



# A New Distributional Treatment for Time Series and An Anomaly Detection Investigation

Kai Ming Ting\*, Zongyou Liu\*, Hang Zhang  
National Key Laboratory for Novel Software Technology,  
Nanjing University  
China  
{tingkm,liuzy,zhanghang}@lamda.nju.edu.cn

Ye Zhu  
School of Information Technology,  
Deakin University  
Australia  
ye.zhu@ieee.org

## ABSTRACT

Time series is traditionally treated with two main approaches, i.e., the time domain approach and the frequency domain approach. These approaches must rely on a sliding window so that time-shift versions of a periodic subsequence can be measured to be similar. Coupled with the use of a root point-to-point measure, existing methods often have quadratic time complexity. We offer the third  $\mathbb{R}$  domain approach. It begins with an insight that subsequences in a periodic time series can be treated as sets of independent and identically distributed (iid) points generated from an unknown distribution in  $\mathbb{R}$ . This  $\mathbb{R}$  domain treatment enables two new possibilities: (a) the similarity between two subsequences can be computed using a distributional measure such as Wasserstein distance (WD), kernel mean embedding or Isolation Distributional kernel (IDK); and (b) these distributional measures become non-sliding-window-based. Together, they offer an alternative that has more effective similarity measurements and runs significantly faster than the point-to-point and sliding-window-based measures. Our empirical evaluation shows that IDK and WD are effective distributional measures for time series; and IDK-based detectors have better detection accuracy than existing sliding-window-based detectors, and they run faster with linear time complexity.

### PVLDB Reference Format:

Kai Ming Ting, Zongyou Liu, Hang Zhang and Ye Zhu. A New Distributional Treatment for Time Series and An Anomaly Detection Investigation. PVLDB, 15(11): 2321 - 2333, 2022.  
doi:10.14778/3551793.3551796

### PVLDB Artifact Availability:

The source code, data, and/or other artifacts are available at <https://github.com/IsolationKernel/Codes/tree/main/IDK/TS>.

## 1 INTRODUCTION

Time series has been studied for more than one century [15]. It is traditionally treated with two main approaches, i.e., the time domain and the frequency domain approaches (see e.g., [31].) The former views lagged relationships as the most important component

Table 1: Three types of measures for time series.

Domain/type	Example measures
Time	$\ell_p$ -norm, DTW, Lorentzian, Jaccard, Canberra
Frequency	Cross-correlation, Wasserstein-Fourier distance
$\mathbb{R}$	WD, KME, IDK, Bergman-Divergence

in modeling; and the latter views cycles as the most important component.

It is therefore of no surprise that, similarity/distance measures, a core operation in data mining, for time series are in time or frequency domain. Typical measures of time domain are lock-step and elastic measures (as examined in [22]), which include  $\ell_p$ -norm and Dynamic Time Warping (DTW) [26]. Cross-correlation measures (e.g., Shape-based distance (SBD) [21]) and a recent Wasserstein-Fourier distance (WFD) [3] are examples in frequency domain.

Both types of measures have a long-standing issue of high computational cost, despite recent progress. Euclidean distance is identified as the fastest among 71 measures in a recent study [22], which is the typical measure used.

We offer a third approach based on an insight that subsequences in a time series can be treated as sets of independent and identically distributed (iid) points generated from an unknown distribution in  $\mathbb{R}$ . With this  $\mathbb{R}$  domain treatment, the similarity between two subsequences can be computed using a distributional measure such as Wasserstein distance (WD) [25], kernel mean embedding (KME) [19] or Isolation Distributional kernel (IDK) [35].

Table 1 shows examples of the three approaches. While a distributional measure has been used previously in time series (e.g., [3, 11]), the time series is represented in the frequency domain before WD is applied, as in WFD [3]. Time series has never been measured using the  $\mathbb{R}$  domain approach, as far as we know.

A possible reason why a distributional measure has not been widely used is that: These distributional measures often require a computationally expensive process. Two examples are: (a) WD [25] requires a process to find an optimal transportation plan to move from one distribution to another; (b) KME needs to convert a Gaussian kernel (the root measure) to a finite-dimensional feature map; and it is a fundamental issue [19]. To compute the similarity between two subsequences, both WD and KME have a time cost of at least  $m^2$ , where  $m$  is the length of each subsequence.

A recent breakthrough in kernel mean embedding called Isolation Distributional Kernel (IDK) [35] enables each distribution to be mapped independently into a finite-dimensional point in Hilbert space; and the similarity between two distributions can then be computed as the similarity of two points in Hilbert space using a

\* Both are equal first authors.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing [info@vldb.org](mailto:info@vldb.org). Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.  
Proceedings of the VLDB Endowment, Vol. 15, No. 11 ISSN 2150-8097.  
doi:10.14778/3551793.3551796

dot product. The total cost is  $m$  because the feature mapping costs  $m$  for each subsequence; and each similarity computation takes constant time.

The proposed distributional similarity measurement in time series is unique in two aspects. First, no existing measures in time series are based on distributional treatments in the  $\mathbb{R}$  domain, as far as we know. Second, one measure, i.e., IDK, computes the similarity between two subsequences via a dot product in Hilbert space, instead of typically using a point-to-point distance function in input space or in frequency domain.

Our contributions are:

- (1) Providing an insight that each subsequence in a periodic time series can be effectively represented as a set of iid points in the  $\mathbb{R}$  domain, generated from an unknown distribution. This enables a distributional measure to be employed to measure the similarity between two subsequences; and a simple non-sliding-window method can be used to extract subsequences from a time series.
- (2) With the above insight, examining subsequence similarity measurements using three distributional measures, i.e., Wasserstein distance (WD), kernel mean embedding (KME) and Isolation Distributional kernel (IDK). We show that IDK is the most effective and efficient measure.
- (3) Defining an anomalous subsequence as a rare sample, generated from a distribution which is different from the one that generates the majority of the samples in the dataset (i.e., the periodic time series.)
- (4) Endowing existing anomaly detectors with WD, KME and IDK, we show that IDK-based detectors are the most effective for anomalous subsequence detection and run faster with linear time complexity.

The advantages of the proposed distributional treatment over the traditional treatments are two-fold. First, in terms of similarity measures, the proposed distributional treatment can better represent a subsequence in a periodic time series than the traditional representation because the former can better differentiate (i) a distribution from its variations (e.g., shortened or lengthened versions) or its noise-corrupted versions; (ii) dissimilar distributions (e.g., a different subsequence from those that are the norm in a periodic time series). Yet, it treats all time-shift versions of a periodic subsequence as samples generated from the same distribution, without additional effort (see the next paragraph).

Second, existing measures in time or frequency domain rely on a sliding window to generate subsequences in a time series. This is required in order to identify time-shift subsequences to be similar. But this has come at a high cost because it has to deal with a total of  $n - m + 1$  subsequences of length  $m$  in a time series of length  $n$ . In contrast, the proposed distributional treatment only needs to examine  $\lfloor n/m \rfloor$  subsequences.

In a nutshell, the proposed distributional treatment in the  $\mathbb{R}$  domain is more powerful and robust than a measure in either the time or frequency domain; and it runs faster because it needs to deal with much fewer subsequences in computing their similarities in a time series.

When the proposed distributional treatment is applied for anomalous subsequence detection, the IDK-based detectors have the unique

characteristic that only two levels of IDK are applied to perform the anomaly detection, without learning or additional processes. This has led to the following advantages:

- IDK-based detectors have linear runtime. In contrast, the fastest state-of-the-art sliding-window-based detector has superlinear runtime, and the other distributional detectors have quadratic runtime.
- IDK-based detectors have overall better detection accuracy than other contenders.

The rest of the paper is organized as follows. Section 2 describes the related work in time series. Section 3 provides the insight and intuitive examples of the proposed distributional treatment. Section 4 presents the assumption and the properties of distributional measures in  $\mathbb{R}$ . Section 5 introduces the detectors, endowed with the three distributional measures (i.e., WD, KME and IDK), to detect anomalous subsequences. Section 6 gives the experimental results. Discussion and conclusions are provided in the last two sections.

## 2 RELATED WORK

We begin our review from a recent study of measures in time series, and re-categorize them along the line between time and frequency domains. As such, the previous categories in [22], i.e., lock-step and elastic measures are grouped into the time domain measures; and the cross-correlation measures rely on the Fast Fourier Transform (FFT) to operate in the frequency domain. There are a total of 63 measures in the time domain; and 4 in the frequency domain [22].

### 2.1 Issues in Time/Frequency Domain Measures

The use of a point-to-point distance in time domain necessitates a point-to-point alignment between two subsequences during the distance measurement. This makes the measurement sensitive to misalignment between subsequences and between time-shift subsequences that occur in a time series. Significant efforts have been invested to address the need for a point-to-point alignment and the resultant high time complexity issue.

Remedies to the misalignment issue include DTW that allows full flexibility in alignment but high computational cost; and constrained DTW (cDTW) [27] allows limited flexibility in alignment in exchange for reduced computational cost. Current research has focused on ways to reduce its time complexity via bounding to reduce its search space (e.g., [14, 30, 33]). The latest version is said to have reduced to subquadratic time complexity, i.e.,  $O(n^2 \log \log n / \log \log n)$  [8].

Matrix Profile (MP) is a data structure that computes the  $z$ -normalized Euclidean distances between all subsequences that can be extracted from a time series and their nearest neighbors in the time series based on a point-to-point distance [7, 41]. A fast implementation is called STOMP [42].

STOMP first computes a series (profile) of  $n - m + 1$  distances between a query of length  $m$  and a time series of length  $n$  via a sliding window. A matrix in STOMP records  $n - m + 1$  distances between  $n - m + 1$  subsequences and their nearest neighbors in the time series [7, 42].

Shape-based distance (SBD) [21] is a cross-correlation measure that utilizes an inner product as its root measure. By examining all possible shifts between two sequences, it is robust to time shift.

**Table 2: Key symbols and notations as used in [36]**

$x$ or $y$	A point in 1-dimensional real domain $\mathbb{R}$
$\ x - y\ _p$	$\ell_p$ -norm of $x - y$ .
$Y_{i,m}$	Periodic subsequence $[y_1, \dots, y_m]_i$ of $m$ points.
$\mathcal{P}_Y$	Probability distribution that generates the set $Y \subset \mathbb{R}$
$\widehat{\mathcal{K}}_I$	Isolation Distributional Kernel (IDK)
$\widehat{\Phi}$	Feature map or kernel mean map of IDK
$\mathbf{g}$	$Y$ is mapped to a point $\mathbf{g} = \widehat{\Phi}(\mathcal{P}_Y)$ in Hilbert space $\mathcal{H}$
$\Pi$	A set of points $\{\mathbf{g}_j \mid j = 1, \dots, n\}$ in $\mathcal{H}$ , $\mathbf{g} \sim \mathbf{P}_\Pi$
$\mathbf{P}_\Pi$	A distribution that generates a set $\Pi$ of points in $\mathcal{H}$
$\text{IDK}^2$	Two levels of IDK associated with $\widehat{\Phi}$ & $\widehat{\Phi}_2$

The convolution of two subsequences is computed as the Inverse Discrete Fourier Transform (IDFT) of the product of the Discrete Fourier Transforms (DFT) of the individual subsequences; and the latter must be computed using a Fast Fourier Transform to get a speedup. Nevertheless, the overall time cost is still high.

In a nutshell, the use of a point-to-point distance in either time or frequency domain is the root cause of the potential misalignment issues and high computational cost. The latter has been a long-standing issue, despite some progress recently (e.g., DTW reduces the time complexity from quadratic to subquadratic [8]).

Here we show that, by abandoning the insistence to use a point-to-point distance in either time or frequency domain, a distributional treatment on  $\mathbb{R}$  domain has none of the misalignment issues; and it is possible to significantly reduce high computational cost.

## 2.2 Distributional Measures

Table 2 shows the key symbols used in this paper.

Rather than representing a time series in either time or frequency domain, this paper examines a distributional approach which requires subsequences in a time series to be represented as iid samples generated from a probability density function (pdf) in  $\mathbb{R}$  domain. Then, a distributional measure such as kernel mean embedding (KME) or Wasserstein distance (WD) can be used to measure similarity between subsequences, as samples of one or more pdfs. We review KME and WD in this section.

Let  $X$  and  $Y$  be two nonempty datasets where each point  $x$  in  $X$  and  $Y$  belongs to  $\mathcal{X} \subseteq \mathbb{R}$  and is drawn from probability distributions  $\mathcal{P}_X$  and  $\mathcal{P}_Y$  defined on  $\mathbb{R}$ , respectively.

Using kernel mean embedding (KME) [19, 32], the estimation of the distributional kernel  $\widehat{\mathcal{K}}$  on  $\mathcal{P}_X$  and  $\mathcal{P}_Y$ , which is based on a point kernel  $\kappa$  on points  $x, y \in \mathcal{X}$ , is given as:

$$\widehat{\mathcal{K}}_G(\mathcal{P}_X, \mathcal{P}_Y) = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} \kappa(x, y). \quad (1)$$

where  $\kappa(x, y) = \exp(-\frac{\|x-y\|_2^2}{2\sigma^2})$ ,  $\sigma > 0$  is Gaussian kernel.

The Wasserstein distance [25] is defined as:

$$W_p(\mathcal{P}_X, \mathcal{P}_Y) = \left( \min_{\lambda \in \Lambda(\mathcal{P}_X, \mathcal{P}_Y)} \int_{\mathbb{R} \times \mathbb{R}} \|x - y\|_2^p d\lambda(x, y) \right)^{1/p} \quad (2)$$

where  $\Lambda(\mathcal{P}_X, \mathcal{P}_Y)$  denotes a collection of all measures on  $\mathbb{R} \times \mathbb{R}$  with marginals  $\mathcal{P}_X$  and  $\mathcal{P}_Y$ .

WD must optimize a transportation plan from  $\Lambda(\mathcal{P}_X, \mathcal{P}_Y)$ ; and KME needs a point kernel  $\kappa$ . Both need a distance measure and cost  $m^2$ , if each of  $X$  and  $Y$  has  $m$  points.

In the next section, we review a distributional measure which employs a data dependent kernel called Isolation Kernel [37].

## 2.3 Isolation Kernel (IK) and IDK

**2.3.1 Isolation Kernel.** IK is a data dependent kernel that derives directly from data without learning, and it has no closed-form expression [37]. It has been shown to improve the task-specific performance of SVM [37] and density-based clustering [24] by simply replacing the data independent kernel/distance in the algorithms. In the context of online kernel learning, IK has been shown to be the most significant factor that enables efficient and effective large-scale online kernel learning, where it eliminates the limiting elements of using a data independent kernel [34].

Let  $D \subset \mathcal{X} \subseteq \mathbb{R}$  be a dataset sampled from an unknown  $\mathcal{P}_D$ ; and  $\mathbb{H}_\psi(D)$  denote the set of all partitionings  $H$  that are admissible from  $\mathcal{D} \subset D$ , where each point  $z \in \mathcal{D}$  has the equal probability of being selected from  $D$ ; and  $|\mathcal{D}| = \psi$ . Each  $\theta[z] \in H$  isolates a point  $z \in \mathcal{D}$ . Let  $\mathbb{1}(\cdot)$  be an indicator function.

**DEFINITION 1.** [24, 37] For any two points  $x, y \in \mathbb{R}$ , Isolation Kernel of  $x$  and  $y$  is defined to be the expectation taken over the probability distribution on all partitionings  $H \in \mathbb{H}_\psi(D)$  that both  $x$  and  $y$  fall into the same isolating partition  $\theta[z] \in H$ , where  $z \in \mathcal{D} \subset D$ ,  $\psi = |\mathcal{D}|$ :

$$\begin{aligned} \kappa_I(x, y | D) &= \mathbb{E}_{\mathbb{H}_\psi(D)} [\mathbb{1}(x, y \in \theta | \theta \in H)] \\ &= \frac{1}{t} \sum_{i=1}^t \mathbb{1}(x, y \in \theta | \theta \in H_i) \\ &= \frac{1}{t} \sum_{i=1}^t \sum_{\theta \in H_i} \mathbb{1}(x \in \theta) \mathbb{1}(y \in \theta) \end{aligned} \quad (3)$$

where  $\kappa_I$  is constructed using a finite number of partitionings  $H_i$ ,  $i = 1, \dots, t$ , where each  $H_i$  is created using randomly subsampled  $\mathcal{D}_i \subset D$ ; and  $\theta$  is a shorthand for  $\theta[z]$ .

**DEFINITION 2.** [35] **Feature map of Isolation Kernel.** For point  $x \in \mathbb{R}$ , the feature mapping  $\Phi : x \rightarrow \{0, 1\}^{t \times \psi}$  of  $\kappa_I$  is a vector that represents the partitions in all the partitioning  $H_i \in \mathbb{H}_\psi(D)$ ,  $i = 1, \dots, t$ ; where  $x$  falls into either only one of the  $\psi$  hyperspheres or none in each partitioning  $H_i$ .

Re-express Eq 3 using  $\Phi$  gives:

$$\kappa_I(x, y | D) = \frac{1}{t} \langle \Phi(x|D), \Phi(y|D) \rangle \quad (4)$$

**2.3.2 Isolation Distributional Kernel.** Based on the same framework of KME [19], IK has been used as the basis to produce a distributional kernel which measures the similarity between two distributions called Isolation Distributional Kernel (IDK) [35]. IDK has since been applied to point anomaly detection as well as group anomaly detection [36].

Given the feature map  $\Phi$  (defined in Definition 2) and Eq 4, the estimation of KME can be expressed based on the feature map of Isolation Kernel  $\kappa_I(x, y)$ .

DEFINITION 3. [35] Isolation Distributional Kernel of two distributions  $\mathcal{P}_X$  and  $\mathcal{P}_Y$  is given as:

$$\begin{aligned}\widehat{\mathcal{K}}_t(\mathcal{P}_X, \mathcal{P}_Y | D) &= \frac{1}{t|X||Y|} \sum_{x \in X} \sum_{y \in Y} \langle \Phi(x|D), \Phi(y|D) \rangle \\ &= \frac{1}{t} \left\langle \widehat{\Phi}(\mathcal{P}_X|D), \widehat{\Phi}(\mathcal{P}_Y|D) \right\rangle\end{aligned}\quad (5)$$

where  $\widehat{\Phi}(\mathcal{P}_X|D) = \frac{1}{|X|} \sum_{x \in X} \Phi(x|D)$  is kernel mean map.

IDK [35] and its resultant group anomaly detector called IDK<sup>2</sup> [36] provide the foundation and the tool required in dealing with anomalous subsequence detection in periodic time series.

We argue that each subsequence in a periodic time series can be better represented as a set of iid points generated from a distribution in  $\mathbb{R}$  than in the time or frequency domain.

On the surface, this appears to be counter-intuitive. We provide the insight and intuitive examples in the next section.

### 3 THE INSIGHT AND INTUITIVE EXAMPLES

A strictly stationary time series is defined as follows [31]:

DEFINITION 4. A time series is strictly stationary if a collection of values  $\{x_{1+h}, \dots, x_{m+h}\} \subset \mathbb{R}$  has the probabilistic behavior which is identical to any  $h$ -time-shift collection. That is  $\mathcal{P}(x_1 \leq c_1, \dots, x_m \leq c_m) = \mathcal{P}(x_{1+h} \leq c_1, \dots, x_{m+h} \leq c_m)$  for all points in the time series and all numbers  $c_1, \dots, c_m$ ; and all time shifts  $h = 0, \pm 1, \pm 2, \dots$

In simple terms, the stationary property denotes ‘regularity’ [31] in a time series that is characterized by recurring subsequences of length  $m$  which have the same joint probability distribution. Note that the subsequences could recur cyclically as in periodic time series, or noncyclically as in aperiodic time series. They are normal subsequences in the context of anomalous subsequence detection.

Let the recurring subsequence be  $X_h = \{x_{1+h}, \dots, x_{m+h}\}$ ; and assume that  $X_h$  is a set of iid points sampled from pdf  $\mathcal{P}_{X_h}$ .

This gives rise to the insight that each subsequence of  $m$  points in a stationary time series has

$$\mathcal{P}_{X_0} = \mathcal{P}_{X_h}, \quad \forall h = \pm 1, \pm 2, \dots$$

Note that the iid assumption is required only to apply an existing distribution measure to a time series. It does not alter the time dependency of adjacent points that has already existed.

In the context of anomalous subsequence detection, the iid assumption is not an issue in a stationary periodic time series if its normal subsequences are assumed to be generated from  $\mathcal{P}_{X_h}$ , and anomalous subsequences are expected to emerge from a variation of  $\mathcal{P}_{X_h}$ , such that the time dependency between adjacent points is sustained albeit altered; but not from an arbitrary distribution where there is no time dependency between adjacent points. Examples of these anomalous subsequences are:

- (i) Subsequences with added noise:  $\mathcal{P}_{X_h + N(0, \sigma^2)}$ ;
- (ii) Shortened/lengthened subsequences:  $\mathcal{P}_{X'_h}$ , where  $X'_h = \{x_{1+h}, \dots, x_{m'+h}\}$  &  $m' \neq m$ ; and
- (iii) Subsequence  $\{x_{1+h}, \dots, x'_1, \dots, x'_w, \dots, x_{m+h}\}$ , where  $X' = \{x'_1, \dots, x'_w\}$  is generated from  $\mathcal{P}_{X'} \neq \mathcal{P}_{X_h}$ .

Figure 1 shows an example time series of sine wave and its displacements of 45 degrees ( $h = 125$ ) and 180 degrees ( $h = 500$ ). Each of the three time series has exactly the same pdf, as shown

Figure 1(d). Figure 1(e) shows the same time series as Figure 1(a), except with added Gaussian noise. Its pdf, shown in Figure 1(f), is different from that which generates all time-shift versions of a noiseless sine wave, shown in Figure 1(d).

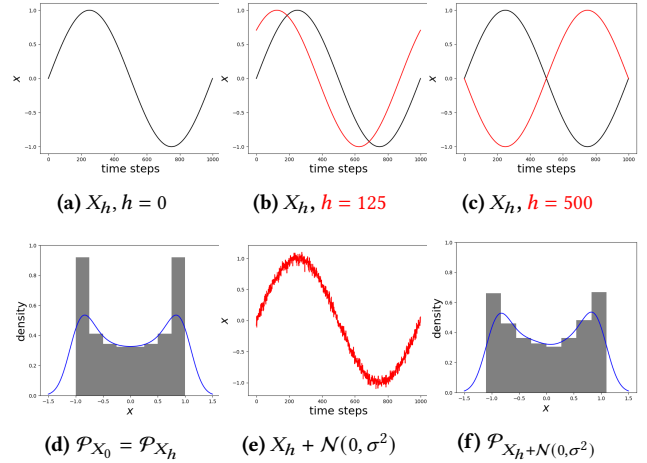
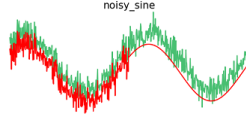
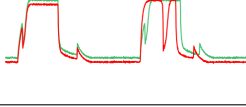
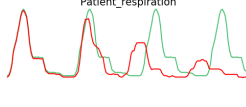


Figure 1: Example sine waves (with  $m = 1000$ ) and their pdfs

Table 3: Example time series and example similarity/distance measurements from IDK, WD, WFD, STOMP and NormA. The green curve is raised slightly so that it is not completely overlapped with the red curve, which contains at least one anomalous subsequence. Example similarity measurements between (1) two normal subsequences (on the row denoted as n-n); and (2) one normal subsequence and one anomalous subsequence (n-a) are also given.

Dataset	IDK	WD	WFD	STOMP	NormA	
noisy_sine 	✓	✓	✓	×	×	
	n-n	0.8	3	7	12	4
	n-a	0.6	18	14	7.5	2.9
TEK 	✓	✓	✓	✓	×	
	n-n	0.9	3	2	11	4.7
	n-a	0.7	8	5	13	4.6
Patient_respiration 	✓	✓	✓	×	✓	
	n-n	0.8	11	14	21	4.7
	n-a	0.5	17	19	20	4.9

Further examples are provided in Table 3.

The first row in Table 3 shows two periods of noisy sine (normal) subsequences in green; and one anomalous subsequence without

noise in red. IDK, WD and WFD measure the anomalous subsequence to be less similar to normal subsequence than that between two normal subsequences. Yet, the reverse is true for both STOMP and NormA which employ Euclidean distance. This is because the noiseless (anomalous) subsequence is indeed having shorter distance to a noisy subsequence than between two noisy (normal) subsequences, based on the lock-step Euclidean distance.

The second row shows that an anomalous subsequence which has subtle but clear differences from a normal one. This is an example of case (iii)  $\mathcal{P}_{X'} \neq \mathcal{P}_{X_h}$ , where  $X'$  is a narrow dip in the middle which does not occur in a normal subsequence. Here, all except NormA can identify the anomalous subsequence.

The third row shows that an anomalous subsequence which has clear and small differences from a normal one, but there is another anomalous subsequence with a peak signal several times larger that exists in other part of time series (not shown). This is an example in which a huge-difference anomalous subsequence can have a negative impact on a measure's ability to detect small-difference anomalous subsequences. Here, all except STOMP can identify the anomalous subsequence.

Note that the illustrations in Figure 1 using pdf are for conceptual explanation only. Though a measure such as WD needs a pdf estimation, not all distributional measures require it. For example, KME and IDK have no such requirement.

It is interesting to note that a sliding window produces  $X_h$ , where  $h = 0, 1, 2, 3, (n - m)$ , with a total of  $n - m + 1$  subsequences significantly overlapping from one subsequence to the next. With the insight and the distribution assumption, we show in the next section that only  $\lfloor n/m \rfloor$  non-overlapping subsequences  $X_h$  and  $X'_h$ , where  $X_h \cap X'_h = \emptyset$ , need to be examined since a part subsequence  $Z_h \subset X_h$  is generated from the same  $\mathcal{P}_{X_h}$ .

Though the stationary property stated in Definition 4 applies to both periodic and aperiodic time series, we focus on periodic time series, like many existing works (e.g., [1, 12, 13]). A discussion on aperiodic time series is provided in Section 7.

#### 4 ASSUMPTION AND PROPERTIES OF DISTRIBUTIONAL MEASURES

**DEFINITION 5.** *A periodic time series  $Y$  is a series of  $n$  points  $y \in \mathbb{R}$  sampled at a fixed time interval, and it has a set of non-overlapping periodic subsequences:  $Y_{i,m}, i = 1, \dots, s$ , where  $s = \lfloor n/m \rfloor$ ; and each subsequence  $Y_{i,m} = [y_1, \dots, y_m]_i$  of a period consists of consecutive  $m$  points in the series.*

For brevity, we use  $Y_i$  to denote  $Y_{i,m}$  hereafter, as  $m$  is the same for all subsequences.

**The assumption of the proposed  $\mathbb{R}$  domain approach: each subsequence  $Y_i$  in a time series is assumed to be iid points in  $\mathbb{R}$  which are generated from an unknown pdf  $\mathcal{P}_{Y_i}$ .**

With the above assumption, we can then employ a distributional measure to compute the similarity between two subsequences  $Y_i$  and  $Y_j$  to determine whether the two subsequences are generated from the same distribution (i.e.,  $\mathcal{P}_{Y_i} = \mathcal{P}_{Y_j}$ ) or different distributions (i.e.,  $\mathcal{P}_{Y_i} \neq \mathcal{P}_{Y_j}$ ).

**DEFINITION 6.** *A distributional similarity measure  $f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_j})$  for two subsequences  $Y_i$  and  $Y_j$  has the following properties:*

- I.  $0 \leq f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_j}) \leq 1$
- II.  $f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_j}) = f(\mathcal{P}_{Y_j}, \mathcal{P}_{Y_i})$
- III.  $f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_j}) = 1 \Leftrightarrow \mathcal{P}_{Y_i} = \mathcal{P}_{Y_j}$

#### Properties in relation to time series

- **Time shift:** If  $Y_i$  is a time-shift subsequence of  $Y_j$ , then  $\mathcal{P}_{Y_i} = \mathcal{P}_{Y_j}$ , i.e.,  $f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_j}) = 1$ .
- **Robust to noise of the same level & sensitive to different noise levels:** Let  $Y_i$  and  $Y_j$  be two normal subsequences with the same noise level  $\beta$ ; and another subsequence  $Y_a$  with a noise level  $\beta_a \neq \beta$ . Then,  $\mathcal{P}_{Y_i} \approx \mathcal{P}_{Y_j} \Rightarrow f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_j}) \approx 1$ .  
 $\mathcal{P}_{Y_i} \neq \mathcal{P}_{Y_a} \Rightarrow f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_a}) \ll 1$ .
- **Shortened/lengthened subsequences:** Let  $Y_i$  be a normal subsequence; and  $Y_s$  be a shortened (or lengthened) version of  $Y_i$ . Then,  $\mathcal{P}_{Y_i} \neq \mathcal{P}_{Y_s} \Rightarrow f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_s}) \ll 1$ .
- **Sensitive to a different subsequence:** Let  $Y_a$  be a subsequence which is generated from a distribution different from that which generates the normal subsequence  $Y_i$ . Then,  $\mathcal{P}_{Y_i} \neq \mathcal{P}_{Y_a} \Rightarrow f(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_a}) \ll 1$ .

### 5 DISTRIBUTION-BASED DETECTORS

**DEFINITION 7.** *An anomalous subsequence  $Y_a$  in a periodic time series is a subsequence which is rare and is generated from a pdf which is different from that generating the majority of the subsequences  $Y_i$  in the series, i.e.,  $\forall_i \mathcal{P}_{Y_i} \neq \mathcal{P}_{Y_a}$ .*

The above definition requires a distributional measure to compute the similarity between two subsequences.

We show here that one may use three existing distributional measures, i.e., IDK, KME and WD, to build distribution-based anomalous subsequence detectors. The methods based on IDK are described in Section 5.1, and those based on KME and WD are in Section 5.2.

#### 5.1 Anomalous Subsequence Detection via IDK

IDK [35] has all the properties mentioned in the last section:

$$\widehat{K}(\mathcal{P}_{Y_i}, \mathcal{P}_{Y_j} | Y) = \frac{1}{t} \left\langle \widehat{\Phi}(\mathcal{P}_{Y_i} | Y), \widehat{\Phi}(\mathcal{P}_{Y_j} | Y) \right\rangle,$$

where the kernel mean map converts a subsequence  $Y_i$  to a point in level-1 Hilbert space, i.e.,  $g_i = \widehat{\Phi}(\mathcal{P}_{Y_i} | Y)$ .

The set of points  $\Pi = \{g_i, i = 1, \dots, s\}$ , representing subsequences  $Y_i, i = 1, \dots, s$ , can then be used to construct the level-2 IDK  $\widehat{K}_2$  and its kernel mean map  $\widehat{\Phi}_2$  [36].

The similarity between  $g_i$  wrt  $\Pi$  is computed as

$$\widehat{K}_2(\delta(g_i), \mathcal{P}_{\Pi} | \Pi) = \frac{1}{t} \left\langle \widehat{\Phi}_2(\delta(g_i) | \Pi), \widehat{\Phi}_2(\mathcal{P}_{\Pi} | \Pi) \right\rangle$$

where  $\delta(g)$  is a Dirac measure of a point  $g$ .

Similarity  $\widehat{K}_2(\delta(g_i), \mathcal{P}_{\Pi})$  has values ranging within  $[0, 1]$ . Point anomalies are those  $g_i$  having the lowest similarities.

Point anomalies in level-2 Hilbert space correspond to the anomalous subsequences in the given periodic time series, i.e., they are detected using level-2 IDK. The entire process is called IDK<sup>2</sup>, and it is shown in Algorithm 1. It is identical to that used for group anomaly detection [36], except line 1 is added to deal with subsequences in a time series.

---

**Algorithm 1** IDK<sup>2</sup> for anomalous subsequence detection

---

**Input:** time series  $Y$ ; period length  $m$ ; sample sizes  $\psi$  of  $Y$  &  $\psi_2$  of  $\Pi$  for two levels of IK mappings  $\Phi$  &  $\Phi_2$ , respectively.

**Output:** A ranked list of periodic subsequences  $Y_{i,m}$ ,  $i = 1, \dots, s$  in ascending order by similarity score  $\alpha_i$ .

- 1:  $Y \rightarrow \{Y_{i,m}, i = 1, \dots, s\}$ , where  $s = \lfloor \text{length}(Y)/m \rfloor$
  - 2: \* Map each periodic subsequence in  $\mathbb{R}$  to a point in level-1 Hilbert space  $\widehat{\Phi}$  \*
  - For each  $i = 1, \dots, s$ ,  $g_i = \widehat{\Phi}(\mathcal{P}_{Y_{i,m}}|Y)$
  - 3:  $\Pi = \{g_i \mid i = 1, \dots, s\}$
  - 4: \* Compute the similarity of a  $\Phi_2$  mapped point wrt the  $\widehat{\Phi}_2$  mapped point of  $\Pi$  in level-2 Hilbert space \*
  - For each  $i = 1, \dots, s$ ,  $\alpha_i = \frac{1}{t} \langle \widehat{\Phi}_2(\delta(g_i)|\Pi), \widehat{\Phi}_2(\mathcal{P}_\Pi|\Pi) \rangle$
  - 5: Sort  $Y_{i,m} \subset Y$  in ascending order by  $\alpha_i$
- 

**Table 4: Variants of IDK-based detectors, obtained by simply replacing  $\alpha_i$  in line 4 in Algorithm 1.**

---

IDK*IK : $\alpha_i = \ \Phi_2(g_i \Pi)\ _2$
Anomalies are close to the origin in level-2 Hilbert space.
k-IDK : $\alpha_i = \text{kth-max}\{\langle g_i, g \rangle \mid g \in \Pi \setminus \{g_i\}\}$
Anomalies have low IDK similarity wrt the kth most similar neighbor in level-1 Hilbert space.

---

Note that  $\widehat{\Phi}_2(\mathcal{P}_\Pi)$  can be viewed as a typical periodic subsequence summarized from all periodic subsequences in a time series. Therefore, anomalous and normal periodic subsequences, as detected by IDK<sup>2</sup>, are defined as:

**DEFINITION 8.** *An anomalous (normal) periodic subsequence has low (high) similarity with respect to the typical periodic subsequence derived from a periodic time series.*

Two variants of IDK-based detectors are provided in Table 4. k-IDK is equivalent to kNN [16], except that the dot product is used for similarity measurement, instead of distance calculation. We create these variants in order to perform an ablation study.

Our proposed use of the distributional treatment is unique in time series; and the application of IDK in IDK<sup>2</sup> and IDK\*IK in time series is unique because it is not a point-to-point measure based method in  $\mathbb{R}$ ; but an operation in Hilbert space.

## 5.2 Anomalous Subsequence Detection Using KME and WD

KME has previously been incorporated in OCSVM [28] to produce OCSMM [20] by replacing the Gaussian kernel with KME, as defined in Eq 1. OCSMM has been used to detect group anomalies [20], where each group of points is assumed to be generated from an unknown distribution.

By assuming that each periodic subsequence (as defined in Definition 5) is generated from an unknown distribution, we can then use OCSMM to detect anomalous subsequences.

Once the Wasserstein distance (WD) is computed between two subsequences, it can be used in two ways. First, converting WD to a kernel such as Laplacian kernel. This was previously used in SVM

to perform graph classification [38]. Here, we use the WD-induced Laplacian kernel in OCSVM to perform anomalous subsequence detection. Second, WD is used in a distance-based anomaly detector such as LOF [2] and kNN [16], by replacing the Euclidean distance.

The above WD is computed in the time domain. When it is computed in the frequency domain, WFD [3] is produced. Similarly, the WFD can be converted to Laplacian kernel; and OCSVM is used as the anomaly detector.

All the above unsupervised detectors are summarized in Table 5. Three existing detectors, as used in [1], are also listed. Note that none of the distribution-based detectors we proposed in this section need sliding-window to generate subsequences from a given dataset, unlike the existing detectors.

However, although all distribution-based detectors need to deal with  $s$  subsequences (instead of  $n - m + 1$  subsequences), all four detectors, i.e., OCSMM, OCSVM, kNN and LOF, have  $n^2$  cost because the detectors or/and WD have high cost.

In contrast, the runtime of each of the three IDK-based detectors is dominated by the feature mapping at the first level, i.e.,  $nt\psi$ . The second level mapping has  $st\psi_2$ , except k-IDK has  $s^2$  due to the use of kNN-like operations and it does not have the second level mapping.

**Table 5: Measures and time complexities of (i) distribution-based detectors and (ii) existing detectors which employ a sliding window.  $s = \lfloor n/m \rfloor$ , as stated in Definition 5.**

Detector	Measure	Time cost
IDK <sup>2</sup> [36]	2 levels of IDK	$nt\psi + st\psi_2$
IDK*IK	1 level of IDK & 1 level of IK	$nt\psi + st\psi_2$
k-IDK	1 level of IDK	$nt\psi + s^2$
OCSMM [20]	KME using Gaussian kernel	$n^2$
OCSVM [28]	WD or WFD $\rightarrow$ Laplacian kernel	$n^2$
kNN & LOF [2]	WD	$n^2$
STOMP [42]	Euclidean Distance	$n^2$
NormA [1]	Euclidean Distance	$n$
iForest [17]	No explicit distance measure	$t(n + \psi)\log\psi$

**Table 6: Parameter search ranges. All distribution-based detectors have  $m = L$ , where  $L$  is the period length of each time series, and have two parameters that need to be tuned. NormA & STOMP have one parameter; iForest has two.**

Algorithm	Parameter search ranges
IDK <sup>2</sup> , IDK*IK, k-IDK	$\psi, \psi_2 \in \{2^q \mid q = 1, 2, \dots, 7\}; t = 100$
OCSVM, OCSMM	$\sigma \in \{10^i \mid i = -4, -3, \dots, 0, 1\}$
LOF, kNN, k-IDK	$k \in \{1, 3, 5, 7, 11, 21, 51, 101, 201, 501, 1001, 2001\}$
Wasserstein	$\#\text{bin} = \{10, 20, 50, 100, 200\}; p = 2$
NormA, STOMP, iForest, 1Line	$\omega \in \{L, L \pm 50, L \pm 100\}$
iForest	$\psi \in \{2^q \mid q = 1, 2, \dots, 11, 12\}; t = 100$

Note that both IDK<sup>2</sup> and IDK\*IK cost  $O(n + s)$  only. In contrast, OCSVM, kNN and LOF cost  $O(n^2)$  in addition to the optimization process of WD. STOMP, NormA and iForest need to examine  $n - m + 1$  subsequences and have cost ranging from linear to quadratic.



**Table 7: Results of eleven detectors for anomalous subsequences in terms of AUC. The lowest two AUCs are underlined. The dashline is drawn based on the sorted AUC of 1Line: those datasets in which 1Line produces below or above AUC=0.9.**

Dataset	Length	#Anomalies	Sliding-window				Distribution-based (non-sliding-window)							
			1Line	STOMP	NormA	iForest	OCSMM	OCSVM		LOF	kNN	k-IDK	IDK*IK	IDK <sup>2</sup>
			$\ell_2$	$\ell_2$	$\ell_2$		KME	WLap	WFLap	WD	WD	IDK	IDK	IDK
GPS_trajectory	17175	2	.6	1	<u>.69</u>	1	<u>.40</u>	.80	1	.90	.80	.96	1	1
Patient_respiration	6500	2	.63	<u>.57</u>	.78	.90	<u>.30</u>	.90	.84	.95	.90	.95	.99	1
TEK	15000	3	.64	.83	<u>.44</u>	.92	<u>.67</u>	1	1	1	1	1	1	1
MBA806	100000	27	.69	.89	.88	<u>.82</u>	<u>.67</u>	.91	.85	.91	.90	.92	.93	.93
noisy_sine	10000	3	.73	<u>.25</u>	<u>.28</u>	.75	.75	.99	1	1	.89	1	1	1
mitdb_100_180	5401	1	.73	.95	<u>.55</u>	<u>.77</u>	.85	1	1	.95	.85	.92	.98	1
MBA820	100000	76	.73	<u>.87</u>	.96	.95	<u>.64</u>	.94	.88	.94	.93	.92	.92	.92
dutch_pwrdemand	35040	6	.76	.98	.97	.97	<u>.72</u>	1	<u>.86</u>	1	.99	1	1	1
mitdbx_108	12992	3	.78	.97	.99	.99	<u>.80</u>	1	<u>.92</u>	.97	.99	.96	.96	.99
ann_gun	11251	5	.86	<u>.98</u>	<u>.99</u>	<u>.99</u>	<u>.99</u>	1	<u>.99</u>	<u>.99</u>	<u>.99</u>	1	1	1
MBA14046	100000	142	.86	<u>.85</u>	.91	.98	<u>.89</u>	.90	.90	<u>.89</u>	<u>.89</u>	.94	.92	.93
ARMA	100000	3	.92	1	1	<u>.59</u>	.60	.70	.66	.90	<u>.57</u>	.99	1	1
stdb_308	5400	1	.92	<u>.75</u>	<u>.81</u>	<u>.75</u>	.92	.92	1	1	<u>.75</u>	.83	.88	.93
MBA803	100000	62	.96	<u>.80</u>	.99	.93	.89	.89	.89	.89	<u>.79</u>	.98	.98	.99
MBA805	100000	133	.97	<u>.69</u>	.94	.97	.89	.90	.91	<u>.87</u>	.89	.92	.91	.91
lstdb_20221_43	3750	1	.98	1	.98	1	1	<u>.90</u>	1	1	<u>.86</u>	1	1	1
lstdb_20321_40	3750	1	1	1	.85	1	<u>.71</u>	.88	.88	.88	<u>.59</u>	1	.99	1
qtdbsele0606	3000	1	1	1	1	1	<u>.65</u>	<u>.95</u>	1	<u>.95</u>	<u>.95</u>	1	1	1
Average Ranking			8.64	7.92	7.08	6.08	9.72	5.89	6.33	5.97	8.31	4.56	4.25	3.25

## 6 EXPERIMENTS

We aim to answer the following questions:

- Does the proposed  $\mathbb{R}$  domain distributional treatment produce an effective measure to compute the similarity between two subsequences in a periodic time series?
- Does an  $\mathbb{R}$  domain distributional measure have any advantage over traditional time/frequency domain measures?

To answer the first question, we examine three distributional measures, i.e., Isolation Distributional Kernel (IDK), Kernel Mean Embedding (KME) and Wasserstein distance (WD). They are used in the distributional anomaly detectors described in Section 5. Because of the distributional treatment, a given time series of  $n$  points is converted into  $\lfloor n/m \rfloor$  subsequences, instead of using sliding-window.

To answer the second question, we compare the distributional treatments with two current state-of-the-art methods in time series, STOMP [42] and NormA [1], which employ the Euclidean distance (ED); and one popular point anomaly detector, iForest [17]. The recent frequency domain measure, i.e., Wasserstein-Fourier-Distance (WFLap) [3] and Wasserstein distance (WLap) are employed in OCSVM. Note that STOMP, NormA and iForest require a given time series of length  $n$  to be converted into  $n - m + 1$  subsequences of length  $m$  using sliding-window. We also use a single line (1Line) of standard library Matlab code [39] to check whether anomalous subsequences in a dataset can be easily detected.

The parameter search ranges of all algorithms used are given in Table 6.

For iForest and IDK-based detectors that rely on randomization, we report the average result of 10 trials on each time series.

The chosen periodic time series have been used in the previous works on time series anomaly detection [1, 12, 13, 29].

All these time series are stationary according to each of the three tests, i.e., (a) the Augmented Dickey-Fuller unit root test [5], (b) DF-GLS test [6] (with the p-value threshold of 0.01 for the first two tests), and (c) KPSS test [10] with the p-value threshold of 0.1. These codes are implemented in Python and can be obtained from <https://github.com/bashtage/arch>.

Details of the experimental settings are given in the Appendix.

### 6.1 Result Analysis

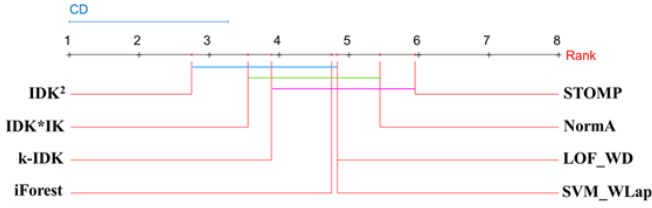
Table 7 provides the anomaly detection accuracy in terms of AUC. The key results are given below.

A direct comparison among the three distributional measures can be summarized as follows:

- The only difference between kNN\_WD and k-IDK is WD vs IDK. IDK is equal to or better than WD in all but three datasets. On seven datasets, the gap in AUC is large ( $> .1$  AUC) in favor of IDK. On the three exceptions where WD is better, the difference is small.
- The three SVM detectors are a direct comparison between KME and WD in time and frequency domains. KME has lower AUC on most datasets than WD. The difference between WD and WFD is small, slightly in favor of WD.

In essence, IDK and WD are better than KME when kNN and SVM detectors are used. This result is consistent with the previous result comparing IDK and KME [36].

In terms of detection accuracy, all three IDK-based detectors are the only detectors that yield the highest or close to the highest



**Figure 2: Friedman-Nemenyi test for the five top-ranked distribution-based and the three sliding-window-based detectors at significance level 0.1. If two algorithms are connected by a CD (critical difference) line, then there is no significant difference between them.**

accuracy in every dataset (out of the 18 datasets) used in our experiment.  $IDK^2$  is the only detector that has  $AUC > 0.9$  on all datasets; and both  $IDK^*IK$  and  $k-IDK$  have only one exception.

The closest contenders are  $LOF\_WD$ , the two OCSVMs employing Wasserstein distance and  $iForest$ , followed by  $NormA$  and  $STOMP$ . The bottom-ranked are  $kNN\_WD$  and  $OCSMM$ .

In general,  $IDK^*IK$  is an improvement over  $k-IDK$  due to the use of level-2 Hilbert space; and  $IDK^2$  is an improvement over  $IDK^*IK$  due to the use of  $IDK$  instead of  $IK$ .

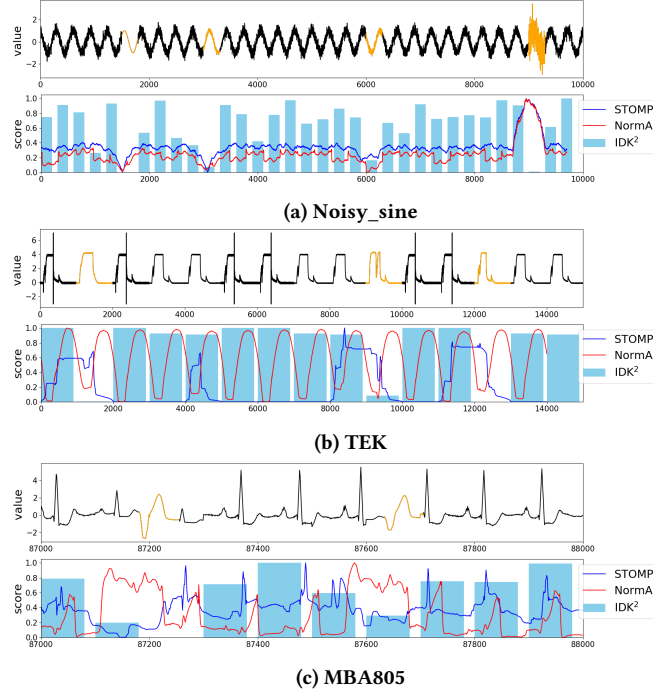
The power of an  $\mathbb{R}$  domain distributional treatment over the traditional sliding-window time domain treatment is demonstrated here that the top-ranked detectors are dominated by detectors which employ  $IDK$  or  $WD$ .

A Friedman-Nemenyi test [4] in Figure 2 shows that the top-ranked  $IDK^2$  is the only detector that is significantly better than  $STOMP$  and  $NormA$ . The second-ranked  $IDK^*IK$  is significantly better than  $STOMP$ . No other detectors are significantly better than  $STOMP$  or  $NormA$ .

With the datasets shown in Figure 3, we find that:

- $NormA$  has difficulty producing a normal model (that consists of mean vectors of clusters of subsequences) that can be differentiated from anomalies on some datasets. This occurs when there are noise and/or anomalies which have small differences from these mean vectors. Examples are shown in Figures 3a and 3b.
- $STOMP$  has problems detecting anomalies of similar characteristics that appear more than once in a time series because of the use 1-NN. An example is  $MBA805$ , shown in Figure 3c. None of the other detectors have this issue. Like  $NormA$ ,  $STOMP$  is also sensitive to noise (see Figure 3a). The effect of both issues could be reduced significantly by using  $kNN$ , which requires to tune parameter  $k$ .

**Summary:** A key determinant of detection accuracy is the similarity measure used. We verified that  $IDK$  and Wasserstein distance are effective in measuring similarity between two subsequences in a periodic time series, without using sliding windows. A common characteristic is that they are both effective in dealing with noisy time series. However,  $IDK$  is better than  $WD$  in dealing with shortened/lengthened subsequences.



**Figure 3: Anomalous and normal subsequences are shown in orange and black, respectively.  $STOMP$  and  $NormA$  return anomaly scores; and  $IDK^2$  returns similarity scores.**

## 6.2 Ablation Studies

Here we provide two ablation studies that examine variants of  $IDK^2$ . These are described in the following two subsections.

**Sliding-window variant of  $IDK^2$ .** To examine the effect of sliding windows on  $IDK^2$ , we create a sliding-window-based variant, denoted as  $s-IDK^2$ . The only change required is in step 1 of Algorithm 1, i.e., to use a sliding window of length  $m$  (instead of non-overlapping windows) to extract subsequences.

The result shown in the first two columns in Table 8 reveals that the difference in AUCs, if any, between  $IDK^2$  and  $s-IDK^2$  is small. Therefore we recommend using  $IDK^2$  in dealing with periodic time series because the non-overlapping treatment runs faster as it produces much fewer subsequences.

Note that, despite the use of a sliding window, the time complexity of  $s-IDK^2$  is still linear, i.e.,  $O(nt\psi + s't\psi/2)$ , where  $s' = n - m + 1$ .

**Variants that employ Gaussian kernel.** To investigate the effect of using a kernel alternative to  $IK$  on  $IDK$ -based detectors, we use Gaussian kernel instead. Three variants are created, denoted as  $GDK-IDK$ ,  $IDK-GDK$  and  $GDK^2$ . They are created by using the feature map of Gaussian kernel approximated from the Nyström method [40] in step 2, step 4, or both steps of Algorithm 1. The sample size of the Nyström method is set to  $\sqrt{n}$  which is also equal to the number of features. The bandwidth of  $GDK$  is searched over  $\{10^m \mid m = -4, -3, \dots, 0, 1\}$ .

The result in the last four columns in Table 8 shows that  $IDK^2$  has the highest detection accuracy in terms of AUC.  $GDK^2$  and  $IDK-GDK$  are not competitive. But the difference between  $IDK^2$



**Table 8: A comparison of Isolation/Gaussian kernel based detectors for anomalous subsequences in terms of AUC. s-IDK<sup>2</sup> is not ranked as it performs similarly to IDK<sup>2</sup>.**

Dataset	s-IDK <sup>2</sup>	IDK <sup>2</sup>	GDK <sup>2</sup>	IDK-GDK	GDK-IDK
GPS_trajectory	1	1	.99	.99	1
Patient_respiration	1	1	.81	.84	.99
TEK	1	1	.8	.91	.96
MBA806	.97	.93	.93	.77	.93
noisy_sine	.98	1	1	.96	1
mitdb_100_180	.95	1	.93	.92	.99
MBA820	.97	.92	.92	.85	.93
dutch_pwrdemand	.99	1	1	.96	1
mitdbx_108	.98	.99	.99	.9	.99
ann_gun	1	1	1	.99	1
MBA14046	.96	.93	.93	.89	.93
ARMA	.99	1	1	.98	1
stdb_308	.95	.93	.68	.75	.83
MBA803	.98	.99	.97	.82	.98
MBA805	.95	.91	.9	.85	.9
lstdb_20221_43	1	1	1	.99	1
lstdb_20321_40	1	1	1	.94	.99
qtdbsele0606	1	1	.96	.99	.98
Average Ranking	—	1.58	2.67	3.69	2.06

and GDK-IDK is small on all datasets, except std\_308. This shows that representing the distribution of subsequences using GDK (or KME via Gaussian Kernel) can also lead to good detection accuracy if IDK is used for anomaly detection. This result again demonstrates the effectiveness of our distributional treatment using Algorithm 1.

It has been shown previously that IDK is a better point anomaly detector than GDK [35]. Thus, the main reason for the poor AUCs of GDK<sup>2</sup> and IDK-GDK is the use of GDK for anomaly detection.

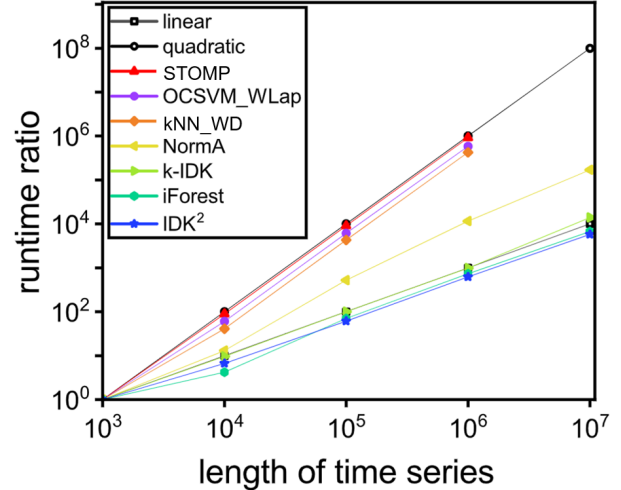
Note that Wasserstein distance (WD) could not be applied in Algorithm 1 because WD does not produce a feature map.

**Table 9: Runtime (in CPU seconds) & ratio of runtime using 10<sup>6</sup> data length and that using 10<sup>3</sup> data length**

Data length	10 <sup>3</sup>	10 <sup>6</sup>	runtime ratio
OCSMM	0.08	74665	933312
STOMP	0.02	18346	917300
OCSVM_WLap	0.02	11701	585050
LOF_WD	0.02	11499	574950
OCSVM_WFLap	0.06	32529	542150
kNN_WD	0.03	12723	424100
NormA	0.07	813	11614
k-IDK	0.65	637	980
IDK*IK	0.72	628	872
iForest	0.51	375	735
IDK <sup>2</sup>	1.06	661	623

### 6.3 Runtime Comparison

Table 9 provides the actual runtimes and runtime ratios when the data size increases from 10<sup>3</sup> to 10<sup>6</sup>. All three IDK-based detectors



**Figure 4: CPU runtime ratio comparison on the MBA14046 (full) dataset. All detectors use  $m = 90$ .**

and iForest have their runtimes increased less than 1000 times. NormA increased by a factor of 12000, i.e., a superlinear increase. OCSMM and STOMP have the highest runtime ratio, with close to a million times increase. OCSVM, LOF, kNN (which employ WD) and STOMP in the same order.

Note that k-IDK, though using the same kNN algorithm, has linear time as opposed to quadratic time for kNN\_WD. This is because WD has quadratic runtime. This shows the runtime advantage of using IDK over WD. The linear time is possible because all pair-wise comparisons need to be conducted for  $\lfloor n/m \rfloor$  subsequences only (not  $n - m + 1$  subsequences as required by sliding-window-based methods such as STOMP).

To avoid overcrowding the graphs, only selected detectors are shown in the comparison given in Figure 4. Notice that the gap between IDK-based detectors and all other contenders (except iForest) widens as the data size increases (at different rates.)

## 7 DISCUSSION

**Recent comparisons with deep learning.** A recent paper has questioned the claims made by deep learning models because many time series datasets used can be solved with similarly high accuracy by using a single line of standard library Matlab code [39]. A different study also found that a deep learning model performed significantly poorer than NormA, even it has been given an unfair advantage of training using normal subsequences only (see [1] for details).

**Do sliding-window-based methods such as STOMP work without sliding windows, using non-overlapped subsequences as in the IDK-based detectors?** We have attempted this, and STOMP’s detection accuracy became much worse. This is not surprising because any sliding-window-based method is sensitive to misalignment between subsequences without using sliding windows. That is the reason why sliding windows are used in the first place to identify similar time-shift subsequences.

**Can IDK-based detectors identify anomalous subsequences of which the values are all the same as normal subsequences**

**but in different order?** An example is to detect ‘21’ as an anomaly in a time series ‘12-12-12-21-12-12’ of which the period is two digits. Here the subsequence ‘21’ is abnormal only because the order of the values in this sequence is not the same as the normal subsequence ‘12’. A simple way to detect such an anomaly using an IDK-based detector is to change the starting step of sliding subsequences. In this example, starting from the second digit, we get subsequences ‘21’, ‘21’, ‘22’, ‘11’, ‘21’ after slicing. Then a distribution-based method such as IDK<sup>2</sup> can easily detect the anomalous subsequences ‘22’ and ‘11’, which correspond to the position of anomalies that we want to detect. In practical applications, we can slice the time series from different starting points within one period. This is sufficient to get the required result. As far as we know, only this kind of anomalies requires such treatment. In the absence of this kind of anomalies, a single run of IDK<sup>2</sup> with any starting point will do the job.

**Further comments on different detectors.** It is interesting to note that Isolation Forest (iForest) [17] with its default setting performed better than or competitive to many more complicated time series anomaly detectors that require a mixture of different algorithms/methods (see Table 4 in [1] for details.) In the context of point anomaly detection, iForest has been shown to be closely related to a kernel-based anomaly detector based on IDK; but iForest is weaker than IDK because of its isolation mechanism and it has no distributional characterization (see Section 7 in [35] for details.)

IDK\*IK is equivalent to using an isolation-based point anomaly detector such as Isolation Forest [17] and iNNE after the given dataset has been mapped to level-1 Hilbert space. A close relationship between isolation-based anomaly detection and IDK anomaly detector has been revealed recently (see Section 7 in [35].)

It is interesting to note that NormA is motivated to tackle noise by using a summarized normal model [1]. The paper has verified that NormA could detect anomalies in a time series corrupted uniformly with Gaussian noise of the same level. However, we showed that, on the noisy\_sine dataset where normal and anomalous subsequences have different noise levels, NormA failed to detect the anomalies.

Our result in Table 7 shows that OCSVM working in the time domain (WLap) may be slightly better than in frequency domain (WFLap); but the difference is small. A recent work shows that WFD is better than Euclidean distance in kNN and logistic classifications [3]. However, they did not compare time versus frequency domains using the same Wasserstein distance as we did.

Recall that IDK (used as level 1 of IDK<sup>2</sup>) is reported to be a better measure than GDK (used as KME in OCSMM) in Section 6, but they are almost equally well when evaluated using an IDK anomaly detector in Section 6.2. This shows that IDK, as an anomaly detector, is able to make full use of the feature map provided by either IDK and GDK, via a distributional characterization, as shown in step 4 of Algorithm 1.

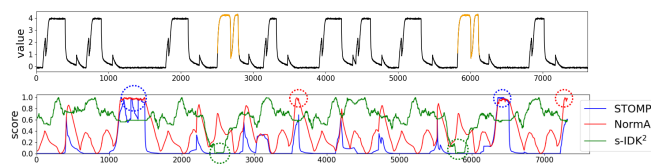
The key advantage of non-sliding window IDK<sup>2</sup> over sliding-window methods such as STOMP and NormA is detection accuracy (results reported in Section 6.1), in addition to the runtime advantage (see Section 6.3). Other non-sliding window methods based on Wasserstein distance and KME have no clear advantage in both detection accuracy and runtime.

**Issues with WD.** Conceptually, WD should be as good as IDK. Our result uncovers weaknesses in WD which could possibly occur in optimizing a transportation plan and the granularity of the

histogram representation. IDK, which needs no optimization and histogram, has no such issues.

**Dealing with aperiodic time series showing recurring normal subsequences.** Many existing methods, including the proposed treatment (IDK<sup>2</sup> being a specific implementation), rely on a time series to be stationary, i.e., having recurring subsequences, in order to differentiate normal (recurring) subsequences from anomalous ones. These include STOMP and NormA which we have used in the comparison. In other words, they are applicable to both periodic as well as aperiodic time series, as long as the time series is stationary. Note that without establishing stationarity in a time series, an analysis becomes intractable [23] (page 152).

Thus, the sliding-window variant s-IDK<sup>2</sup> (studied in Section 6.2) can be expected to detect anomalous subsequences which are dissimilar to the normal subsequences in an aperiodic time series.



**Figure 5: An example of aperiodic time series (only a short interval of the series is shown for clarity). Anomalous and normal subsequences are shown in orange and black, respectively. s-IDK<sup>2</sup> returns a similarity score for each subsequence of time series; STOMP and NormA return anomaly scores. The lowest similarity scores of s-IDK<sup>2</sup> are highlighted in green circles; the highest anomaly scores of both STOMP and NormA are shown in blue circles; and the highest anomaly scores of NormA only in red circles.**

As an example, we create an aperiodic time series of the TEK dataset with normal subsequences recurring aperiodically as shown in Figure 5. Here all methods: STOMP and NormA, s-IDK<sup>2</sup> use a sliding window of length 300.

As shown in the bottom subfigure, s-IDK<sup>2</sup> detects the two anomalies (drawn in orange in the top subfigure) by returning the lowest similarity scores. In contrast, STOMP and NormA produce the highest anomaly scores in normal regions only, and fail to detect the two anomalies.

This example further verifies that s-IDK<sup>2</sup> is more effective in detecting anomalous subsequences than STOMP and NormA. The same issues of STOMP and NormA dealing with periodic time series (stated in Section 6) also exist in aperiodic time series.

**Dealing with time series having multiple periodicity.** Here we examine the detection capabilities of IDK<sup>2</sup>, STOMP and NormA in dealing with a time series having different period lengths at different intervals. Note that all these methods require to set a window size which is slightly larger than the largest period that appears in a time series with multiple periodicity.

To create a time series having multiple periodicity, we concatenate two or four time series of single periodicity as shown in Table 10. Note that apart from different shapes of normal patterns, the period length of each pattern also differs in each time series before they are concatenated.

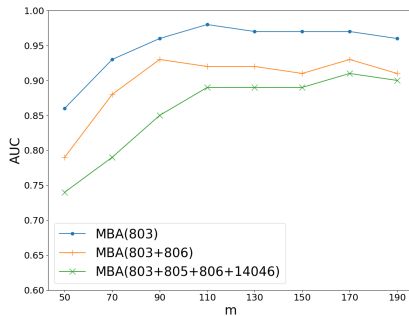
**Table 10: AUC of IDK<sup>2</sup>, STOMP and NormA applied to time series having multiple periodicity.**

Dataset	IDK <sup>2</sup>	STOMP	NormA
MBA(803+805)	.9	.71	.88
MBA(803+806)	.92	.82	.95
MBA(803+820)	.92	.81	.95
MBA(803+14046)	.96	.83	.77
MBA(805+806)	.9	.82	.93
MBA(805+820)	.91	.79	.9
MBA(805+14046)	.93	.8	.66
MBA(806+820)	.88	.89	.95
MBA(806+14046)	.91	.88	.8
MBA(820+14046)	.94	.86	.7
MBA(803+805+806+820)	.89	.82	.92
MBA(803+805+806+14046)	.89	.82	.71
MBA(803+805+820+14046)	.91	.81	.67
MBA(803+806+820+14046)	.91	.87	.73
MBA(805+806+820+14046)	.9	.86	.69
Average Ranking	1.4	2.4	2.2

Table 10 shows that IDK<sup>2</sup> has stable and high anomaly detection accuracy, having at least AUC=0.88 on all fifteen datasets, and it has the highest average rank (shown in the last row). In contrast, STOMP and NormA have their lowest AUCs of 0.71 and 0.66, respectively, and have 10 and 8 datasets with AUC ≤ 0.83, respectively.

In summary, the last two experimental results show that s-IDK<sup>2</sup> deals with aperiodicity, and IDK<sup>2</sup> deals with multiple periodicity a lot better than STOMP and NormA.

**Effects of parameter period length  $m$ .** We evaluate the influence of input parameter  $m$  on IDK<sup>2</sup> using 3 time series. The result in Figure 6 shows that the detection accuracy of IDK<sup>2</sup> becomes stable with a small variation when  $m$  is larger than the (maximal) period length of each time series.



**Figure 6: AUC of IDK<sup>2</sup> with different values of  $m$ .**

## 8 CONCLUSIONS

This paper shows that the insight of the distributional representation for subsequences (stated in Section 3) empowers a transformative treatment for time series. It is a paradigm shift from the time/frequency domain approaches that have been around for more

than one century [15]. It also shows that the paradigm can use existing distributional measures and point anomaly detectors to achieve what would otherwise be impossible in the existing paradigm.

Among the three existing distributional measures, we find that IDK is the best for periodic time series because it is more effective in detecting anomalous subsequences that are shortened/lengthened, generated from a different distribution, and subject to a different noise level. IDK also runs orders of magnitude faster because it needs no additional process apart from the feature mapping, unlike KME (using Gaussian kernel) and Wasserstein distance.

When applied to anomalous subsequence detection, IDK-based detectors are the first method which can achieve linear runtime in practice. This is because IDK is so powerful that it does not need additional learning to accomplish the task with high accuracy. In contrast, both WD and KME require to use OCSVM or kNN, which adds to the already high cost of their similarity calculations.

Although we advocate the use of IDK based on our evaluation, our proposal to use a distributional measure for time series is a generic one, not tied to IDK. Any distributional measures, existing or emerge in the future, can be applied to time series based on the insight revealed in this paper.

We have shown that the proposed distributional treatment works well for periodic as well as aperiodic time series. Our work opens up opportunities to use the same distributional treatment (not limited to IDK) for other data mining tasks in time series. The key to all these endeavors is to define what a subsequence is, to be represented as a set of iid points, generated from an unknown distribution.

## ACKNOWLEDGMENTS

This project is supported by National Natural Science Foundation of China (Grant No. 62076120).

## APPENDIX

The machine used in the experiments has two AMD7742 64-core CPUs with 1024GB memory.

Subsequences in each time series are preprocessed with z-score normalization. All final scores output by detectors are normalized in  $[0, 1]$ . For detectors that rely on randomization, we report the average result of 10 trials on each time series.

**[Datasets]** Synthetic datasets noisy\_sine and ARMA are originated from a previous work [12]. Real-world datasets include MIT-BIH Supraventricular Arrhythmia Database (MBA) [9, 18] and other datasets from various domains have been studied in earlier works [12, 13, 29].

Some datasets have two versions, e.g., ann\_gun and stdb\_308; and each version uses one of the two variables. When either version produces similar AUC for most detectors, we have chosen to use one only. Some datasets are trivial, e.g., chfdb\_chf0175 and qtddb102; and all detectors have the perfect result (AUC=1), so we do not show them in Table 7.

We labelled anomalous periods for each time series following the previous work [12, 13, 29]. Details are given in Table 11. Positions of anomalies in MBA datasets can be seen in folder "MBA\_Annotation".

The period of some datasets varies slightly at different time steps in the series; but it has no effect on the detection accuracy of all

**Table 11: Locations of anomalous periodic subsequences in each dataset in terms of index  $i$  in  $Y_{i,m}$ , where the period length is  $m$ . Other details are in the MBA\_Annotation folder at [github.com/IsolationKernel/Codes/tree/main/IDK/TS](https://github.com/IsolationKernel/Codes/tree/main/IDK/TS).**

Dataset	period length	anomalous period index $i$
noisy_sine	300	6,11,21,31
ARMA	500	101,102,161
GPS_trajectory	2200	3,6
Patient_respiration	150	7,34
TEK	1000	2,10,13
dutch_pwrdemand	672	1,13,18,19,20,52
ann_gun	150	3,15,16,17,19
mitdb_100_180	250	8
mitdbx_108	370	12,28,29,30,31
stdb_308	400	7
lstdb_20221_43	170	5
lstdb_20321_40	200	5
qtdbsele0606	140	9
MBA803	105	see details in folder: MBA_Annotation
MBA805	100	
MBA806	75	
MBA820	100	
MBA14046	90	

algorithms. Our algorithm works well when the subsequence length is set to be roughly the length of the period.

Brief descriptions of some datasets are given as follows.

**dutch\_pwrdemand:** There are a total of 6 anomalous weeks. Some papers [1, 12, 13] use this dataset with fewer anomalies because they treat continuous anomalous weeks as one anomaly.

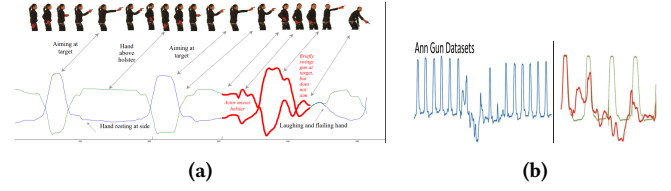
**ann\_gun:** It has only one anomalous period when it was first used in Keogh’s work [13], as shown in Figure 7a. Other anomalous periods in this dataset were later identified [1], and they are shown in Figure 7b.

**Patient\_respiration:** Like the previous work [12], we use the subset that begins at 15500 and ends at 22000 from the nprs44 dataset [13]. There are one apparent anomaly and one subtle anomaly in this dataset as shown in Figure 8a.

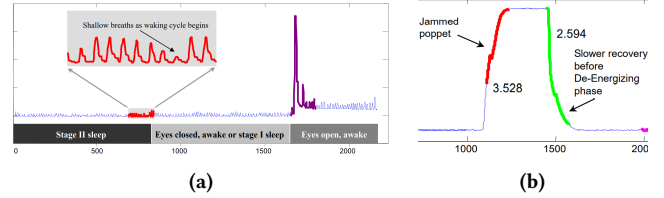
**TEK:** Following the previous work [12], we also concatenate dataset TEK14, TEK16 and TEK17 as TEK of length 15000. In Keogh’s work [13], a total of 4 anomalies are marked. But TEK14 has 2 anomalous snippets belonging to the same period as shown in Figure 8b. Since we regard each anomaly as an anomalous subsequence of one complete period, it is treated as one anomalous periodic subsequence of length 1000. So there are a total of 3 anomalous subsequences in our annotations of this dataset.

**MBA803,MBA805,MBA806,MBA820,MBA14046:** These datasets are subsets of the full MBA dataset, as used in the previous work [1].

**[Algorithms]** The STOMP [42] implementation of MP is used; NormA is from <http://helios.mi.parisdescartes.fr/themisp/norma/>; IDK-based detectors are our implementations based on [36]; and WFD is from [github.com/GAMES-UChile/Wasserstein-Fourier](https://github.com/GAMES-UChile/Wasserstein-Fourier). Others are from scikit-learn.org. All are in Python.



**Figure 7: (a) One anomaly period and (b) additional anomalous periods as identified by [1] in the ann\_gun dataset. The diagrams are extracted from [13] and [1], respectively.**



**Figure 8: Anomalies in (a) Patient\_respiration; (b) a period of TEK14. The diagrams are extracted from [13].**

As for the 1Line method, we use one of the following five types of basic vectorized primitive functions in Matlab as an anomaly score for each sliding window of size  $\omega$ :

- (i)  $\pm\text{diff}(Y)$ : the difference between the current point and the previous point. Here  $\omega = 1$ .
- (ii)  $\pm\text{movmax}(Y, \omega)$
- (iii)  $\pm\text{movmin}(Y, \omega)$
- (iv)  $\pm\text{movmean}(Y, \omega)$
- (v)  $\pm\text{movstd}(Y, \omega)$

where  $Y$  is the time series; and the maximum, minimum, mean or standard deviation is computed for each window of  $\omega$  points.

We run these 5 one-liner on each dataset and report the median AUC (out of the five values) in Table 7. Low median values indicate that the datasets are hard to detect using the 1Line method; otherwise, the datasets have anomalies that can be easily detected.

**[Measures]** The detection accuracy of an anomaly detector is measured in terms of AUC (Area under ROC curve). As all the anomaly detectors are unsupervised learners, all models are trained with the given datasets with no labels. Only after the trained models have made predictions, the ground truth labels are used to compute the AUC for each dataset.

Given a periodic time series  $Y$  of length  $n$  and period length  $m$ , a subsequence  $Y_{i,m}$  of  $Y$  is a subset of contiguous values of length  $m$ , for  $i = 1, \dots, s$ , where  $s = \lfloor n/m \rfloor$ . A distribution-based (non-sliding-window) algorithm outputs a score of each periodic subsequence  $Y_{i,m}$ . Then AUC can be calculated based on scores  $\alpha_i$  for  $Y_{i,m} \forall i = 1, \dots, s$ .

An anomaly detector using the sliding window size  $\omega$  produces a total of  $n - \omega + 1$  subsequences from  $Y$ . When calculating AUC, scores of the sliding subsequences are transformed into periodic subsequence scores as follows: Let  $S_j$  be the anomaly score of subsequence  $Y_{j,\omega}$ , where  $1 \leq j \leq (n - \omega + 1)$ . The final score corresponds to a periodic subsequence  $Y_{i,m}$  is the maximum score of  $S_j \forall j$  such that at least half of  $Y_{j,\omega}$  is included in  $Y_{i,m}$ .



## REFERENCES

- [1] Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2021. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal* (2021), 1–23.
- [2] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. 93–104.
- [3] Elsa Cazelles, Arnaud Robert, and Felipe Tobar. 2021. The Wasserstein-Fourier Distance for Stationary Time Series. *Transactions on Signal Processing* 69 (2021), 709–721.
- [4] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7 (2006), 1–30.
- [5] David A. Dickey and Wayne A. Fuller. 1979. Distribution of the Estimators for Autoregressive Time Series With a Unit Root. *J. Amer. Statist. Assoc.* 74, 366 (1979), 427–431. <http://www.jstor.org/stable/2286348>
- [6] Graham Elliott, Thomas J Rothenberg, and James H Stock. 1992. Efficient tests for an autoregressive unit root.
- [7] Shaghayegh Gharghabi, Shima Imani, Anthony Bagnall, Amirali Darvishzadeh, and Eamonn Keogh. 2020. An ultra-fast time series distance measure to allow data mining in more complex real-world deployments. *Data Mining and Knowledge Discovery* 34 (2020), 1104–1135.
- [8] Omer Gold and Micha Sharir. 2018. Dynamic Time Warping and Geometric Edit Distance: Breaking the Quadratic Barrier. *ACM Transactions on Algorithms* 14, 4, Article 50 (2018).
- [9] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* 101, 23 (2000), e215–e220.
- [10] Bart Hobijn, Philip Hans Franses, and Marius Ooms. 2004. Generalizations of the KPSS-test for stationarity. *Statistica Neerlandica* 58, 4 (2004), 483–502.
- [11] F. Itakura. 1968. Analysis synthesis telephony based on the maximum likelihood method. In *Proceedings of the 6th International Congress on Acoustics*. 280–292.
- [12] Michael Jones, Daniel Nikovski, Makoto Imamura, and Takahisa Hirata. 2016. Exemplar learning for extremely efficient anomaly detection in real-valued time series. *Data mining and knowledge discovery* 30, 6 (2016), 1427–1454.
- [13] E. Keogh, J. Lin, and A. Fu. 2005. HOT SAX: efficiently finding the most unusual time series subsequence. In *Proceedings of the IEEE International Conference on Data Mining*. 226–233.
- [14] Eamonn Keogh and Chotirat Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and Information Systems* 7 (01 2005), 358–386.
- [15] Judy L. Klein. 2005. *Statistical Visions in Time: A History of Time Series Analysis, 1662–1938*. Cambridge University Press.
- [16] Edwin M. Knorr and Raymond T. Ng. 1998. Algorithms for Mining Distance-Based Outliers in Large Datasets. In *Proceedings of the 24th International Conference on Very Large Data Bases*. 392–403.
- [17] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *Proceedings of IEEE International Conference on Data Mining*. 413–422.
- [18] G.B. Moody and R.G. Mark. 2001. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50.
- [19] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. 2017. Kernel Mean Embedding of Distributions: A Review and Beyond. *Foundations and Trends in Machine Learning* 10 (1–2) (2017), 1–141.
- [20] Krikamol Muandet and Bernhard Schölkopf. 2013. One-class Support Measure Machines for Group Anomaly Detection. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*. 449–458.
- [21] John Paparrizos and Luis Gravano. 2016. k-Shape: Efficient and Accurate Clustering of Time Series. *SIGMOD Record* 45, 1 (2016), 69–76.
- [22] John Paparrizos, Chunwei Liu, Aaron J. Elmore, and Michael J. Franklin. 2020. Debunking Four Long-Standing Misconceptions of Time-Series Distance Measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1887–1905.
- [23] Kun Il Park. 2018. *Fundamentals of Probability and Stochastic Processes with Applications to Communications*. Springer Publishing Company, Incorporated.
- [24] Xiaoyu Qin, Kai Ming Ting, Ye Zhu, and Vincent Cheng Siong Lee. 2019. Nearest-Neighbour-Induced Isolation Similarity and Its Impact on Density-Based Clustering. In *Proceedings of The Thirty-Third AAAI Conference on Artificial Intelligence*. 4755–4762.
- [25] L. Rueschendorf. 2002. Wasserstein metric. In *Encyclopedia of Mathematics*. EMS Press.
- [26] H. Sakoe and S. Chiba. 1971. A Dynamic Programming Approach to Continuous Speech Recognition. In *Proceedings of the 7th International Congress on Acoustics*. 65–69.
- [27] H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1 (1978), 43–49.
- [28] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. 2001. Estimating the Support of a High-Dimensional Distribution. *Neural Computing* 13, 7 (2001), 1443–1471.
- [29] Pavel Senin, Jessica Lin, Xing Wang, Tim Oates, Sunil Gandhi, Arnold P Boedihardjo, Crystal Chen, and Susan Frankenstein. 2015. Time series anomaly discovery with grammar-based compression. In *Proceedings of the 18th International Conference on Extending Database Technology*. 481–492.
- [30] Yilin Shen, Yanping Chen, Eamonn Keogh, and Hongxia Jin. 2018. Accelerating Time Series Searching with Large Uniform Scaling. In *Proceedings of the SIAM International Conference on Data Mining*. 234–242.
- [31] Robert Shumway and David Stoffer. 2017. *Time Series Analysis and Its Applications With R Examples, Fourth Edition*. Springer.
- [32] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. 2007. A Hilbert Space Embedding for Distributions. In *Algorithmic Learning Theory*. Marcus Hutter, Rocco A. Servedio, and Eiji Takimoto (Eds.). Springer, 13–31.
- [33] Chang Wei Tan, François Petitjean, and Geoffrey Webb. 2019. Elastic bands across the path: A new framework and method to lower bound DTW. In *Proceedings of the SIAM International Conference on Data Mining*. 522–530.
- [34] Kai Ming Ting, Jonathan R. Wells, and Takashi Washio. 2021. Isolation Kernel: The X Factor in Efficient and Effective Large Scale Online Kernel Learning. *Data Mining and Knowledge Discovery* 35 (2021), 2282–2312.
- [35] Kai Ming Ting, Bi-Cun Xu, Takashi Washio, and Zhi-Hua Zhou. 2020. Isolation Distributional Kernel: A New Tool for Kernel based Anomaly Detection. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 198–206.
- [36] Kai Ming Ting, Bi-Cun Xu, Takashi Washio, and Zhi-Hua Zhou. 2021. Isolation Distributional Kernel: A New Tool for Point and Group Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering* (2021). 10.1109/TKDE.2021.3120277
- [37] Kai Ming Ting, Yue Zhu, and Zhi-Hua Zhou. 2018. Isolation Kernel and its effect on SVM. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2329–2337.
- [38] Matteo Togninalli, Elisabetta Ghisu, Felipe Llinares-López, Bastian Rieck, and Karsten Borgwardt. 2019. Wasserstein Weisfeiler-Lehman Graph Kernels. In *Advances in neural information processing systems*.
- [39] Renjie Wu and Eamonn Keogh. 2021. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [40] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. 2012. Nyström method vs random fourier features: A theoretical and empirical comparison. *Advances in neural information processing systems* 25 (2012), 476–484.
- [41] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*. 1317–1322.
- [42] Yan Zhu, Zachary Zimmerman, Nader Shakibay Senobari, Chin-Chia Michael Yeh, Gareth Funning, Abdullah Mueen, Philip Brisk, and Eamonn Keogh. 2016. Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins. In *Proceedings of the IEEE 16th International Conference on Data Mining*. 739–748.