Intelligent Data Analysis 26 (2022) 1451–1467
DOI 10.3233/IDA-216128
IOS Press

Unsupervised feature extraction from multivariate time series for outlier detection

Kiyotaka Matsue[a,b,c,*] and Mahito Sugiyama[a,b]
[a]*National Institute of Informatics, Tokyo, Japan*
[b]*The Graduate University for Advanced Studies, SOKENDAI, Kanagawa, Japan*
[c]*Toshiba Infrastructure Systems & Solutions Corporation, Kanagawa, Japan*

Abstract. Although various feature extraction algorithms have been developed for time series data, it is still challenging to obtain a flat vector representation with incorporating both of time-wise and variable-wise association between multiple time series. Here we develop an algorithm, called Unsupervised Feature Extraction using Kernel and Stacking (UFEKS), that constructs feature vector representation for multiple time series in an unsupervised manner. UFEKS constructs a kernel matrix for the set of subsequences from each time series and *horizontally concatenates* all matrices. Then we can treat each row as a feature vector representation of its corresponding subsequence of times series. We examine the effectiveness of the extracted features under the unsupervised outlier detection scenario using synthetic and real-world datasets, and show its superiority compared to well-established baselines.

Keywords: Feature extraction, unsupervised outlier detection, multivariate time series, kernel method

1. Introduction

Internet of things (IoT) devices, composed of many sensors such as temperature, humidity, and pressure, are installed in various types of systems, and multivariate time series data are being collected in a wide range of fields. For example, in the task of facility maintenance for a building, it may be possible to know the best timing of their replacement by monitoring and analyzing collected data. Although this process, called condition based maintenance (CBM) [12], is well known technology in industrial fields, this task is still challenging as we need to use many multivariate time series to find relationship between sensors. In the case of sensing at multiple locations in a building, sensors located in the same local area are expected to record similar values. If one sensor takes different values from others, the sensor might be broken and need to be replaced to the new one. However, it is hard to find the sensor taking different values because we need to find different combinatorial relationships between sensors.

In this paper, we consider *association between multiple time series*. An example of association is shown in Fig. 1. There are two time series from the corresponding variables composed of lines and sine waves with noise. In an orange frame, the time series is composed of line and sine wave (up and bottom), while the other subsequences out of the orange frame are composed of the combination of only lines or sine waves. Hence only the subsequence in the orange frame has a different combination. This is an example of a combinatorial outlier, and finding such outliers is fundamentally difficult as *they cannot be found if*

*Corresponding author: Kiyotaka Matsue, National Institute of Informatics, Tokyo, Japan. E-mail: matsue@nii.ac.jp.

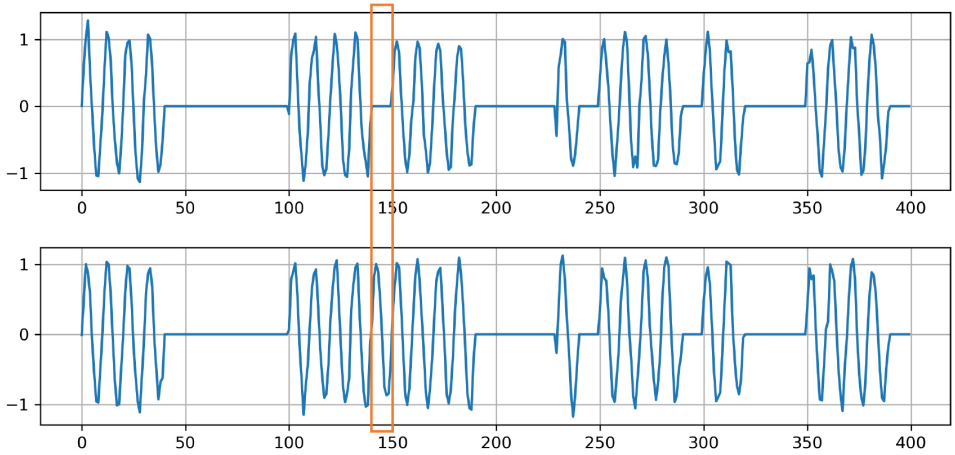


Fig. 1. An example of association for multivariate time series data.

we look at each time series separately. In the case of CBM, if building managers find sudden changes of values like the orange frame, they are probably considered as signals of equipment failures and managers can start to inspect their equipment in more detail. Finding a different combination from multivariate time series data is one of the most important tasks in a number of sensor monitoring tasks including CBM.

In this paper, we focus on the task of extracting feature vector representation from multivariate time series data that can incorporate combinatorial association between two or more time series. This approach is unsupervised, hence it enables us to apply general conventional machine learning algorithms to multivariate time series.

To extract feature vectors from multivariate time series, we propose a new algorithm, called UFEKS (Unsupervised Feature Extraction using Kernel and Stacking), and apply it for outlier detection. The proposed method UFEKS uses a kernel method to extract features from multivariate time series. It first divides a given time series into a set of its subsequences, and makes a kernel matrix from each univariate time series. Then it *horizontally concatenates* all of the kernel matrices. Our idea is to treat each row in the concatenated matrix as a feature vector, which means that each row corresponds to a point in the multidimensional Euclidean space. Our approach allows us to use standard outlier detection algorithms to detect outliers from multivariate time series with considering combinatorial association between two time series. We examine the proposed method using unsupervised outlier detection as it is commonly used in practical situations such as CBM. It should be noted that *our proposed method does not use any labeled data* for detecting outliers from multivariate time series. Moreover, the UFEKS can be employed for not only outlier detection but other data mining tasks.

To summarize, the main contributions of our work are:

- Our proposed method UFEKS can extract features from multivariate time series with incorporating combinatorial association between time series. The obtained features can be applied to a variety of applications such as outlier detection or other data mining tasks.
- In outlier detection, the proposed method can detect outliers that cannot be found if we look at each of multivariate time series separately.

The remainder of this paper is organized as follows: In Section 2, we discuss related work of our research in terms of feature extraction and outlier detection for time series. In Section 3, we present our algorithm UFEKS and explain combination with outlier detection techniques as one of the applications. In

Section 4, we examine performance of UFEKS on synthetic and real-world datasets. Finally, we conclude this paper by including some remarks in Section 5.

2. Related work

To date, several feature extraction algorithms from time series for outlier detection have been developed. *Discrete Fourier Transform* (DFT), *Discrete Wavelet Transform* (DWT), and *Discrete Cosine Transformation* (DCT) are well known algorithms to extract features from time series used in signal processing fields and data mining fields [3,16,23]. Other algorithms such as *Piece-wise Aggregate Approximation* [10,14,33], *Symbolic Approximation* [13], and fundamental statistics like mean and variance are widely used in data mining fields. However, the above algorithms mainly used for signal processing cannot be directory applied to multivariate time series. Kernel methods are also used to extract features from time series [9,35]. For example, a kernel matrix using the *Radial Basis Function* (RBF) kernel [9] or the linear kernel [35] from a time series has been used for outlier detection. However, since these approaches use only the integrated signal across multiple time series, they cannot treat combinatorial association of time series. In contrast, our proposal treats each time series separately when we apply kernels, hence we can treat such combinatorial effects.

Outlier detection for time series have been actively studied and a number of methods have been proposed [1,9,17,22,25,28,30]. In particular for outlier detection from a univariate time series, *autoregressive moving average* (ARMA) and an *autoregressive integrated moving average* (ARIMA) have been commonly used [28]. They can find outliers from differences between predicted values by ARMA or ARIMA and actual values. However, they are considered to be sensitive to noise, resulting in increasing false positives when the noise level is severe [28,35]. *Dynamic time warping* (DTW) is another representative method, which measures similarity between two time series by aligning them [6,21,26]. DTW is widely used in a variety of applications because it is robust to different frequencies or lengths. DTW can be used for univariate time series, however, it may not be directly used for multivariate time series.

In terms of outlier detection from multivariate time series, several algorithms have been proposed [9, 17,30]. Takeishi and Yairi [30] have proposed an algorithm using sparse representation. This method is designed in a supervised manner and requires labeled data. Moreover, they do not focus on combinatorial association between time series and may not detect combinatorial outliers.

Nowadays, a number of outlier detection methods for time series have been proposed based on neural networks [20,34–37]. One of the outlier detection methods for multivariate time series is the *Multi-Scale Convolutional Recurrent Encoder-Decoder* (MSCRED), which is an algorithm using attention-based convolutional long-short time memory [35]. It extracts features and detects outliers from multivariate time series by constructing a kernel matrix. Many algorithms based on neural networks are usually useful and widely used in real world. However, most of neural network based models have many parameters to be tuned, which is fundamentally difficult in the unsupervised setting. Furthermore, they are often designed as supervised, hence ground-truth labels are required to train their models to perform outlier detection.

Numerous algorithms have been proposed so far for outlier detection for non-time series data [1,2, 8,11,27,32,36,37]. Representative algorithms include *local outlier factor* (LOF), *κth-nearest neighbor* (κNN) [15], ORCA [5], *one-class support vector machine* (OCSVM) [19], *isolation forest* (iForest) [18], sampling-based outlier detection [29], and *self-organizing maps (SOM)* [24]. Furthermore, heatmaps might be also one of the options to detect outliers because one can detect outliers by visual inspection of a heatmaps representing the reconstruction errors resulted from an autoencoder. Although the above algorithms have been widely used in a variety of fields, they fundamentally assume i.i.d. data and cannot

be applied to time series directly. Our method extracts non-time series feature vectors and allows us use the above methods for outlier detection from time series.

3. The UFEKS algorithm and outlier detection

We formulate our method UFEKS in Section 3.1, which extracts feature vectors from multivariate time series, and introduce its application to outlier detection in Section 3.2.

3.1. Extraction of features from multivariate time series

Many algorithms to extract features from time series have focused on its subsequences [9,17,30,35]. In this paper, we follow the idea of using subsequences and extract features based on the similarity between subsequences. We use a kernel method to measure the similarity between subsequences as it is widely used in data analysis for time series and its effectiveness is well known.

Assume that there are P variables indexed from 1 to P. Given a multivariate time series with the length T as a matrix $\mathbf{X} = (x_{ij}) \in \mathbb{R}^{P\times T}$, where each row vector $\mathbf{x}^{(p)} = (x_{p1}, x_{p2}, \cdots, x_{pT}) \in \mathbb{R}^T$ represents a time series of a variable p and each column vector $\mathbf{x}_t = (x_{1t}, x_{2t}, \cdots, x_{Pt})^\mathsf{T} \in \mathbb{R}^P$ represents a multivariate vector at a time stamp t. A submatrix $\mathbf{X}_t \in \mathbb{R}^{P\times w}$, which is a part of $\mathbf{X}$ with respect to time stamps from t to $t+w-1$, is denoted as

$$\mathbf{X}_t = \begin{bmatrix} x_{1t} & x_{1(t+1)} & \cdots & x_{1(t+w-1)} \\ x_{2t} & x_{2(t+1)} & \cdots & x_{2(t+w-1)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{Pt} & x_{P(t+1)} & \cdots & x_{P(t+w-1)} \end{bmatrix}, \tag{1}$$

where each row $\mathbf{x}_t^{(p)} = (x_{pt}, x_{p(t+1)}, \cdots, x_{p(t+w-1)}) \in \mathbb{R}^w$ represents a subsequence at t of the p-th time series with length w.

First, we consider extraction of feature vectors from a univariate time series $\mathbf{x}^{(p)}$. Given two subsequences of p-th univariate time series $\mathbf{x}_i^{(p)}, \mathbf{x}_j^{(p)} \in \mathbb{R}^w$ with the length w. We use the RBF kernel to obtain the similarity between them, which is given as

$$k_{ij}^{(p)} = \exp\left\{-\frac{\sum_{s=0}^{w-1}(x_{p(i+s)} - x_{p(j+s)})^2}{\sigma^2}\right\}, \tag{2}$$

where $\sigma \in \mathbb{R}$ is a parameter. Every $k_{ij}^{(p)}$ takes a value in $(0, 1]$. The RBF kernel for similarity computation between subsequences was first employed in [9] and we follow this strategy. The resulting kernel matrix $\mathbf{K}^{(p)} \in \mathbb{R}^{(T-w+1)\times(T-w+1)}$ becomes a non-negative square matrix given as

$$\mathbf{K}^{(p)} = \begin{bmatrix} k_{11}^{(p)} & k_{12}^{(p)} & \cdots & k_{1(T-w+1)}^{(p)} \\ k_{21}^{(p)} & k_{22}^{(p)} & \cdots & k_{2(T-w+1)}^{(p)} \\ \vdots & \vdots & \ddots & \vdots \\ k_{(T-w+1)1}^{(p)} & k_{(T-w+1)2}^{(p)} & \cdots & k_{(T-w+1)(T-w+1)}^{(p)} \end{bmatrix}, \tag{3}$$

where we denote each row vector as $\mathbf{k}_t^{(p)} = (k_{t1}^{(p)}, k_{t2}^{(p)}, \cdots, k_{t(T-w+1)}^{(p)}) \in \mathbb{R}^{(T-w+1)}$ and T is the length of time series. Each row $\mathbf{k}_t^{(p)}$ of the kernel matrix $\mathbf{K}^{(p)}$ is a feature vector representation of the subsequence $\mathbf{x}_t^{(p)}$, and it incorporates association between $\mathbf{x}_t^{(p)}$ and all the other subsequences.

Now we extend our feature vector representation for univariate time series to multivariate time series. First we generate kernel matrices $\mathbf{K}^{(1)}, \mathbf{K}^{(2)}, \ldots, \mathbf{K}^{(P)}$ for all variables $1, 2, \ldots, P$. Then, we horizontally concatenate all kernel matrices with each other and generate a single matrix. The resulting matrix $\mathbf{K} \in \mathbb{R}^{(T-w+1)\times(T-w+1)P}$ for multivariate time series is given as

$$\begin{aligned}\mathbf{K} &:= [\mathbf{K}^{(1)}, \cdots, \mathbf{K}^{(P)}] \\ &= \begin{bmatrix} k^{(1)}_{11} & \cdots & k^{(1)}_{1(T-w+1)} & \cdots & k^{(P)}_{11} & \cdots & k^{(P)}_{1(T-w+1)} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ k^{(1)}_{(T-w+1)1} & \cdots & k^{(1)}_{(T-w+1)(T-w+1)} & \cdots & k^{(P)}_{(T-w+1)1} & \cdots & k^{(P)}_{(T-w+1)(T-w+1)} \end{bmatrix}\end{aligned} \tag{4}$$

and its row vector $\mathbf{k}_t \in \mathbb{R}^{(T-w+1)P}$ is given as

$$\mathbf{k}_t = (k^{(1)}_{t1}, \cdots, k^{(1)}_{t(T-w+1)}, \cdots, k^{(P)}_{t1}, \cdots, k^{(P)}_{t(T-w+1)}). \tag{5}$$

This matrix $\mathbf{K}$ is considered to be the set of feature vectors $\{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_{T-w+1}\}$. We treat each row $\mathbf{k}_t$ as a feature vector representation of the corresponding multivariate subsequence $\mathbf{X}_t$, which is expected to encode the association between variables with respect to the subsequence from t to $t + w - 1$.

The pseudo code of our method UFEKS is shown in Algorithm 1.

Algorithm 1: The UFEKS algorithm

Input: $\mathbf{X} \in \mathbb{R}^{P\times T}, w, \sigma, R \in \mathbb{R}$
Output: $\mathbf{f}_1, \ldots, \mathbf{f}_{T-w+1} \in \mathbb{R}^R$
// Construct kernel matrices from multivariate time series
for $p = 1$ to P **do**
  for $(i, j) = (1, 1)$ to $(T - w + 1, T - w + 1)$ **do**
    $k^{(p)}_{ij} \leftarrow \exp\{ -\sum_{s=0}^{w-1} (x_{p(i+s)} - x_{p(j+s)})^2 / \sigma^2 \}$
  end for
end for
// Feature Extraction from Kernel Matrices
Construct $\mathcal{K} \in \mathbb{R}^{P\times(T-w+1)\times(T-w+1)}$ from $\mathbf{K}^{(1)}, \ldots, \mathbf{K}^{(P)}$
$(\mathcal{C}, \mathbf{F}^{(1)}, \mathbf{F}^{(2)}, \mathbf{F}^{(3)}) \leftarrow \mathrm{Tucker}(\mathcal{K}, [P, R, R])$
for $i = 1$ to $T - w + 1$ **do**
  $\mathbf{f}_i \leftarrow i$th-row vector of $\mathbf{F}^{(2)}$
end for
return $\mathbf{f}_1, \ldots, \mathbf{f}_{T-w+1}$

3.2. Outlier detection from multivariate time series

We propose to apply our method UFEKS to the problem of outlier detection from multivariate time series. When we use our method for a multivariate time series $\mathbf{X}$, we can extract feature vectors for subsequences. Then we can directly perform outlier detection on the extracted vectors to find outlier subsequences. In this paper, we use κNN [15], LOF, OCSVM [19], iForest [18], which are popular outlier detection algorithms. In the following, we briefly summarize the outlier detection problem and the κNN algorithm as one of examples.

To detect outliers from multivariate time series $\mathbf{X}$, we measure the outlierness of a subsequence $\mathbf{X}_t$, denoted as $q(\mathbf{X}_t)$. When we construct the matrix $\mathbf{K}$ via UFEKS, the score $q(\mathbf{X}_t)$ is obtained as

$$q(\mathbf{X}_t) = d^{\kappa}(\mathbf{k}_t; \mathcal{S}_{\mathrm{fvec}}), \tag{6}$$

where $d^{\kappa}(\mathbf{k}_t; \mathcal{S}_{\mathrm{fvec}})$ is Euclidean distance from $\mathbf{k}_t \in \mathbb{R}^{(T-w+1)P}$, which is the feature vector representation of $\mathbf{X}_t$ obtained by UFEKS, to its *κth-nearest neighbor* (κNN) in the set $\mathcal{S}_{\mathrm{fvec}} = \{\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_{T-w+1}\}$.

4. Experiments

We empirically evaluate our algorithm on synthetic and real-world datasets.

4.1. Comparison partners

We compare our algorithm with two feature representation algorithms, the *PageRank kernel* (PRK), and *Subsequence* (SS), under four different unsupervised outlier detection algorithms, κNN, LOF, OCSVM, and iForest, which are popular and widely used in data analysis. The PageRank kernel, which we denote by PRK, has been proposed in the outlier detection method PR [9], which is considered to be the state-of-the-art technique for unsupervised outlier detection from multivariate time series. PR is a kernel-based method using the PageRank algorithm. It constructs a state transition probability matrix converted from a kernel matrix calculated by the RBF kernel, and the state transition probability matrix is used for the PageRank algorithm to detect outliers. Given a multivariate time series $\mathbf{X} \in \mathbb{R}^{(P\times T)}$ with P variables with the length T, PR constructs a kernel matrix $K \in \mathbb{R}^{(T-w+1)\times(T-w+1)}$ such that

$$\begin{aligned} &k_{ij} = \exp\left\{-\frac{\sum_{p=1}^{P}\sum_{s=0}^{w-1}(x_{i+s}^{(p)} - x_{j+s}^{(p)})^2}{\sigma^2}\right\}, \\ &i,j \in \{1,2,\ldots,T-w+1\}, \end{aligned} \tag{7}$$

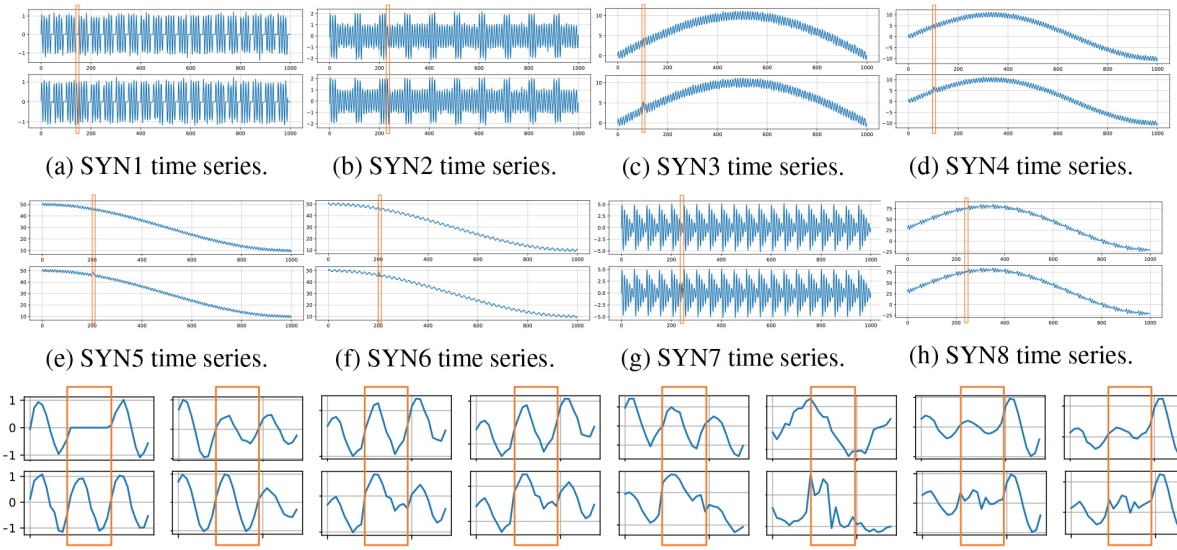
where w is the length of each subsequence. The difference between this kernel and our kernel function in Eq. (2) is whether or not values representing associations between subsequences calculated for each variable are summed up. In the case of Eq. (7), information of association among variables may be lost by its summation. After obtaining the kernel matrix defined in Eq. (7), PR constructs a weighted graph $G = (V, E)$, where V corresponds to the set of subsequences, and use the PageRank algorithm on the graph to quantify the outlierness of each subsequence, where anomalous subsequences will receive low score in the method. We considered that the kernel matrix, PRK, was effective for feature extraction from time series and hence added in our experiments for comparison with our algorithm. In comparison with PRK, the subsequence based approach, which we denote by SS, is widely used in extracting features from time series. A subsequence is a sequence extracted from time series. Given $\mathbf{X} = (x_{ij}) \in \mathbb{R}^{P\times T}$ with P variables with the length T, first we obtain a single time series $\mathbf{x}_{ss}$ by summing up every variable at each time t,

$$\mathbf{x}_{ss} = \left(\sum_{p=1}^{P} x_1^{(p)}, \ldots, \sum_{p=1}^{P} x_T^{(p)}\right). \tag{8}$$

When we denote each element of $\mathbf{x}_{ss}$ at time t as $x_t^{ss} = \sum_{p=1}^{P} x_t^{(p)}$, the subsequence based matrix $\mathbf{X}_{ss} \in \mathbb{R}^{(T-w+1)\times w}$ with the window size w is defined as

$$\mathbf{X}_{ss} = \begin{bmatrix} x_1^{ss} & \cdots & x_w^{ss} \\ x_2^{ss} & \cdots & x_{w+1}^{ss} \\ \vdots & \ddots & \vdots \\ x_{T-w+1}^{ss} & \cdots & x_T^{ss} \end{bmatrix}. \tag{9}$$

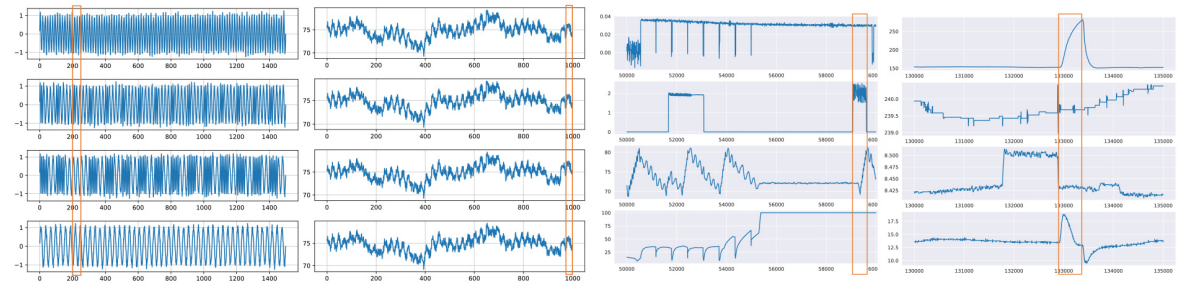
By considering each row in Eq. (9) to be a multidimensional data point, outliers can be detected by conventional algorithms like κNN. We compare our algorithm with PRK and SS.



(a) SYN1 time series. (b) SYN2 time series. (c) SYN3 time series. (d) SYN4 time series.

(e) SYN5 time series. (f) SYN6 time series. (g) SYN7 time series. (h) SYN8 time series.

(i) The plots from left to right correspond to enlarged plots between specific time spans that include outliers from SYN1 to SYN8 time series.



(j) SYN9 time series. (k) ATSF time series. (l) WADI time series. (m) SWaT time series.

Fig. 2. Examples of synthetic and real-world datasets.

4.2. *Datasets*

We prepared nine types of synthetic multivariate time series datasets and six types of real-world multivariate time series datasets shown in Fig. 2 and Table 1. Each synthetic dataset includes two or more time series and outlierness behavior occurs in only one of time series, which are shown as an orange solid area in Figures from 2a to j. Those outliers have simulated spike noises in the real-world. All the nine synthetic datasets are composed of sine waves or straight lines, and Gaussian noise were added to every data point, where noise were generated by Gaussian distribution with zero mean and 0.1 standard deviation $\mathcal{N}(0, 0.1^2)$. Those noises have simulated the real-world datasets that are taken by sensors like temperature. Any missing values does not included in the datasets because we assumed that any other troubles like losing network connection between data collection system and every sensors had not been occurred. Furthermore, we prepared ten patterns of each synthetic dataset as we change noise pattern because of evaluating accuracy of detecting outliers for our algorithm.

Eight figures from Fig. 2a–h illustrate synthetic multivariate time series datasets, each of which is composed of two time series. Each time series has 1,000 time stamps, and outliers with the length of ten

Table 1
Summary of datasets. Datasets between SYN1 and SYN9 are synthetic, and ATSF8, ATSF16, ATSF32, ATSF64, WADI, and SWaT are real-world datasets

Name of datasets	Number of variables	Range of outliers	Length of time series
SYN1	2	140–150	1,000
SYN2	2	231–240	1,000
SYN3	2	101–110	1,000
SYN4	2	101–110	1,000
SYN5	2	201–210	1,000
SYN6	2	201–210	1,000
SYN7	2	241–250	1,000
SYN8	2	241–250	1,000
SYN9	4	201–250	1,500
ATSF8	8	951–1,000	1,000
ATSF16	16	951–1,000	1,000
ATSF32	32	951–1,000	1,000
ATSF64	64	951–1,000	1,000
WADI	93	9,054–9,644	10,000
SWaT	39	2,918–3,380	5,000

or eleven time stamps are injected. SYN1 time series in Fig. 2a is composed of sine waves and straight line. SYN2 time series in Fig. 2b is composed of two sine waves with different amplitude. Datasets from SYN3 to SYN6 illustrated in Fig. 2c–f are composed of sine waves, and their averages in subsequence are swaying over time. Moreover, a phase shift occurs between their time series in SYN6. SYN7 time series in Fig. 2g is composed of sine waves with different amplitude. SYN8 time series is almost the same as SYN7 except for changes of their averages over time. Enlarged plots between specific time spans that include outliers from SYN1 to SYN8 are shown in Fig. 2i. The SYN9 dataset in Fig. 2j has a set of four time series and each time series has 1,500 time stamps. Their wavelengths of the top and the bottom time series are fifteen and thirty, respectively. The second time series is combined with two wavelengths of ten and twenty. Similarly, the third time series is combined with two wavelengths of ten and twenty-five. Consequently, all of four time series are composed of different wavelengths.

Real-world datasets called *ambient temperature system failure* (ATSF) are shown in Fig. 2k. Since it is hard to find ground truth combinatorial outliers in multivariate time series from real-world datasets, we collected univariate time series and artificially simulated combinatorial outliers on it. The dataset [4][1] comes from the Numenta Anomaly Benchmark (NAB) v1.1, which is publicly available. We used one of the real-world univariate time series called ATSF, which was ambient temperature in an office setting measured every hour. Although several types of datasets including outliers are available, most of such outliers can be easily detected by checking each time series separately, hence they are not appropriate for our evaluation. Time series we extracted have successive 1,000 time stamps out of 7,267 where it corresponds between November 1, 2013 and December 13, 2013, and we created 8, 16, 32, and 64 variants with adding noise generated by Gaussian distribution with $\mathcal{N}(0, 0.1^2)$. Furthermore, we artificially injected outliers in the range from 951 to 1,000 by adding about one percent values of the original datasets to only one of the variables. Their outliers simulate abnormal drift of a temperatures sensor in the period of time. We consider that those injection does not bring about any bias because of just adding small values to the original datasets and directly irrelevant to subsequence length. Note that it is difficult to detect such outliers if one checks each time series separately. In the same as synthetic datasets, we created ten patterns of each ATSF dataset.

[1] https://github.com/numenta/NAB/tree/master/data.

Table 2
Parameters for algorithms

Name	Value
κ for κ-th nearest neighbor (κNN)	5
σ for PageRank kernel (PRK)	1
Length of subsequence (SS)	2

Furthermore, we employed another types of multivariate time series called *Water Distribution* (WADI)[2] illustrated in Fig. 2l and *Secure Water Treatment* (SWaT)[3] illustrated in Fig. 2m from *Singapore University of Technology and Design*. Those real-world multivariate time series datasets are also publicly available and outliers have been already included in them. We extracted successive 10,000 out of 172,801 time stamps between 50,001 and 60,000 from WADI datasets, and successive 5,000 out of 449,919 time stamps between 130,000 and 135,000 from SWaT datasets. Their subsets include some outliers that can be obviously and visually identified as outliers. Note that, in comparison with ATSF, we do not artificially inject outliers in both WADI and SWaT datasets.

4.3. Environment

We used CentOS release 6.10 with 4x 22-Core model 2.20 GHz Intel Xeon CPU E7-8880 v4 processors and 3.18 TB memory. All methods are implemented in Python 3.7.6 and all experiments are also performed in the same platform.

4.4. Experimental results

We performed three feature representation algorithms: UFEKT, PageRank kernel (PRK), and subsequence (SS), combined with four outlier detection algorithms: κNN, LOF, OCSVM, and iForest, resulting in twelve combinations of feature extraction and outlier detection in total. In addition to them, we tried to perform one of the popular algorithms called Prophet [31], which is a forecasting procedure for univariate time series. However, we could not get results of Prophet due to high computational cost and it is hard to decide date and time information correctly for our datasets, which is required for Prophet as additional input. We show each parameter that we used in the algorithms in Table 2. The parameters κ and σ are used for κNN and PRK, respectively. We set $\kappa = 5$, which is commonly used in literature [5,7], and set $\sigma = 1$, which is known to be an appropriate value for normalized datasets. The length of subsequences, or the window size, is used in not only SS but all algorithms to convert a given time-series to a set of subsequences. The window size was set to be two to avoid low resolution and to improve accuracy of outlier detection. If the window size increases further, it becomes low resolution, resulting in low accuracy. The effectiveness of each method was evaluated by the *area under precision-recall curve* (AUPRC) [1]. The AUPRC score takes values between zero and one, and higher is better.

Results are summarized in Figs 3 and 4, and Tables 3 and 4. OD, FR, PRK, and SS in the figures stand for Outlier Detection, Feature Representation, PageRank Kernel, and SubSequences, respectively. The datasets except for WADI and SWaT are prepared ten patterns for each as we inject Gaussian noises and Mean $\pm$ standard deviation in ten trials are shown in the table. Best scores are denoted in bold.

4.4.1. Synthetic datasets

Figures from Figs 3a to 5a show results of synthetic datasets. There are four plots in each figure

[2]https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_wadi/.
[3]https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_swat/.

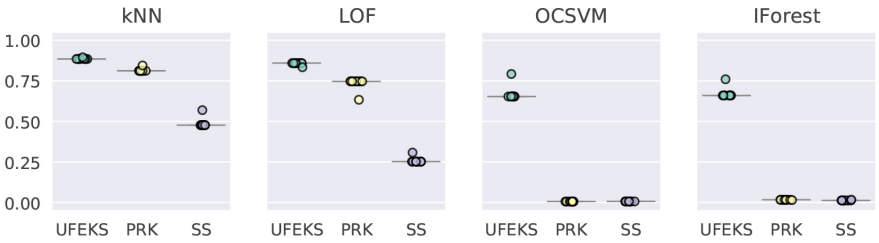
Table 3
Area under precision-recall curve (AUPRC) for synthetic datasets

OD		κNN			LOF	
FR	UFEKS	PRK	SS	UFEKS	PRK	SS
SYN1	**0.886** $\pm$ 0.003	0.815 $\pm$ 0.010	0.486 $\pm$ 0.028	0.857 $\pm$ 0.008	0.736 $\pm$ 0.034	0.258 $\pm$ 0.017
SYN2	**0.913** $\pm$ 0.054	0.734 $\pm$ 0.048	0.575 $\pm$ 0.055	0.803 $\pm$ 0.043	0.062 $\pm$ 0.031	0.380 $\pm$ 0.122
SYN3	**0.980** $\pm$ 0.013	0.625 $\pm$ 0.167	0.016 $\pm$ 0.005	0.953 $\pm$ 0.028	0.118 $\pm$ 0.041	0.011 $\pm$ 0.002
SYN4	**0.956** $\pm$ 0.028	0.573 $\pm$ 0.111	0.010 $\pm$ 0.002	0.901 $\pm$ 0.017	0.114 $\pm$ 0.073	0.011 $\pm$ 0.003
SYN5	**0.990** $\pm$ 0.005	0.046 $\pm$ 0.009	0.007 $\pm$ 0.001	0.957 $\pm$ 0.028	0.006 $\pm$ 0.000	0.011 $\pm$ 0.003
SYN6	**0.237** $\pm$ 0.068	0.063 $\pm$ 0.019	0.182 $\pm$ 0.051	0.023 $\pm$ 0.009	0.007 $\pm$ 0.001	0.139 $\pm$ 0.043
SYN7	0.853 $\pm$ 0.012	0.779 $\pm$ 0.057	0.010 $\pm$ 0.001	0.867 $\pm$ 0.031	0.047 $\pm$ 0.029	0.060 $\pm$ 0.069
SYN8	**0.739** $\pm$ 0.074	0.012 $\pm$ 0.001	0.006 $\pm$ 0.000	0.006 $\pm$ 0.000	0.007 $\pm$ 0.000	0.007 $\pm$ 0.000
SYN9	**0.375** $\pm$ 0.026	0.368 $\pm$ 0.041	0.033 $\pm$ 0.001	0.140 $\pm$ 0.020	0.069 $\pm$ 0.020	0.035 $\pm$ 0.002
Average	**0.770**	0.446	0.147	0.612	0.129	0.101
OD		OCSVM			IForest	
FR	UFEKS	PRK	SS	UFEKS	PRK	SS
SYN1	0.668 $\pm$ 0.042	0.006 $\pm$ 0.000	0.007 $\pm$ 0.000	0.670 $\pm$ 0.030	0.017 $\pm$ 0.000	0.013 $\pm$ 0.001
SYN2	0.600 $\pm$ 0.036	0.006 $\pm$ 0.000	0.018 $\pm$ 0.000	0.303 $\pm$ 0.143	0.006 $\pm$ 0.000	0.019 $\pm$ 0.001
SYN3	0.418 $\pm$ 0.105	0.006 $\pm$ 0.000	0.008 $\pm$ 0.000	0.017 $\pm$ 0.005	0.006 $\pm$ 0.000	0.011 $\pm$ 0.001
SYN4	0.508 $\pm$ 0.109	0.006 $\pm$ 0.000	0.007 $\pm$ 0.000	0.014 $\pm$ 0.001	0.006 $\pm$ 0.000	0.007 $\pm$ 0.000
SYN5	0.545 $\pm$ 0.114	0.006 $\pm$ 0.000	0.010 $\pm$ 0.000	0.022 $\pm$ 0.009	0.006 $\pm$ 0.000	0.008 $\pm$ 0.001
SYN6	0.008 $\pm$ 0.002	0.007 $\pm$ 0.001	0.008 $\pm$ 0.000	0.007 $\pm$ 0.000	0.007 $\pm$ 0.000	0.014 $\pm$ 0.005
SYN7	**0.894** $\pm$ 0.013	0.006 $\pm$ 0.000	0.006 $\pm$ 0.000	0.374 $\pm$ 0.049	0.007 $\pm$ 0.000	0.006 $\pm$ 0.000
SYN8	0.331 $\pm$ 0.075	0.012 $\pm$ 0.003	0.014 $\pm$ 0.000	0.162 $\pm$ 0.026	0.051 $\pm$ 0.016	0.006 $\pm$ 0.000
SYN9	0.117 $\pm$ 0.021	0.020 $\pm$ 0.001	0.032 $\pm$ 0.001	0.070 $\pm$ 0.020	0.030 $\pm$ 0.002	0.032 $\pm$ 0.001
Average	0.454	0.008	0.012	0.182	0.015	0.013

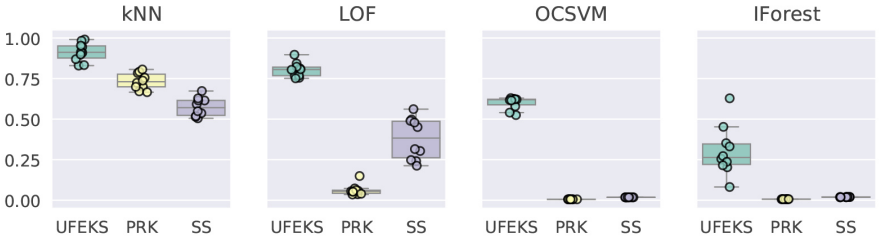
Table 4
Area under precision-recall curve (AUPRC) for real-world datasets

OD		κNN			LOF	
FR	UFEKS	PRK	SS	UFEKS	PRK	SS
ATSF8	0.914 $\pm$ 0.027	0.199 $\pm$ 0.028	0.040 $\pm$ 0.001	**0.960** $\pm$ 0.006	0.187 $\pm$ 0.030	0.067 $\pm$ 0.001
ATSF16	**0.868** $\pm$ 0.018	0.260 $\pm$ 0.038	0.039 $\pm$ 0.001	0.630 $\pm$ 0.012	0.066 $\pm$ 0.009	0.064 $\pm$ 0.001
ATSF32	**0.614** $\pm$ 0.027	0.186 $\pm$ 0.018	0.039 $\pm$ 0.001	0.176 $\pm$ 0.010	0.032 $\pm$ 0.002	0.063 $\pm$ 0.001
ATSF64	**0.306** $\pm$ 0.015	0.089 $\pm$ 0.005	0.038 $\pm$ 0.001	0.080 $\pm$ 0.003	0.028 $\pm$ 0.000	0.063 $\pm$ 0.000
WADI	0.092	0.047	0.128	0.057	0.064	0.055
SWaT	0.143	0.085	0.118	0.088	0.070	0.100
Average	0.489	0.144	0.067	0.332	0.074	0.069
OD		OCSVM			IForest	
FR	UFEKS	PRK	SS	UFEKS	PRK	SS
ATSF8	0.899 $\pm$ 0.006	0.026 $\pm$ 0.000	0.034 $\pm$ 0.000	0.461 $\pm$ 0.049	0.029 $\pm$ 0.000	0.035 $\pm$ 0.001
ATSF16	0.821 $\pm$ 0.026	0.026 $\pm$ 0.000	0.034 $\pm$ 0.000	0.247 $\pm$ 0.042	0.030 $\pm$ 0.000	0.034 $\pm$ 0.000
ATSF32	0.329 $\pm$ 0.026	0.027 $\pm$ 0.000	0.034 $\pm$ 0.000	0.132 $\pm$ 0.021	0.030 $\pm$ 0.000	0.034 $\pm$ 0.000
ATSF64	0.189 $\pm$ 0.003	0.032 $\pm$ 0.003	0.034 $\pm$ 0.000	0.113 $\pm$ 0.024	0.034 $\pm$ 0.000	0.034 $\pm$ 0.000
WADI	0.249	0.032	**0.751**	0.172	0.033	0.628
SWaT	0.783	0.118	**0.881**	0.643	0.054	0.365
Average	**0.545**	0.044	0.295	0.295	0.035	0.188

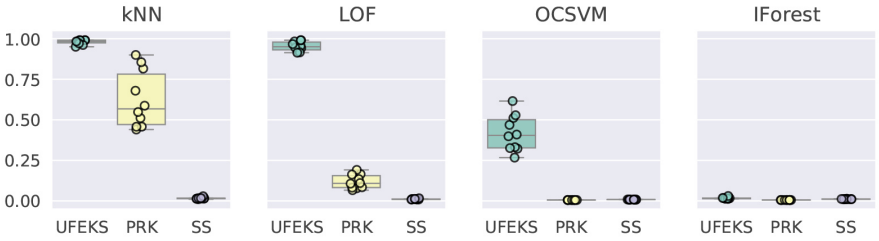
and each title in their plots shows the name of the corresponding outlier detection algorithm. Our algorithm, UFEKS, is superior to the other feature representation algorithms PageRank Kernel (PRK) and SubSequence (SS) in all synthetic datasets. In comparison with PRK and SS, our kernel matrix used in the algorithm does not sum up elements in a row of the kernel matrix that represents association between subsequences, while PRK and SS sum up them. Therefore, it is considered that UFEKS has high capability of representing features. Moreover, by using Gaussian kernel, it is expected that UFEKS can



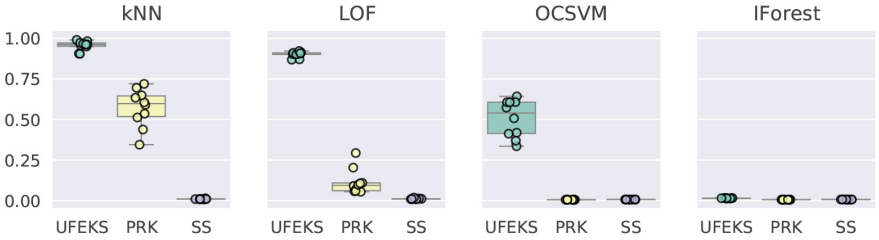
(a) SYN1 time series.



(b) SYN2 time series.



(c) SYN3 time series.

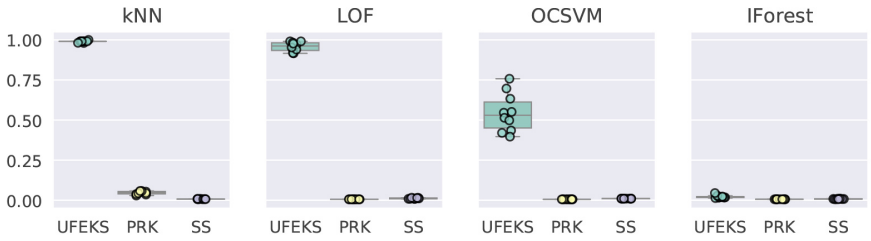


(d) SYN4 time series.

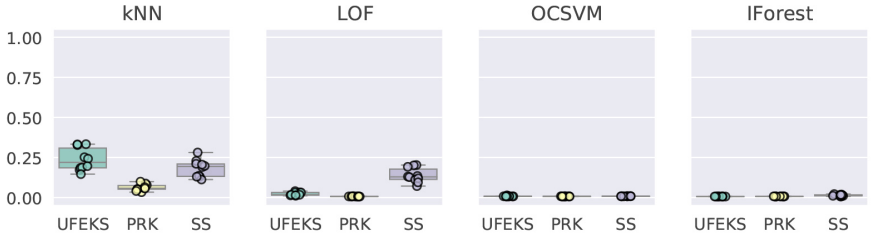
Fig. 3. AUPRC of SYN1, SYN2, SYN3, ans SYN4 time series.

reduce noise and extract features easier than SS. Furthermore, it is also interesting that κNN, which is an outlier detection algorithm, also tend to make better results than the other algorithms except for SYN8 datasets. Their details are shown in Table 3. The reasons why κNN have a good result is that outlier points tend to place far away from normal points.

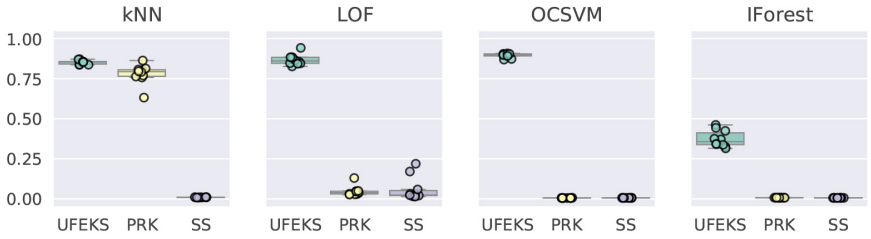
To analyze difference between feature representation methods deeper, we apply Principal Component Analysis (PCA) for the obtained feature representations from SYN7 datasets as a representative example. The result for Subsequence is plotted in Fig. 7 and those for UFEKS and PageRank kernel are in Fig. 8. The x- and y-axes in Fig. 7 indicate the first and the second principle components, respectively. In Fig. 8, we plot the first and second principle components in the left-hand side plots, the second and third ones for the middle plots, and the second and forth ones for the right-hand side plots. The circles and crosses



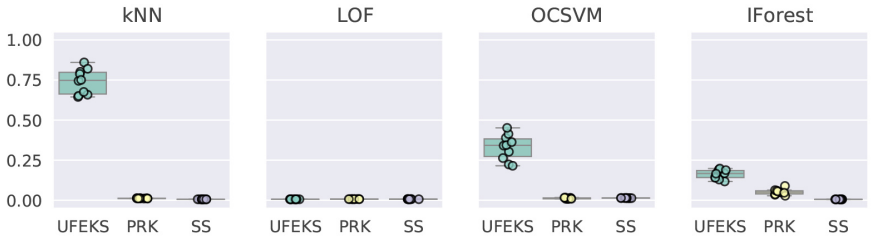
(a) SYN5 time series.



(b) SYN6 time series.



(c) SYN7 time series.



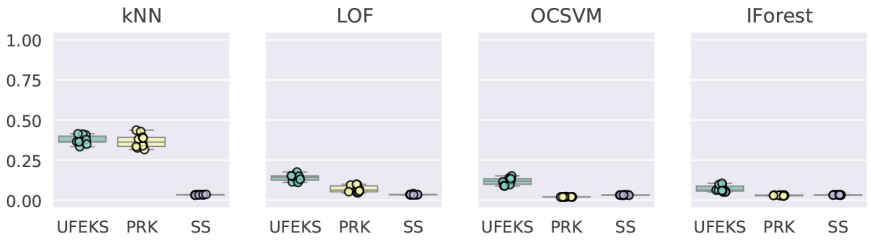
(d) SYN8 time series.

Fig. 4. AUPRC of SYN5, SYN6, SYN7, and SYN8 time series.

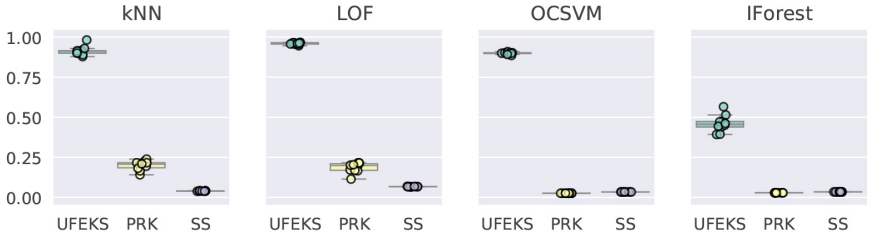
in all plots denote normal and outlier points, respectively. As the Fig. 7 shows, it seems to be difficult to detect outliers by a conventional algorithm like κNN from this feature representation because most of outliers are close to the normal data points. In contrast, we can see some outliers that are apart from normal data points in Fig. 8b and f. This demonstrates the effectiveness of our approach as it means that there is a possibility to detect such outliers by distance-based outlier detection methods from these feature representations. Note that a feature representation matrix from Subsequence (SS) has only two dimensions as we set the length of subsequence to be two.

4.4.2. ATSF datasets

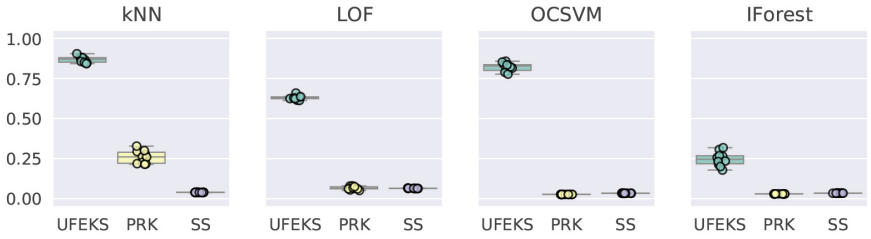
Figures from Figs 5b to 6a show results of ATSF datasets. A detail of the results is shown in Table. 4.



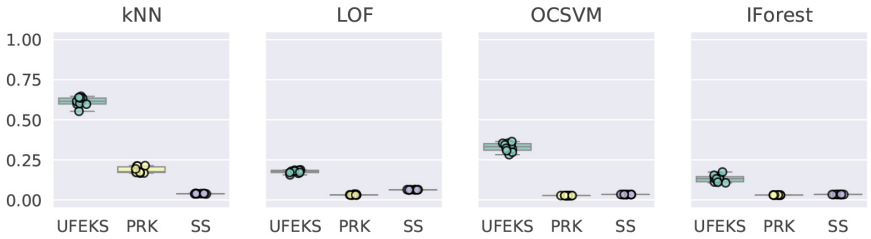
(a) SYN9 time series.



(b) ATSF8 time series.



(c) ATSF16 time series.



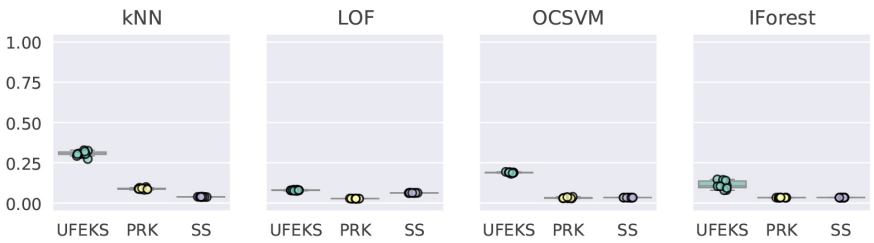
(d) ATSF32 time series.

Fig. 5. AUPRC of SYN9, ATSF8, ATSF16, and ATSF32 time series.

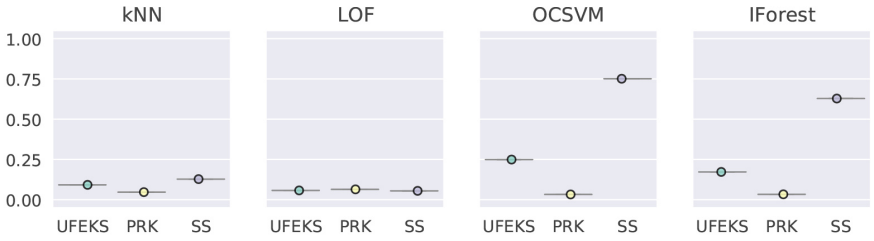
The results tend to be similar to synthetic datasets, that is, combinations of κNN and UFEKS have resulted in high accuracy for all datasets except for ATSF8.

4.4.3. WADI and SWaT datasets

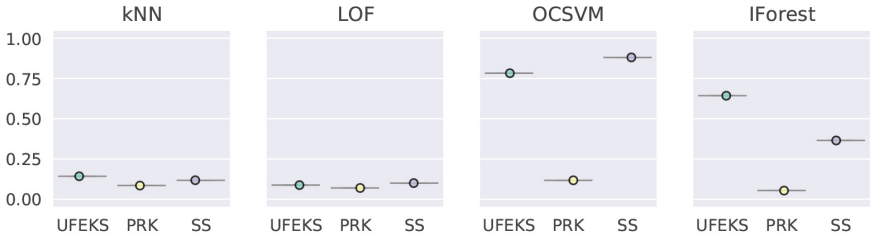
Figure 6b and c show results for WADI and SWaT datasets. These results are different from those for the other datasets because OCSVM and IForest have better results than κNN. Moreover, our algorithm, UFEKS, is not better than SS. To analyze this reason, we illustrate results of PCA for SWaT datasets in Fig. 9. As shown in Fig. 8, the x- and y-axes indicate the first and second principle component in the left plot, the second and the third principle components in the middle plot, and the second and forth principle components in the right plot. The circles and crosses in these plots denote normal and outlier



(a) ATSF64 time series.



(b) WADI time series.



(c) SWaT time series.

Fig. 6. AUPRC of ATSF64, WADI and SWaT time series.

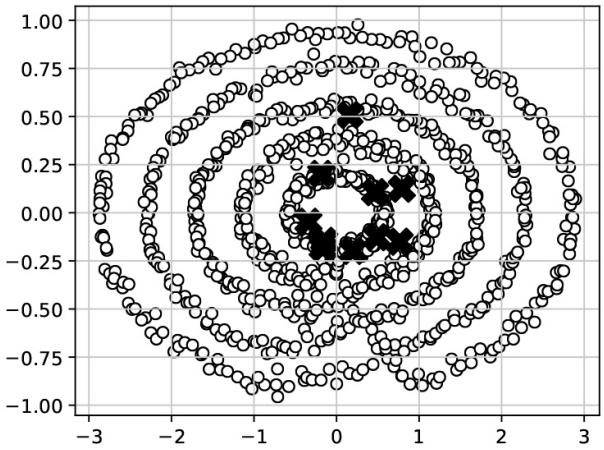
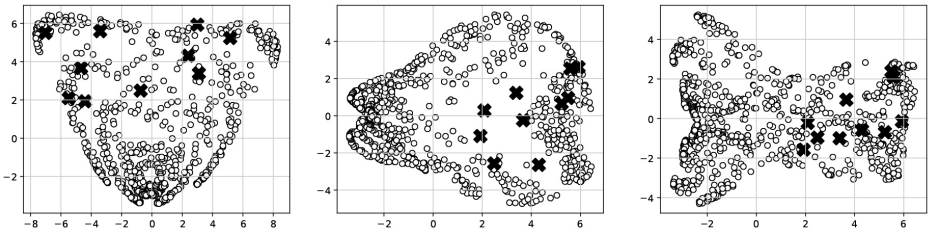
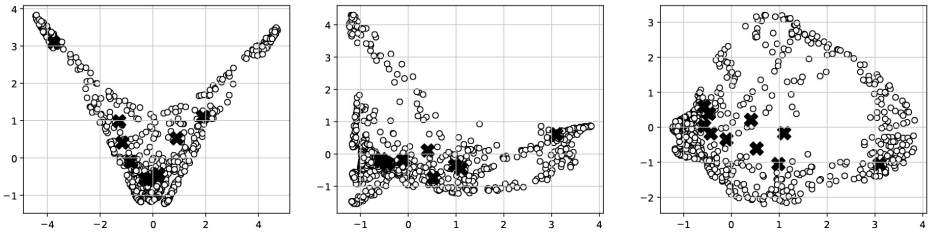


Fig. 7. A result of PCA that is applied to the feature representation obtained by Subsequence (SS) for the SYN7 time series dataset.

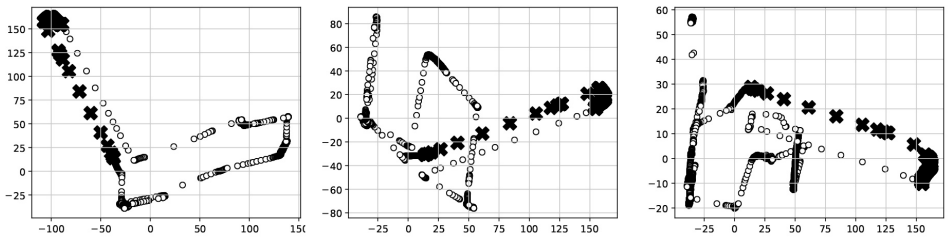


(a) The 1st and 2nd principal components for UFEKS. (b) The 2nd and 3rd principal components for UFEKS. (c) The 2nd and 4th principal components for UFEKS.



(d) The 1st and 2nd principal components for PRK. (e) The 2nd and 3rd principal components for PRK. (f) The 2nd and 4th principal components for PRK.

Fig. 8. Results of PCA for feature representations obtained by UFEKS (upper row) and PageRank kernel (PRK; lower row) for the SYN7 time series dataset.



(a) The 1st and 2nd principal components for UFEKS. (b) The 2nd and 3rd principal components for UFEKS. (c) The 2nd and 4th principal components for UFEKS.

Fig. 9. Results of PCA for the feature representation obtained by UFEKS from SWaT time series.

points, respectively. These three plots have different features compared with other PCA plots for synthetic datasets that we have shown before. Outliers and normal data points are aligned each other. Furthermore, distances between data points seem to be comparatively equal. In this case, detecting outliers using distance-based outlier detection algorithms such as κNN may be difficult. Although it might be possible to detect them by supervised learning, it is out of scope of this paper. We consider that both WADI and SWaT time series include a variety types of data patterns and it is fundamentally difficult to detect outliers from these datasets in an unsupervised manner.

5. Conclusion

In this paper, we have proposed a new algorithm, called *Unsupervised Feature Extraction using Kernel*

Method and Stacking (UFEKS), to extract features from multivariate time series without any labels. The UFEKS (1) divides a given time series into a set of its subsequences, (2) makes a kernel matrix from each subsequences using the RBF kernel, (3) horizontally concatenates kernel matrices into a single kernel matrix, and (4) extracts row vectors in the concatenated matrix as feature vectors. To evaluate our algorithm, we have applied it to the outlier detection task with four outlier detection methods for non-time series data, such as *κth-Nearest Neighbor* (κNN), *Local Outlier Factor* (LOF), *One-class Support Vector Machine* (OCSVM), and *Isolation Forest* (IForest), and examined the performance on nine types of synthetic and six types of real-world multivariate time series datasets. We have empirically shown that UFEKS is particularly effective in outlier detection; we can find combinatorial outliers in multivariate time series without labels more accurately from the extracted feature vectors. Furthermore, we have empirically analyzed behavior of UFEKS with its comparison to other feature extraction methods by visualizing the resulting kernel matrices by *Principle Component Analysis* (PCA).

Since our method offers a powerful feature extraction scheme that can be applied to any multivariate time series data, it is our interesting future work to apply UFEKS to not only outlier detection but other data mining tasks such as clustering for multivariate time series data.

Acknowledgments

This work was supported by JST, PRESTO Grant Number JPMJPR1855, Japan and JSPS KAKENHI Grant Number JP21H03503 (MS).

References

[1] C.C. Aggarwal, *Outlier Analysis*, Springer, 2017.
[2] C.C. Aggarwal and S. Sathe, *Outlier Ensembles*, Springer, 2017.
[3] R. Agrawal, C. Faloutsos and A. Swami, Efficient similarity search in sequence databases, in: *Lecture Notes in Computer Science*, Vol. 730, pages 69–84. 1993.
[4] S. Ahmad, A. Lavin, S. Purdy and Z. Agha, Unsupervised real-time anomaly detection for streaming data, *Neurocomputing* **262** (nov 2017), 134–147.
[5] S.D. Bay and M. Schwabacher, Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule, in: *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38, 2003.
[6] D. Berndt and J. Clifford, Using dynamic time warping to find patterns in time series, *Workshop on Knowledge Knowledge Discovery in Databases* **398** (1994), 359–370.
[7] K. Bhaduri, B.L. Matthews and C.R. Giannella, Algorithms for speeding up distance-based outlier detection, in: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, number August 2011, pages 859–867, 2011.
[8] V. Chandola, A. Banerjee and V. Kumar, Anomaly detection, *ACM Computing Surveys* **41**(3) (jul 2009), 1–58.
[9] H. Cheng, P.-N. Tan, C. Potter and S. Klooster, Detection and Characterization of Anomalies in Multivariate Time Series, in: *Proceedings of the 2009 SIAM International Conference on Data Mining*, Vol. 1, pages 413–424, apr 2009.
[10] C. Guo, H. Li and D. Pan, An improved piecewise aggregate approximation based on statistical features for time series mining, in: Y. Bi and M.-A. Williams, editors, *Knowledge Science, Engineering and Management*, pages 234–244, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
[11] M. Gupta, J. Gao, C.C. Aggarwal and J. Han, Outlier detection for temporal data: A survey, *IEEE Transactions on Knowledge and Data Engineering* **26**(9) (sep 2014).
[12] A.K. Jardine, D. Lin and D. Banjevic, A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical Systems and Signal Processing* **20**(7) (2006), 1483–1510.
[13] E. Keogh, S. Lonardi and C.A. Ratanamahatana, Towards parameter-free data mining, in: *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD '04*, ACM Press, 2004.
[14] E.J. Keogh and M.J. Pazzani, A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases, in: *4th Pacific-Asia Conference, PAKDD 2000*, pages 122–133, 2000.

[15] E.M. Knorr, R.T. Ng and V. Tucakov, Distance-based outliers: Algorithms and applications, *The VLDB Journal* **8**(3–4) (2000), 237–253.
[16] F. Korn, H.V. Jagadish and C. Faloutsos, Efficiently supporting ad hoc queries in large datasets of time sequences, *ACM SIGMOD Record* **26**(2) (jun 1997), 289–300.
[17] J. Lee, H.S. Choi, Y. Jeon, Y. Kwon, D. Lee and S. Yoon, Detecting System Anomalies in Multivariate Time Series with Information Transfer and Random Walk, in: *Proceedings of 5th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, pages 71–80, 2018.
[18] F.T. Liu, K.M. Ting and Z.-H. Zhou, Isolation Forest, in: *Proceedings of 2008 IEEE International Conference on Data Mining*, pages 413–422. IEEE, dec 2008.
[19] J. Ma and S. Perkins, Time-series Novelty Detection Using One-class Support Vector Machines, in: *Proceedings of the International Joint Conference on Neural Networks*, Vol. 3, pages 1741–1745, 2003.
[20] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal and G. Shroff, LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection, in: *ICML 2016 Anomaly Detection Workshop*, 2016.
[21] J. Mei, M. Liu, Y.-F. Wang and H. Gao, Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification, *IEEE Transactions on Cybernetics* **46**(6) (jun 2016), 1363–1374.
[22] Mingyan Teng, Anomaly detection on time series, in: *2010 IEEE International Conference on Progress in Informatics and Computing*, pages 603–608, dec 2010.
[23] F. Mörchen, Time series feature extraction for data mining using DWT and DFT, *Technical Report, No. 33, Department of Mathematics and Computer Science, University of Marburg, Germany*, pages 1–31, 2003.
[24] A. Munoz and J. Muruzabal, Self-Organizing Maps for Outlier Detection, 1995.
[25] H. Qiu, Y. Liu, N.A. Subrahmanya and W. Li, Granger Causality for Time-Series Anomaly Detection, in: *Proceedings of 12th International Conference on Data Mining*, pages 1074–1079, 2012.
[26] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria and E. Keogh, Searching and mining trillions of time series subsequences under dynamic time warping, in: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 262–270, 2012.
[27] N.N.R. Ranga Suri, N. Murty and G. Athithan, *Outlier Detection: Techniques and Applications*, Vol. 155 of *Intelligent Systems Reference Library*, Springer, 2019.
[28] L. Seymour, P.J. Brockwell and R.A. Davis, *Introduction to Time Series and Forecasting*, Springer, 2016.
[29] M. Sugiyama and K.M. Borgwardt, Rapid distance-based outlier detection via sampling, in: *Advances in Neural Information Processing Systems*, pages 1–9, 2013.
[30] N. Takeishi and T. Yairi, Anomaly detection from multivariate time-series with sparse representation, in: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pages 2651–2656, 2014.
[31] S.J. Taylor and B. Letham, Forecasting at Scale, *PeerJ*, 2017.
[32] H. Wang, M.J. Bah and M. Hammad, Progress in outlier detection techniques: A survey, *IEEE Access* **7** (2019), 107964–108000.
[33] B.K. Yi and C. Faloutsost, Fast time sequence indexing for arbitrary 4 norms, in: *Proceedings of the 26th International Conference on Very Large Data Bases, VLDB'00*, pages 385–394, 2000.
[34] S. Zhai, Y. Cheng, W. Lu and Z. Zhang, Deep Structured Energy Based Models for Anomaly Detection, in: *Proceedings of 33rd International Conference on Machine Learning*, Vol. 3, may 2016, pp. 1742–1751.
[35] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen and N.V. Chawla, A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33, jul 2019, pp. 1409–1416.
[36] C. Zhou and R.C. Paffenroth, Anomaly Detection with Robust Deep Autoencoders, in: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, aug 2017.
[37] B. Zong, Q. Song, M.R. Min, W. Cheng, C. Lumezanu, D. Cho and H. Chen, Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection, in: *Proceedings of 6th International Conference on Learning Representations*, pages 1–19, 2018.