences in Set, we propose a hierarchical representation to project all the sequences into a series of multi-domain representation space. Related definitions are given as follows.

Definition 3 (Hierarchical Representation) For the lth sequence $S_l \in Set$, it is projected into D hierarchical multi-domain space to transform corresponding representations, expressed as

$$\mathcal{R}_l = \{\Upsilon_l^1, \cdots, \Upsilon_l^d, \cdots, \Upsilon_l^D\} \tag{4}$$

where the dth Υ_l^d is evenly subdivided into $\lambda(2^d)$ regions, expressed as

$$\Upsilon_l^d = \begin{bmatrix} r_{\lambda,1} \cdots r_{\lambda,j} \cdots r_{\lambda,\lambda} \\ \vdots \quad \ddots \qquad\quad \vdots \\ r_{j,1} \cdots r_{j,j} \cdots r_{j,\lambda} \\ \vdots \quad \cdots \qquad \ddots \ \vdots \\ r_{1,1} \cdots r_{1,j} \cdots r_{1,\lambda} \end{bmatrix}, \tag{5}$$

where $1 \leq j \leq \lambda$ (2^d), each element $r_{j,j} \in \Upsilon_l^d$ contains the amplitude value $m_{j,j}$ and the temporal ratio $u_{j,j}$, expressed as $r_{j,j} = \{m_{j,j}, u_{j,j}\}$. Specially, $m_{j,j}$ is the mean value of the corresponding subsequences with the (j, j)th region, and $u_{j,j}$ denotes the ratio of the current duration to the length of sequence.

With the help of hierarchical representation, all the sequences in $\widetilde{S}$ can be represented by standing on a hierarchical amplitude-temporal view. Thereafter, we propose an anomaly evaluation strategy to find the corresponding abnormal sequences from $\widetilde{S}$, defined as follows.

Definition 4 (Anomaly Sequence) For a given sequence set $\widetilde{S}$, the representations of two sequences (S_p,S_q) are $\mathcal{R}_p$ and $\mathcal{R}_q$, the difference between S_p and S_q can be calculated as

$$Dist(S_p, S_q) = \|\mathcal{R}_p - \mathcal{R}_q\| = \sqrt{\sum_{d=1}^{D}(\left|\Upsilon_p^d - \Upsilon_q^d\right|)^2}, \tag{6}$$

where $1 \leq p, q \leq L$. Therefore, all the sequences in $\widetilde{S}$ can be evaluated in pairs to form the matrix $\mathcal{M}$, expressed as

$$\mathcal{M} = \begin{bmatrix} Dist(S_1,S_1) & Dist(S_1,S_2) & \cdots & Dist(S_1,S_k) & \cdots & Dist(S_1,S_L) \\ \vdots & \ddots & \cdots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \cdots & \cdots & Dist(S_l,S_l) & \cdots & Dist(S_l,S_L) \\ \vdots & \cdots & \cdots & \cdots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \cdots & \cdots & Dist(S_L,S_L) \end{bmatrix}, \tag{7}$$

due to the commutative law of distance calculation (i.e., $Dist(S_p, S_q) = Dist(S_q, S_p)$), $\mathcal{M}$ is an upper triangular matrix. Besides, the distance between S_l and itself is 0 (i.e., $Dist(S_l, S_l) = 0$). Accordingly, the total distance from S_p to other sequences in $\widetilde{S}$, dubbed as anomaly score of S_p, can be calculated as

$$\mathcal{D}_l = \sum_{j=1}^{L}(Dist(S_l, S_j)), \tag{8}$$

without loss of generality, the anomaly threshold is set to ξ, if $\mathcal{D}_l \geq \xi$, S_l is recognized as an anomaly sequence in $\widetilde{S}$.

Our proposed method

In this section, we first introduced our proposed hierarchical representation (HR). Subsequently, we illustrated HR based anomaly sequence detection.

Algorithm 1 Hierarchical representation

```
Input: $T$: time series, $k$: length of sliding window, $D$: the number of
    hierarchical space;
Output: $\widehat{\mathcal{R}}$: HR results of $T$;
 1: for each sequence $S_l$ in $\widetilde{S}$ do
 2:     $v^l_{max} = Maximum(S_l)$;
 3:     $v^l_{min} = Minimum(S_l)$;
 4:     $\mathcal{R}_l = BuildHieSpace(S_l, D, v^l_{max}, v^l_{min})$ //build hierarchical
        multi-domain space $\mathcal{R}_l$ according to Definition 3.
 5:     for each $\Upsilon^d_l$ in $\mathcal{R}_l$ do
 6:         $r_{j,j} = (m_{j,j}, u_{j,j})$; // obtain each representation $r_{j,j}$ according
            to Definition 3.
 7:         insert each $r_{j,j}$ into $\Upsilon^d_l$;
 8:     end for
 9:     complete $\mathcal{R}_l$ ;
10: end for
11: return $\widehat{\mathcal{R}}$; // return hierarchical representation of $T$.
```

Hierarchical representation

According to Definition 1, Definition 2, all the sequences of time series T can be generated by sliding window. To make our point of view more clearly, the part of time series is taken as an example. In Fig. 3a, time series T with length 400 is divided into two sequences S_1 and S_2.

Thereafter, accordingly to Definition 3, each sequence is projected into D hierarchical representation space. As shown in Fig. 3b, S_1 is projected into 3 hierarchical representation space (Υ^1_1, Υ^2_1, and Υ^3_1). The 3-th representation layer is shown in Fig. 3c, containing 16 regions. According to Definition 3, the corresponding amplitude and temporal features of each region can be obtained, shown in Fig. 3d. We take $r_{4,1}$ within Υ^3_1 as an example, $r_{4,1} = (0.85, 0.075)$ denotes the corresponding amplitude value and temporal duration. Obviously, the original sequence S_1 containing 200 data points is transformed into $\mathcal{R}_1$ with 21 elements, each of which contains two scalars (mean value, duration ratio). In other words, with the help of HR, R_1 is compressed to 10.5% ($21 * 2/200 * 2$) compared to the original length, which is undoubtedly reduce the overhead of storage and calculation.

The corresponding pseudo code for HR on T is shown in Algorithm 1. For each sequence S_l, the corresponding boundaries should be determined in the first place (line $2-3$). Secondly, a loop is utilized to project each sequence S_l into the hierarchical space to generate the corresponding representation $\mathcal{R}_l$ (line $4-9$). Thirdly, similar operations are repeated until all the sequences in $\widetilde{S}$ have been represented completely. Finally, time series T is represented as $\widehat{\mathcal{R}}$ for subsequent anomaly evaluation.

Algorithm 2 Hierarchical representation based anomaly detection

```
Input: $\widehat{\mathcal{R}}$: HR results of $T$;
Output: $\mathcal{A}$: anomaly sequence set;
 1: $\mathcal{A} = \emptyset$
 2: $L = sizeof(\widehat{\mathcal{R}})$;
 3: initializes $\mathcal{M}$ filled with 0;
 4: for $p$ in $1 : L$ do
 5:     for $q$ in $(p+1) : L$ do
 6:         $\mathcal{M}[p, q] = Dist(S_p, S_q)$; // calculate the difference between
            $S_p$ and $S_q$ according to Eq. 6.
 7:     end for
 8: end for
 9: $\widehat{\mathcal{D}} = calcD(\mathcal{M})$; // calculate the anomaly score for all the sequences
    in $\widetilde{S}$, according to Eq. 7 and Eq. 8.
10: for $l$ in $1 : L$ do
11:     $\theta_l = \widehat{\mathcal{D}}[l]$;
12:     if $\zeta_l > \xi$ then
13:         insert $S_l$ into $\mathcal{A}$;
14:     end if
15: end for
16: return $\mathcal{A}$; // identify all the anomaly sequences according to Def-
    inition 4.
```

HR for anomaly detection

Subsequently, according to Definition 4, the corresponding anomaly sequences in $\widetilde{S}$ can be detected effectively. The corresponding pseudo code on anomaly detection is shown in Algorithm 2. Concretely, from line 4 to line 9, the upper triangular matrix $\mathcal{M}$ and all the anomaly scores can be generated. Subsequently, from line 10 to line 15, for a certain sequence, whose anomaly score is larger than the threshold (ξ), is selected and inserted into anomaly sequence set $\mathcal{A}$. Finally, $\mathcal{A}$ containing all the anomaly sequences within T can be obtained and Algorithm 2 ends.

Experiment and analysis

In this section, we first introduce the corresponding experimental settings. And then we perform extensive comparison experiments to evaluate the detection performance.

Experimental settings

To evaluate the performance objectively, we select 18 real word time series datasets, including 15 open source datasets

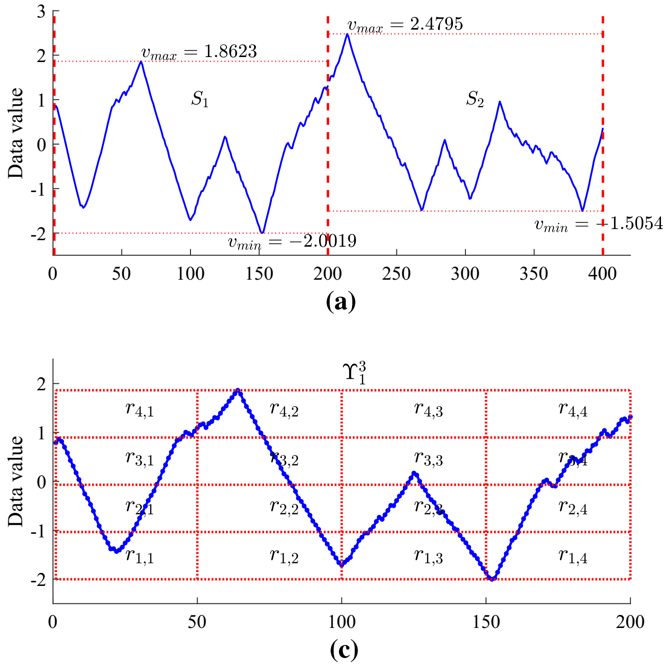


Fig. 3 The details of hierarchical representation on T. **a** Time series T is divided into 2 sequences S_1, S_2 by SW. **b** S_1 is projected into 3 layers hierarchical representation space. **c** Υ_1^3 is composed of 16 regions. **d** Each region of Υ_1^3 can be jointly represented by the corresponding amplitude and temporal features

from UCR Time Series Archive[1] and 3 our collected IIoT time series datasets (Inflow, Outflow, and Totalflow), for comparison experiments. On the other hand, we choose 3 highly cited anomaly detection methods SAX–AD (Keogh et al. 2006), PAA–AD (Keogh et al. 2001) and APAA–AD (Ren et al. 2018) as the baseline methods for the following comparison experiments.

Moreover, to thoroughly measure our method and the baselines, the number of actually detected anomaly sequences ($\mathcal{N}$), and anomaly detection accuracy (Ω), are selected as the evaluation metrics. According to Definition 4, Ω is calculated as follows

$$\Omega = \frac{\mathcal{N}}{N}, \tag{9}$$

where N is the total number of anomaly sequences. According to Definition 4, with the fixed threshold ξ, the larger Ω is, the stronger detection ability of corresponding method has, and vice versa.

Comparison experiments on anomaly detection

To evaluation the anomaly detection performance of HR–AD and three baseline methods, we perform extensive comparison experiments on 18 time series datasets. The corresponding comparison results of 4 methods on 18 datasets are shown in Table 1. L and N denote the total number of sequences in sequence set $\widetilde{S}$, and the number of abnormal sequences in anomaly sequence set $\mathcal{A}$, respectively.

According to the experimental results in Table 1, we have three observations.

1. Compared with other competitors, SAX–AD and PAA–AD deliver inferior detection performance, which can be attributed their temporal representation via either mean-value transformed symbols or roughly mean values, and thus fail to effectively recognize the corresponding anomaly sequences.
2. APAA–AD has relatively higher detection rate Ω than SAX–AD and PAA–AD, which verifies that the effectiveness of amplitude domain representation for anomaly detection.
3. Compared with 3 competitors, HR–AD achieves the highest detection rate Ω. Concretely, the average Ω of HR–AD is at least 7% higher than that of three benchmark methods, which demonstrates the effectiveness of our proposed hierarchical representation for anomaly detection.

Thereafter, we perform the detection efficiency comparison experiments. Concretely, after completing corresponding temporal representations, we utilize them for anomaly detection on 18 datasets. Moreover, to compare the efficiency of these methods more clearly, we set the detection time

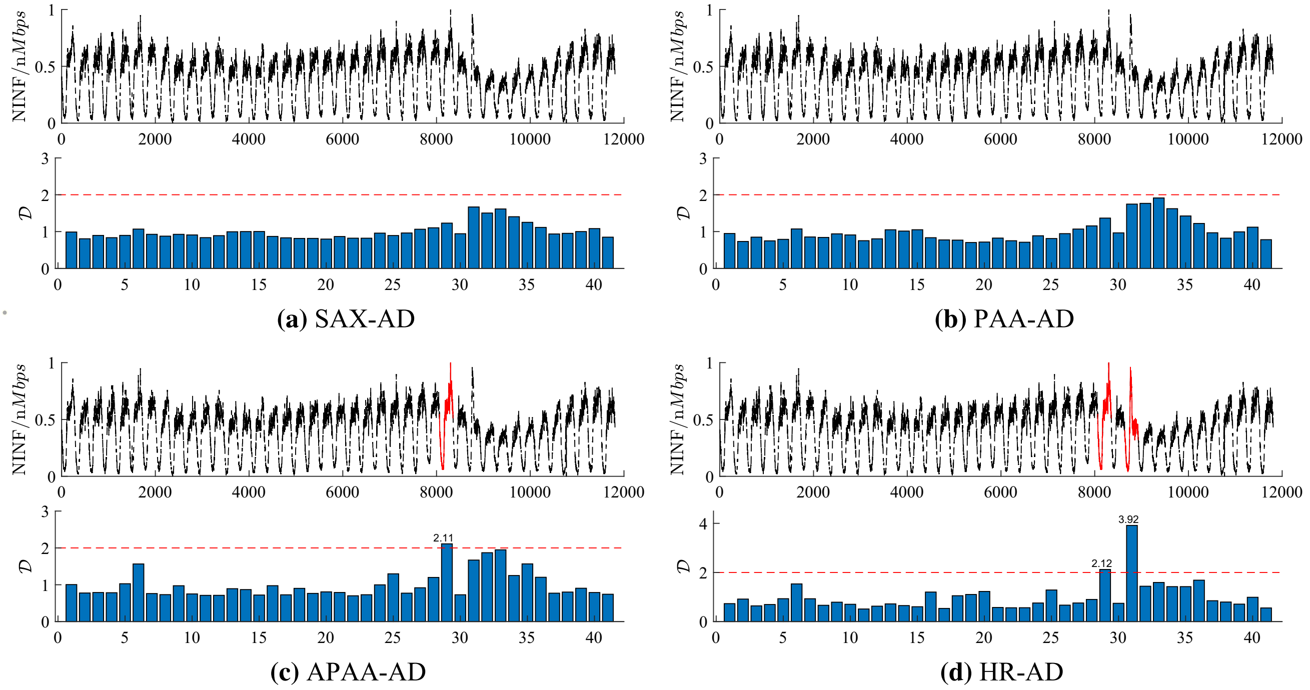
[1] https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Table 1 Comparison experiments on detection accuracy

Dataset	L	N	SAX–AD $\mathcal{N}$	PAA–AD $\mathcal{N}$	APAA–AD $\mathcal{N}$	HR–AD $\mathcal{N}$
BirdChicken	512	9	5	8	8	9
CBF	128	9	8	7	7	7
Computers	720	12	9	9	8	10
Cricket_Z	300	14	9	8	9	11
FaceFour	350	15	11	11	10	13
FacesUCR	131	23	17	16	17	19
Fish	463	22	20	21	19	20
Ham	431	14	6	8	9	10
Inflow	288	2	0	0	1	2
Mallat	1024	20	15	16	17	18
Meat	448	8	4	6	8	6
OliveOil	570	15	14	11	15	12
OSULeaf	427	12	8	8	7	9
Outflow	288	2	1	2	2	2
Plane	144	32	26	24	26	28
Symbols	398	11	10	8	8	11
Totalflow	288	2	1	1	1	2
Yoga	426	14	10	12	12	12
Average Ω			73.7%	74.5%	77.9%	85.1%

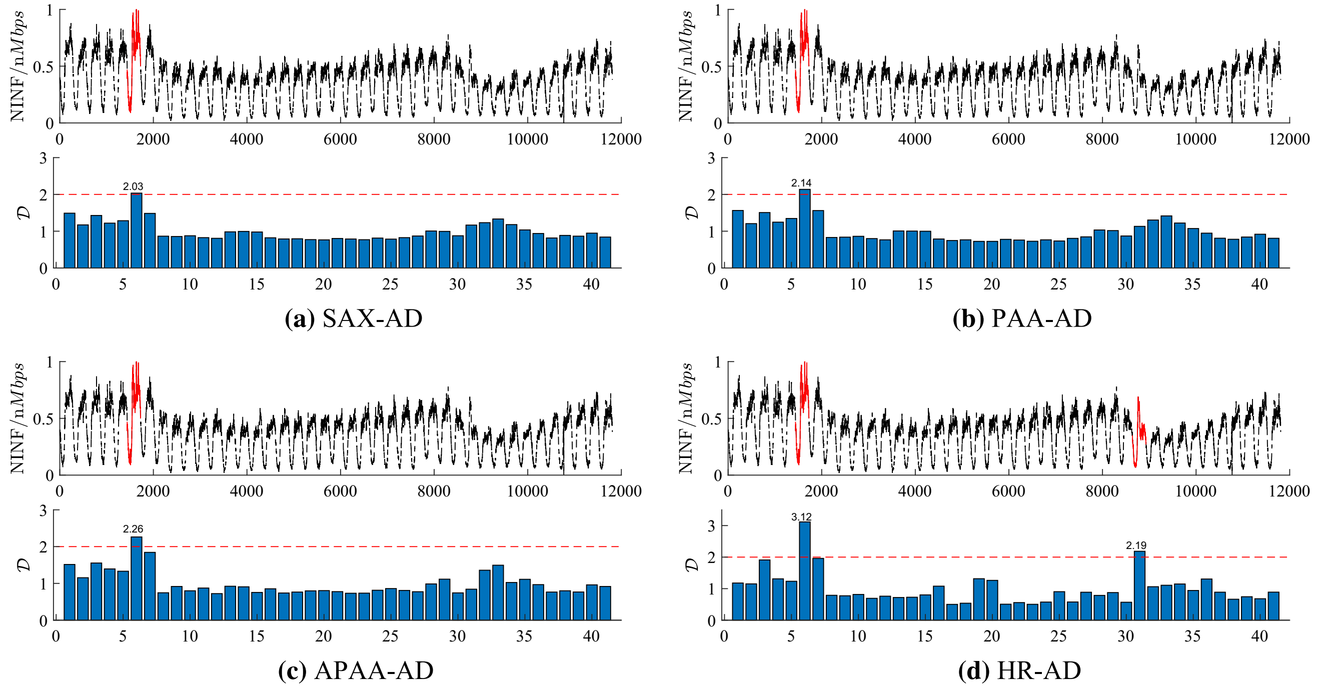
Table 2 Comparison experiments on detection efficiency

Dataset	SAX-AD	PAA-AD	APAA-AD	HR-AD
BirdChicken	1.00	0.82	0.56	0.35
CBF	1.00	0.58	0.76	0.52
Computers	1.00	0.69	0.51	0.29
Cricket_Z	1.00	0.58	0.47	0.31
FaceFour	1.00	0.59	0.66	0.64
FacesUCR	1.00	0.44	0.59	0.25
FISH	1.00	0.11	0.11	0.09
Ham	1.00	0.18	0.19	0.14
Inflow	1.00	0.59	0.44	0.26
MALLAT	1.00	0.17	0.10	0.08
Meat	1.00	0.23	0.14	0.11
OliveOil	1.00	0.21	0.15	0.48
OSULeaf	1.00	0.61	0.59	0.31
Outflow	1.00	0.67	0.45	0.36
Plane	1.00	0.37	0.37	0.23
Symbols	1.00	0.46	0.23	0.39
Totalflow	1.00	0.61	0.46	0.31
Yoga	1.00	0.48	0.56	0.28
Average normalized time	1.00	0.47	0.41	0.30



(a) SAX-AD **(b)** PAA-AD **(c)** APAA-AD **(d)** HR-AD

Fig. 4 Anomaly detection results on Inflow dataset



(a) SAX-AD **(b)** PAA-AD **(c)** APAA-AD **(d)** HR-AD

Fig. 5 Anomaly detection results on Totalflow dataset

of SAX–AD as the benchmark (1.00), and normalize the execution time of the other 3 methods accordingly. The corresponding experimental results are listed in Table 2.

According to the normalized experimental results in Table 2, we can find our proposed HR–AD is more efficient than 3 competitors, which illustrates that HR–AD has stronger detection capability.

To further illustrate the effectiveness of HR–AD, we visualize the detection results of the above four methods on dataset Inflow and Totalflow respectively. In Fig. 4, we present the detection results of 4 methods on Inflow. Specifically, in Fig. 4a, b, the distance based calculation results $\mathcal{D}$s are lower than the threshold. As a result, both SAX–AD and PAA–AD fail to detect anomalies. In Fig. 4c, due to $\mathcal{D}_{29} > \xi$, the sequence S_{29} is detected by APAA–AD. However, the obvious abnormal sequence S_{31} is missed by APAA–AD. As for HR–AD, the sequences S_{29} and S_{31} are detected successfully.

In Fig. 5, we visualize the detection results on Totalflow. Compared to other 3 baselines, the corresponding anomaly scores calculated by HR–AD are much higher than that of other 3 methods. Specially, $\mathcal{D}_6$ and $\mathcal{D}_{31}$ of HR–AD are 3.12 and 2.19, which not only exceeds the threshold $\xi = 2.00$, but also significantly higher than the values calculated by the other 3 baselines under the same conditions.

According to the above experimental analysis, we can find that hierarchical representation based HR–AD has stronger anomaly detection ability than other 3 baselines.

Conclusion

In this paper, we propose an effective hierarchical representation for time series anomaly detection, named HR–AD. It can not only capture the hierarchical amplitude and temporal features to provide more differentiable representation for time series, but also utilize comprehensively evaluation strategy to complete abnormal sequence detection. Extensive comparison experiments on benchmark datasets have demonstrated the superiority of our proposed method. In future, we plan to combine HR–AD with the parallel computing strategy to further improve detection performance on streaming time series.

Acknowledgements The authors would like to thank the anonymous reviewers and the editors for their insightful comments and suggestions, which are greatly helpful for improving the quality of this paper. This work is supported by the National Natural Science Foundation of China, No.: 62002209; the Natural Science Foundation of Shandong Province, No.: ZR2020QF111; the project of CERNET Innovation (NGII20190109); the project of Qingdao Postdoctoral Applied Research.

References

Abdulla, S., & Hashimy, A. S. A. (2018). Tisefe: Time series evolving fuzzy engine for network traffic classification. *International Journal of Communication Networks and Information Security*, *10*, 116–124.

Breunig, MM., Kriegel, H., Ng, RT., & Sander, J. (2000). LOF: identifying density-based local outliers. In *Proceedings of the ACM international conference on management of data* (pp. 93–104).

Cheng, Y., Bi, L., Tao, F., & Ji, P. (2020). Hypernetwork-based manufacturing service scheduling for distributed and collaborative manufacturing operations towards smart manufacturing. *Journal of Intelligent Manufacturing*, *31*, 1707–1720.

Dong, L., Wang, P., & Yan, F. (2019). Damage forecasting based on multi-factor fuzzy time series and cloud model. *Journal of Intelligent Manufacturing*, *30*, 521–538.

Guerrero, J. L., Berlanga, A., García, J., & Molina, J. M. (2010). Piecewise linear representation segmentation as a multiobjective optimization problem. In *Distributed computing and artificial intelligence* (pp. 267–274).

Gupta, M., Gao, J., Aggarwal, CC., & Han, J. (2014). Outlier detection for temporal data. Synthesis Lectures on Data Mining and Knowledge Discovery, Morgan & Claypool Publishers.

Helman, P., & Bhangoo, J. (1997). A statistically based system for prioritizing information exploration under uncertainty. *IEEE Transactions on Systems, Man, and Cybernetics*, *27*, 449–466.

Hsu, C., & Liu, W. (2021). Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, *32*, 823–836.

Huang, S., Guo, Y., Yang, N., Zha, S., Liu, D., & Fang, W. (2020). A weighted fuzzy c-means clustering method with density peak for anomaly detection in iot-enabled manufacturing process. *Journal of Intelligent Manufacturing*, 1–17.

Hu, Y., Ji, C., Zhang, Q., Chen, L., Zhan, P., & Li, X. (2020a). A novel multi-resolution representation for time series sensor data analysis. *Soft Computing*, *24*, 10535–10560.

Hu, Y., Ren, P., Luo, W., Zhan, P., & Li, X. (2019). Multi-resolution representation with recurrent neural networks application for streaming time series in iot. *Computer Networks*, *152*, 114–132.

Hu, Y., Zhan, P., Xu, Y., Zhao, J., Li, Y., & Li, X. (2020b). Temporal representation learning for time series classification. *Neural Computing and Applications*, *32*, 1–14.

Keogh, E., & Smyth, P. (1997). A probabilistic approach to fast pattern matching in time series databases. In *Proceedings of the international conference on knowledge discovery and data mining* (pp. 24–30).

Keogh, E., Lin, J., & Fu, AW. (2005). HOT SAX: Efficiently finding the most unusual time series subsequence. In *Proceedings of the international conference on data mining* (pp. 226–233).

Keogh, E., Chakrabarti, K., Pazzani, M. J., & Mehrotra, S. (2001). Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, *3*, 263–286.

Keogh, E., Lin, J., Fu, A. W., & Herle, H. V. (2006). Finding unusual medical time-series subsequences: Algorithms and applications. *IEEE Transactions Information Technology in Biomedicine*, *10*, 429–439.

Kha, N. H., & Anh, D. T. (2015). From cluster-based outlier detection to time series discord discovery. In *Proceedings of the Pacific-Asia conference on knowledge discovery and data mining* (pp. 16–28).

Leng, M., Yu, W., Wu, S., & Hu, H. (2013). Anomaly detection algorithm based on pattern density in time series. In *Proceedings of the emerging technologies for information systems, computing, and management* (pp. 305–311).

Luo, W., Li, Y., Yao, F., Wang, S., Li, Z., Zhan, P., & Li, X. (2020). Multi-resolution representation for streaming time series retrieval. *International Journal of Pattern Recognition and Artificial Intelligence*, 1–18.

Pang, J., Liu, D., Peng, Y., & Peng, X. (2018). Intelligent pattern analysis and anomaly detection of satellite telemetry series with improved time series representation. *Journal of Intelligent and Fuzzy Systems*, *34*, 3785–3798.

Ren, H., Liao, X., Li, Z., & Al-Ahmari, A. (2018). Anomaly detection using piecewise aggregate approximation in the amplitude domain. *Applied Intelligence*, *48*, 1097–1110.

Sellami, C., Miranda, C., Samet, A., Tobji, M. A. B., & Beuvron, Fd Bd. (2020). On mining frequent chronicles for machine failure prediction. *Journal of Intelligent Manufacturing*, *31*, 1019–1035.

Tsay, R. S., Pena, D., & Pankratz, A. E. (2000). Outliers in multivariate time series. *Biometrika*, *87*, 789–804.

Wei, L., Keogh, E., & Xi, X. (2006). Saxually explicit images: Finding unusual shapes. In *Proceedings of the IEEE international conference on data mining* (pp. 711–720).

Xue, J., Zhou, S., Liu, Q., Liu, X., & Yin, J. (2018). Financial time series prediction using rf-elm. *Neurocomputing*, *277*, 176–186.

Yamanishi, K., Takeuchi, J., Williams, G., & Milne, P. (2004). On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, *8*, 275–300.

Yang, H., Kumara, S., Bukkapatnam, S., & Fg, Tsung. (2019). The internet of things for smart manufacturing: A review. *Institute of Industrial and Systems Engineers Transactions*, *51*, 1190–1216.

Zhan, P., Sun, C., Hu, Y., Luo, W., Zheng, J., & Li, X. (2020). Feature-based online representation algorithm for streaming time series similarity search. *International Journal of Pattern Recognition and Artificial Intelligence*, *34*, 1–25.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.