Multi-Kernel Times Series Outlier Detection

Florian Kalinke[0000−0002−0443−6288], Edouard Fouché[0000−0003−0157−7648], Haiko Thiessen, and Klemens Böhm

Karlsruhe Institute of Technology (KIT)
florian.kalinke@kit.edu, edouard.fouche@kit.edu, haikothiessen@gmail.com, klemens.boehm@kit.edu

Abstract. Time series are sequences of observations ordered by time. Detecting outliers in a set of time series is very important for many use cases, including fraud detection and predictive maintenance. However, this task continues to be difficult: First, time series may be of different lengths and conventional distance measures like the Euclidean distance can not capture their similarity well. Workarounds like feature engineering require domain knowledge and render solutions domain-specific. Second, many existing techniques are supervised, but training labels are expensive if not impossible to obtain. In this paper, we propose Multi-Kernel Times Series Outlier Detection (MK-TSOD), a method that combines the Fourier Transform, Global Alignment Kernels, and Multiple Kernel Learning with Support Vector Data Description. We describe its specifics, and show that MK-TSOD outperforms existing methods on standard benchmark data.

Acknowledgements This work was supported by the DFG Research Training Group 2153: "Energy Status Data — Informatics Methods for its Collection, Analysis and Exploitation". This version of the article has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/978-3-031-45275-8_46.

Keywords: Time Series · Outlier Detection · Global Alignment Kernel · Fourier Transform · Support Vector Data Description.

1 Introduction

Outlier detection is of fundamental importance for many real-world applications, such as fraud detection or predictive maintenance. In such settings, data is often collected over time; the data has the form of time series. In the literature on time series, "outlier" either refers to anomalous subsequences [24] or to anomalous full sequences [14]. This article addresses the latter, that is, detecting few outlying time series from a set of time series.

Outlier detection in time series continues to be challenging, for two reasons: (1) First, most outlier detection algorithms rely on a notion of **distance** to

quantify data dissimilarity. Yet, time series may have different lengths and be shifted in time, which makes classic distance measures (e.g., the Euclidean distance) inadequate. As a workaround, many existing techniques rely on extracted features instead of directly comparing the series by a distance measure. However, extracting features limits the applicability of respective algorithms and generally leads to a loss of information. (2) Second, the outlier detection problem is **unsupervised** in nature and typically imbalanced, that is, outliers are rare, so that optimizing the parameters of outlier detectors is hardly feasible in practice.

One common way to tackle both problems is using dynamic time warping (DTW; [20]) together with Support Vector Data Description (SVDD; [23]). SVDD is a kernel-based approach that encloses a predefined share of the data within a hypersphere of minimal volume; points outside the sphere are outliers. The kernel function quantifies the dissimilarity of the data in an implicit feature space. However, DTW does not yield a valid kernel function; the theory [21,22] supporting support vector-based approaches does not hold when using DTW with SVDD [8].

In this paper, we propose a kernel-based method for time series outlier detection that addresses all challenges identified above:

We propose Multi-Kernel Time Series Outlier Detection (MK-TSOD). Our idea is to combine SVDD with multiple kernels, which can capture frequency information of time series with fast Fourier transform, and time information with Global Alignment Kernels (GAK; [8]). Unlike DTW, GAK is guaranteed to work with SVDD. We combine the time and frequency information in an optimal way with Multiple Kernel Learning (MKL; [18]). MK-TSOD has one parameter, the expected outlier ratio, which is intuitive to set.

We run extensive experiments on standard benchmark data. They reveal that the proposed method outperforms the existing approaches on 9 out of 15 data sets with the balanced accuracy metric. We release the implementation together with our experiments on GitHub.[1]

Paper outline: Section 2 presents related work. Section 3 presents the definitions and the existing elements our method uses. Section 4 introduces the proposed approach. The experiments are in Section 5, and Section 6 concludes.

2 Related Work

While outlier detection has been well addressed for numerous types of data, e.g., numerical, categorical, mixed, or text data, detecting outliers from time series remains particularly challenging.

Due to the lack of proper distance measures for time series, most outlier detectors use extracted features instead. Examples are Highest Density Regions (HDR; [15]), and α-hull [15]: HDR extracts features of the time series and then applies Principal Component Analysis (PCA) to project the features to the first two principal components. It then estimates the local density of each observation.

[1] `https://github.com/flopska/mk-tsod/`

Observations whose density is below a threshold are the outliers. α-hull is similar to HDR as it also uses PCA. However, instead of using a density-based approach, it relies on α-convex hulls. Both algorithms use the same set of extracted features.

Finding a good set of features tends to be difficult. Established approaches to find such sets are either expert- or algorithm-based. The expert-based ones are costly and require domain knowledge. The algorithm-based ones, e.g., [6,16], only target classification and regression, i.e., supervised settings.

DOTS (Detection of Outlier Time Series; [3]) does not rely on extracted features, but clusters the data based on DTW and then uses the entropy to find an optimal positioning of cluster centers. It ranks the "outlierness" of observations based on their distances to the clusters. DOTS has more free parameters than our approach, and several of them are difficult to optimize in an unsupervised setting, e.g., the regularization parameter λ and the number of clusters k.

ADSL (Anomaly Detection algorithm with shapelet-based Feature Learning; [2]) is so far the approach most related to ours, as it also bases on SVDD. The main difference is that ADSL applies SVDD to a learned intermediate representation, namely shapelets, but not explicitly to time series data, as we do. It then classifies outliers based on their distance to the shapelets. However, this approach is only successful in cases where shapelets are indeed a meaningful representation.

The DeepSVDD [19] approach combines ideas from neural networks with the outlier detection paradigm of SVDD, that is, it learns a hypersphere encompassing the networks representation of most observations with minimal volume. — Similarly to SVDD, DeepSVDD labels points outside this sphere as outliers. However, the algorithm is limited to data with fixed length and thus not applicable to time series.

While theoretically unsound, support vector-based approaches with DTW-based kernels can work in practice [13]. DTW-SVDD, ADSL, DOTS, HDR, and α-hull form a set of strong baselines against which we compare in Section 5.

3 Background

This section summarizes the definitions (Section 3.1). To render the article fully self-contained, we recall SVDD (Section 3.2), and GAK (Section 3.3).

3.1 Definitions

A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel on an input space $\mathcal{X}$ if there exists a real Hilbert space $\mathcal{H}$ and a map $\phi : \mathcal{X} \to \mathcal{H}$ such that $k(x, x') = \langle \phi(x), \phi(x') \rangle$ for all $x, x' \in \mathcal{X}$. We call ϕ the feature map and $\mathcal{H}$ the feature space of k [22]. The corresponding Gram matrix for a subset $\{x_1, \ldots, x_l\} \subseteq \mathcal{X}$ is the symmetric $l \times l$ matrix $\mathbf{K}_k = [k(x_i, x_j)]_{i,j=1}^{l} \in \mathbb{R}^{l \times l}$.

A time series x with length n is a sequence $x := (x_m)_{m=1}^{n}$ with $x_m \in \mathbb{R}$ for $m = 1, \ldots, n$. In what follows, we consider the input space $\mathcal{X} = \{x_1, \ldots, x_l\}$, i.e., a set of l time series with potentially different lengths. The proposed method is

straightforward to extend to $\mathbb{R}^D$ ($D \in \mathbb{N}$), but for clarity, we consider observations taking values in $\mathbb{R}$ in what follows.

With those definitions, detecting outlying time series can be seen as finding the $l \cdot \theta$ time series that are the most dissimilar within a set of time series $\mathcal{X}$, where the parameter $\theta \in (0, 1)$ specifies the expected ratio of outliers in that set. Since outliers are rare, θ is typically small.

3.2 Support Vector Data Description (SVDD)

The construction of SVDD is similar to the well-known SVM (Support-Vector Machine; [7]). In short, the goal is to solve the constrained optimization problem

$$\min R^2 + C \sum_{i=1}^{l} \xi_i, \quad \text{s.t.} \quad \|\phi(x_i) - a\|^2 \leq R^2 + \xi_i,\ \xi_i \geq 0,$$

for $i = 1, \ldots, l$, that is, to find a sphere with center a and radius R^2 so that most observations are enclosed. The slack variables ξ_i allow points to lie outside the sphere with a penalty controlled by parameter C. [23] recommends setting

$$C = 1/(l \cdot \theta), \tag{1}$$

where θ is the expected ratio of outliers in the data, and l the size of the data set. Hence, θ can be chosen intuitively. The corresponding dual problem is

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{l} \alpha_i k(x_i, x_i) - \sum_{i,j=1}^{l} k(x_i, x_j), \\ &\text{s.t.} \sum_{i=1}^{l} \alpha_i = 1,\ 0 \leq \alpha_i \leq C, \end{aligned} \tag{2}$$

for all $i = 1, \ldots, l$ and with Lagrange multipliers $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_l)^\mathsf{T}$.

Having obtained a solution to (2), a time series $z \in \mathcal{X}$ is an outlier if and only if

$$\|\phi(z) - a\|^2 = k(z, z) - 2 \sum_{i=1}^{l} \alpha_i k(z, x_i) + \sum_{i,j=1}^{l} \alpha_i \alpha_j k(x_i, x_j) > R^2, \tag{3}$$

with the radius R^2 computed as

$$R^2 = k(x_k, x_k) - 2 \sum_{i=1}^{l} \alpha_i k(x_i, x_k) + \sum_{i,j=1}^{l} \alpha_i \alpha_j k(x_i, x_k), \tag{4}$$

with any $x_k \in \mathcal{X}$ for which the corresponding Lagrange multiplier α_k fulfills $0 < \alpha_k < C$.

3.3 Global Alignment Kernels (GAK)

GAKs [8] extend DTW to the kernel setting. The definition of GAK bases on the notion of alignment: An alignment π of length p between $x, y \in \mathcal{X}$ of lengths n, n' is a pair (π_1, π_2) that fulfills the following conditions:

Boundary & Monotonicity. The first observation in x must map to the first observation in y and analogously for the last observations. Also, the alignment must be increasing. Formally, one has

$$\begin{aligned} 1 = \pi_1(1) \leq \cdots \leq \pi_1(p) = n, \\ 1 = \pi_2(1) \leq \cdots \leq \pi_2(p) = n'. \end{aligned} \tag{5}$$

Continuity. There must not be any gap in the alignment path, i.e., each observation must map to at least one other observation. Further, there must not be any repetition. Formally, for all $1 \leq i, j \leq p-1$,

$$\begin{aligned} \pi_1(i+1) \leq \pi_1(i) + 1, \pi_2(j+1) \leq \pi_2(j) + 1, \\ (\pi_1(i+1) - \pi_1(i)) + (\pi_2(i+1) - \pi_2(i)) \geq 1. \end{aligned} \tag{6}$$

Adjustment Window. Given an observation $x_i \in \mathbb{R}$, $i = 1, \ldots, n$ of time series $x \in \mathcal{X}$ and parameter T, x_i must map to an observation $y_i \in \mathbb{R}$, $i = 1, \ldots, n'$ of $y \in \mathcal{X}$ that is "sufficiently close", i.e., strictly less than T steps away and vice versa. Formally, for all $1 \leq i \leq p-1$

$$|\pi_1(i) - \pi_2(i)| < T. \tag{7}$$

While not strictly necessary, the adjustment window condition speeds up the computation by reducing the number of alignments considered without impacting result quality by much [8]; we confirm this in our experiments.

The kernel k_{GAK} sums all distances computed over alignments that satisfy (5), (6), and (7):

$$k_{GAK}(x, y) = \sum_{(x', y') \in \mathcal{M}(n, n')} \mathbf{k}(x', y'), \tag{8}$$

with $\mathcal{M}(n, n') = \{(x'_{\pi_1}, y'_{\pi_2}) \mid \pi = (\pi_1, \pi_2) \in \mathcal{A}(n, n')\}$, where $\mathcal{A}(n, n')$ is the set of all valid alignments, and where $\mathbf{k}(x'_{\pi_1}, y'_{\pi_2}) = \prod_{i=1}^{|\pi|} \kappa(x'_{\pi_1(i)}, y'_{\pi_2(i)})$ for a so-called local kernel κ.

[8] shows that k_{GAK} is not positive definite for all such κ. This is problematic, as (2) is then non-convex, and the global optimum might be not be found. Additionally, the theory that supports kernel functions does not hold in such cases. However, [8] proves that $\kappa/(1+\kappa)$ being positive definite is a sufficient condition to guarantee that k_{GAK} is positive definite and show that this holds for the local kernel,

$$\kappa(x, y) = \exp\left\{-\frac{\|x - y\|^2}{2\sigma^2} - \log\left(2 - e^{-\frac{\|x-y\|^2}{2\sigma^2}}\right)\right\},$$

where, by abuse of notation, $x, y \in \mathbb{R}$ in our case, and $\|\cdot\|$ the Euclidean distance.

In turn, DTW is defined as the minimum distance over all valid alignments

$$\mathrm{DTW}(x,y) = \min_{\pi \in \mathcal{A}(n,n')} \sum_{i=1}^{|\pi|} \left\| x_{\pi_1(i)} - y_{\pi_2(i)} \right\|^2, \quad x, y \in \mathcal{X},$$

with the corresponding DTW kernel

$$k_{DTW}(x,y) = \exp\left\{-\gamma \cdot \mathrm{DTW}(x,y)\right\}. \tag{9}$$

As DTW does not fulfill the triangle inequality, the kernel k_{DTW} is not guaranteed to be positive definite, a problem one avoids with GAK.

GAK and DTW have a recursive formulation that one can compute with dynamic programming. So their complexity when comparing two time series of lengths n and n' and dimensionality d is $\mathcal{O}(dnn')$. As GAK only considers alignments within a band of width T, its runtime reduces to $\mathcal{O}(dT\min(n,n'))$.

GAK and DTW only consider the time information of the respective time series. But it is known that considering the frequency information can prove beneficial when working with time series. The proposed method that we present next builds upon this observation.

4 Multi-Kernel Time Series Outlier Detection

Depending on the characteristics of a time signal that one wishes to highlight, it is common to represent the signal in the time or in the frequency domain. Accordingly, we propose a kernel k_{FFT} (Section 4.1) that considers similarities in the frequency domain, which we then combine with k_{GAK} in an optimal fashion with Multiple Kernel Learning (MKL; [18]). This guarantees that the proposed method (Section 4.2) detects outliers by taking both time and frequency information into account. We analyze the runtime complexity of MK-TSOD in Section 4.3.

4.1 Fast Fourier Transform Kernels

The Fourier transformation of a time series $x = (x_m)_{m=1}^n$ is the sequence $X = (X_k)_{k=1}^n$ of the Fourier coefficients

$$X_k = \sum_{m=1}^{n} x_m \exp\left\{-2\pi i \frac{(k-1)(m-1)}{n}\right\}, \quad k = 1, \ldots, n,$$

with $i^2 = -1$ the imaginary number. Let $x, y \in \mathcal{X}$ be time series of lengths n, n', having Fourier coefficients $X = (X_k)_{k=1}^n$, $Y = (Y_k)_{k=1}^{n'}$, respectively. To compare x and y, we propose k_{FFT} as a modified Gaussian kernel that truncates the sequence of Fourier coefficients, that is,

$$k_{FFT}(x,y) := \exp\left\{-\gamma \sum_{j=1}^{t} (X_j - Y_j)^2\right\}, \tag{10}$$

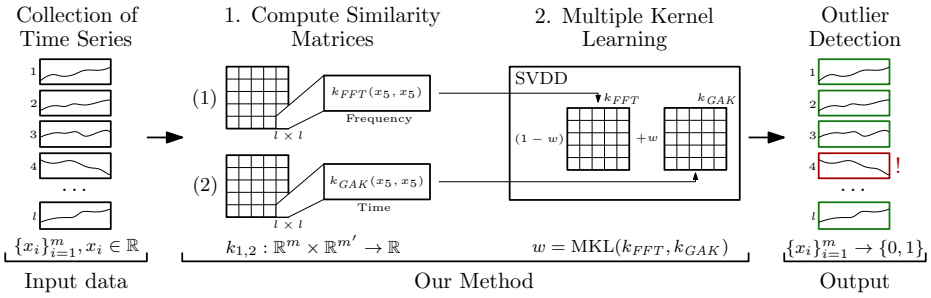

Fig. 1: Schematic representation of the proposed outlier detection method.

with smoothing parameter γ, and $1 \leq t \leq \min(n, n')$. Hence, parameter t controls the quality of the approximation by restricting the number of coefficients.

To select the bandwidth parameter γ in an unsupervised fashion, we use an argument from [12], which states that dissimilarities in the input space and dissimilarities in the feature space behave similarly:

$$\frac{\delta_1}{\delta_2} \approx \frac{\exp\left(-\gamma\delta_1^2\right)}{\exp\left(-\gamma\delta_2^2\right)},$$

where we denote by $\delta_i = \|\cdot\|$ $(i \in \{1, 2\})$ the Euclidean distance between the truncated Fourier transformations of two arbitrary observations. One solves for γ and sets

$$\gamma = \frac{-\ln\left(\frac{\delta_{\min}}{\delta_{\text{avg}}}\right)}{\delta_{\text{avg}}^2 - \delta_{\min}^2}, \tag{11}$$

with the quantities $\delta_{\min} := \left\|x_q - x_{1-\mathrm{NN}(q)}\right\|$, $\delta_{\text{avg}} := \frac{1}{n-1}\sum_{i \neq q}\|x_q - x_i\|$, and $q := \arg\min_{1 \leq i \leq n}\left\|x_i - x_{1-\mathrm{NN}(i)}\right\|$. Here, $1 - \mathrm{NN}(k)$ denotes the index of the nearest neighbor of x_k, i.e., the transformation with the smallest distance in the frequency domain to x_k. Hence, x_q is the time series with the smallest distance to its nearest neighbor. $\delta_{\min}$ is the smallest distance between the Fourier coefficients of any two time series, and δ_{avg} is the average distance of all time series to x_q w.r.t. their Fourier coefficients. (11) allows accounting for the characteristics of the frequencies observed.

4.2 MK-TSOD Algorithm

Figure 1 provides an intuitive schematic representation of the proposed algorithm, which we elaborate in what follows.

To merge kernel k_{GAK} and the proposed kernel k_{FFT}, we first recall a property of kernels [22, Lemma 4.5] that allows their combination. We then detail how we adapt MKL to SVDD in order to optimize over the free parameter that results from the kernel combination, and conclude the section with the presentation and runtime analysis of the full algorithm.

Lemma 1 (Additivity). *Let $\mathcal{X}$ be a set, $\beta \geq 0$, and k, k_1, and k_2 be kernels on $\mathcal{X}$. Then βk and $k_1 + k_2$ are kernels on $\mathcal{X}$ as well.*

With Lemma 1, a convex combination with weight $w \in [0, 1]$ of GAK kernel k_{GAK} and the proposed kernel k_{FFT} is a valid kernel that takes the form

$$k(x, y) = w \cdot k_{GAK}(x, y) + (1 - w) \cdot k_{FFT}(x, y),$$

and incorporates information of both the time and the frequency domain of $x, y \in \mathcal{X}$.

More generally, the MKL problem [18] is to find the Lagrange multipliers α_i of a kernel machine and the weights $\mathbf{w} = (w_1, \ldots, w_M)^\mathsf{T}$ for a convex combination k of kernels k_m given by

$$k(x, y) = \sum_{m=1}^{M} w_m k_m(x, y), \text{ s.t. } w_m \geq 0 \wedge \sum_{m=1}^{M} w_m = 1. \tag{12}$$

It follows from Lemma 1 and an induction argument that (12) defines a valid kernel. To find the solution, we proceed as follows:

The Lagrangian of (2) is

$$L = \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) - \sum_i \alpha_i k(x_i, x_i).$$

Combining this with $k(x, y)$ from (12), we obtain the MKL problem for SVDD

$$L = \sum_{i,j=1}^{l} \alpha_i \alpha_j \sum_{m=1}^{M} w_m k_m(x_i, x_j) - \sum_{i=1}^{l} \alpha_i \sum_{m=1}^{M} w_m k_m(x_i, x_j).$$

To optimize w.r.t. $\mathbf{w}$, [18] propose SimpleMKL, a gradient descent-based approach. Hence, we compute the partial derivative w.r.t. w_m, which for SVDD takes the form

$$\frac{\partial L}{\partial w_m} = \boldsymbol{\alpha}^\mathsf{T} \mathbf{K}_m \boldsymbol{\alpha} - \boldsymbol{\alpha}^\mathsf{T} \mathrm{diag}(\mathbf{K}_m)$$

with Gram matrix $\mathbf{K}_m$ associated with kernel k_m, and then apply their framework: In the present case, $M = 2$, $k_1 = k_{GAK}$, and $k_2 = k_{FFT}$. Performing the gradient descent optimization yields a weight w so that the volume of the hypersphere is again minimized.

Algorithm 1 presents MK-TSOD in full. The method has a total of five parameters. We recommend values for T, σ^2, and t in Section 5.1. Parameter γ is set according to (11); C is set by (1).

4.3 Complexity Analysis

The runtime of MK-TSOD depends on that of computing the Gram matrices for kernels k_{FFT}, k_{GAK}, and on that of solving the MKL problem. For a worst-case scenario, we assume that the longest time series is of length n, and that

Algorithm 1 MK-TSOD

Require: Time series $\mathcal{X} = \{x_1, \ldots, x_l\}$, outlier ratio θ

$C \leftarrow 1/(l \cdot \theta)$ ▷ Equation (1)
$\mathbf{K}_{k_{FFT}} = [k_{FFT}(x_i, x_j)]_{ij}$ for $i, j = 1, \ldots, l$ ▷ Equation (10)
$\mathbf{K}_{k_{GAK}} = [k_{GAK}(x_i, x_j)]_{ij}$ for $i, j = 1, \ldots, l$ ▷ Equation (8)
$\mathbf{w}, \boldsymbol{\alpha} \leftarrow \mathrm{MKL}(\mathbf{K}_{k_{FFT}}, \mathbf{K}_{k_{GAK}}, C)$ ▷ Equation (12)
$\mathbf{K} \leftarrow w_1 \cdot \mathbf{K}_{k_{FFT}} + (1 - w_1) \cdot \mathbf{K}_{k_{GAK}}$
$R^2 \leftarrow (\mathbf{K})_{kk} - 2\sum_{i=1}^{l} \alpha_i (\mathbf{K})_{ik} + \boldsymbol{\alpha}^\mathsf{T} \mathbf{K} \boldsymbol{\alpha}$ ▷ Equation (4)
outliers $\leftarrow \emptyset$
for $x_i \in \mathcal{X}$ **do**
    if $(\mathbf{K})_{ii} - 2\sum_{j=1}^{l} \alpha_j (\mathbf{K})_{ij} + \boldsymbol{\alpha}^\mathsf{T} \mathbf{K} \boldsymbol{\alpha} > R^2$ **then** ▷ Equation (3)
        outliers $\leftarrow$ outliers $\cup\, x_i$
return outliers

one observes l time series. Then the runtime of k_{GAK} per pair of observations is in $\mathcal{O}\left(n^2\right)$ [8]. As the Gram matrix computes all pairwise combinations, its computational cost is $\mathcal{O}\left(n^2 l^2\right)$. Computing the Fourier coefficients of a time series of length n has a complexity of $\mathcal{O}\left(n \log(n)\right)$, and, by the same reasoning as before, obtaining the corresponding Gram matrix costs $\mathcal{O}\left(n \log(n) l^2\right)$. The worst-case bound for an optimal solution of SVDD is $\mathcal{O}\left(l^3\right)$ [4]. As the number of SimpleMKL iterations is bounded and does not depend on l [18], running SimpleMKL does not affect the worst-case estimate. Putting the previous estimates together, we obtain a total runtime complexity of $\mathcal{O}\left(n^2 l^2 + n \log(n) l^2 + l^3\right)$.

While the worst-case complexity is relatively high, the actual runtime is reasonable for practical applications and often lower than that of competitors, as our experiments show. In practice, one typically uses an approximate solver, such as sequential minimal optimization (SMO; [17]), which yields a solution to the SVDD problem (2) in $\mathcal{O}\left(l^2\right)$; this reduces the runtime cost.

5 Experiments

In our experiments, we compare the proposed technique to the state of the art, both in terms of outlier detection quality and runtime; we also conduct a parameter sensitivity and ablation analysis. We start by describing the experiment setup (Section 5.1), collect the results w.r.t. balanced accuracy, runtime, and parameter sensitivity in Section 5.2, and compare to ablations in Section 5.3.

5.1 Setup

Metrics & Evaluation. Our experiments evaluate the balanced accuracy (BA), which is commonly used for outlier detection tasks. We repeat each experiment 10 times, keeping the normal data but sampling a different set of outliers, and report the mean score and standard deviation. We run all algorithms on a server running Ubuntu 20.04 with 124 GB RAM, and 32 cores with 2 GHz each.

Table 1: Summary of the 15 data sets. *Length* is the length of the time series in the respective data set, *#N / #O* is the absolute count of normal and outlying observations, and *#C (N)* is the number of classes in the original data set together with the class set as normal.

Data set	Length	#N / #O	#C (N)
ArrowHead	251	65 / 3	3 (2)
CBF	128	310 / 16	3 (2)
Ch.Concent.	166	1000 / 52	3 (1)
ECG200	96	133 / 7	2 (1)
ECGFiveDays	136	442 / 23	2 (1)
GunPoint	150	100 / 5	2 (1)
Ham	431	103 / 5	2 (1)
Herring	512	77 / 4	2 (1)
Lightning2	637	73 / 3	2 (1)
MoteStrain	84	685 / 36	2 (1)
Strawberry	235	351 / 18	2 (1)
ToeSeg1	277	140 / 7	2 (0)
ToeSeg2	343	124 / 6	2 (0)
Wafer	152	6402 / 336	2 (1)
Wine	234	57 / 3	2 (1)

Data Sets & Data Preparation. We follow the approach by [11,10] and adapt time series classification data sets from the UCR repository [1,9] to our setting. To improve comparability, our process mirrors the selection and preprocessing from [2] but we exclude data sets with fewer than 50 time series [2, Table 1], due to their small size. Specifically, for binary classification problems, we choose the majority class as "normal" class, and for multiclass classification problems, we set the class that is visually the most distinct as "normal". We then sample 5% of the observations from the respective other class(es), which constitute the "outliers". This yields 15 diverse data sets. Training sets include the outliers, as this is closer to real-world settings, but the outliers are regenerated between runs. Table 1 summarizes the respective characteristics of the data sets.[2]

Configurations. In the following, we detail the parameter settings for each algorithm. We start with a recommendation for the parameters of our algorithm.

MK-TSOD. We set the regularization parameter C as in (1) with an expected outlier ratio $\theta = 0.05$. For k_{GAK}, we follow the recommendation of [8] and set $\sigma^2 = a^2 \cdot \text{median}(\|\mathbf{x} - \mathbf{y}\|) \cdot \sqrt{\text{median}(|\mathbf{x}|)}$, with $a = 1.5$, and $T = b \cdot \text{median}(|\mathbf{x}|)$, with $b = 0.5$. We vary factors a, b in the parameter sen-

[2] We abbreviate the data sets "ChlorineConcentration", "ToeSegmentation1", and "ToeSegmentation2" as "Ch.Concent.", "ToeSeg1", and "ToeSeg2", respectively.

sitivity analysis. To solve the optimization problem (2), we use libsvm[3], which we adapt to use precomputed Gram matrices with SVDD. We set the smoothing parameter γ of the Fourier transform-based kernel using (11), and set $t = 20$, based on the parameter analysis we present at the end of the section. The code for reproducing our experiments is available on GitHub.[4]

SVDD with DTW. As a baseline, we use k_{DTW} (9) with SVDD; recall that while the approach is theoretically unsound, it has shown good results in practice. Because of the absence of a heuristic, we set $\gamma = 1$. We set the regularization parameter C as in MK-TSOD.

HDR and α-hull. We use the reference implementations provided by the authors together with the recommended parameters.[5]

DOTS. We set the regularization parameter λ to 0.045, and the number of medoids k to the number of classes per data set, as recommended by the authors, and use their reference implementation.[6] To compute the BA, we cut off the ranking based on the expected ratio of outliers, which is 0.05.

ADSL. We set the maximum number of iterations to 1000, $k = 0.02$, and $l = 0.2$, as in [2]. We obtained the code from the authors.

LOF with DTW. As an additional baseline, we combine the well-known Local Outlier Factor (LOF; [5]) with DTW in place of the Euclidean distance. As LOF is sensitive to the amount of neighbors n to consider, we set $n \in \{5, 10, 20\}$ and report the best results.

5.2 Results

Performance. Table 2 shows the average Balanced Accuracy (BA).[7] N/A indicates that the respective algorithm did not complete a single run in 24 hours.

One sees that MK-TSOD achieves the best score on 9 out of 15 data sets with the BA metric, and that LOF-DTW performs second best, that is, it performs better than the competitors.

Runtime. We measure the absolute runtime of each algorithm w.r.t. the number and length of times series. We use the data set featuring the longest time series, Lightning2, and simulate different input configurations. To vary their length, we sub- or oversample the measured points. To vary the size of the set, we sub- or oversample the time series themselves. When oversampling, we add Gaussian noise with a standard deviation of 10^{-3}. This mimics that real-world data does not consist of duplicates only.

Figure 2 shows our results. MK-TSOD is slower than HDR and α-hull but faster than ADSL and DOTS w.r.t. the number of time series. However, the slope of MK-TSOD and DOTS is similar, so differences in runtime might be due

[3] `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`
[4] `https://github.com/flopska/mk-tsod/`
[5] `https://github.com/robjhyndman/anomalous-acm`
[6] `https://github.com/B-Seif/anomaly-detection-time-series`
[7] Here, MK denotes MK-TSOD; DTW denotes DTW-SVDD.

Table 2: Mean BA over 10 runs. Bold print highlights the best results.

Data set	MK	DTW	HDR	DOTS	α-hull	ADSL	LOF-DTW
ArrowHead	**0.70 ± 0.2**	0.58 ± 0.2	0.67 ± 0.1	0.51 ± 0.1	0.67 ± 0.2	0.49 ± 0.0	0.52 ± 0.1
CBF	**0.66 ± 0.0**	0.49 ± 0.1	0.50 ± 0.0	0.49 ± 0.0	0.50 ± 0.0	0.50 ± 0.0	0.65 ± 0.1
Ch.Concent.	0.49 ± 0.0	0.48 ± 0.0	0.50 ± 0.0	0.50 ± 0.0	0.50 ± 0.0	0.50 ± 0.0	**0.63 ± 0.0**
ECG200	**0.67 ± 0.1**	0.55 ± 0.1	0.50 ± 0.0	0.55 ± 0.1	0.50 ± 0.1	0.52 ± 0.0	0.65 ± 0.1
ECGFiveDays	0.64 ± 0.0	0.58 ± 0.0	0.52 ± 0.0	0.54 ± 0.0	0.52 ± 0.0	0.50 ± 0.0	**0.77 ± 0.0**
GunPoint	**0.72 ± 0.1**	0.61 ± 0.1	0.49 ± 0.0	0.64 ± 0.1	0.50 ± 0.0	0.62 ± 0.1	0.70 ± 0.1
Ham	**0.51 ± 0.1**	0.48 ± 0.1	0.49 ± 0.0	0.48 ± 0.0	0.49 ± 0.0	0.49 ± 0.0	0.49 ± 0.0
Herring	**0.52 ± 0.1**	0.51 ± 0.1	0.50 ± 0.1	0.50 ± 0.1	0.47 ± 0.0	0.50 ± 0.0	0.50 ± 0.1
Lightning2	0.57 ± 0.2	0.49 ± 0.2	0.48 ± 0.0	0.50 ± 0.1	0.51 ± 0.1	0.64 ± 0.1	**0.72 ± 0.2**
MoteStrain	**0.70 ± 0.0**	0.62 ± 0.1	0.52 ± 0.0	0.61 ± 0.0	0.52 ± 0.0	0.51 ± 0.0	0.55 ± 0.0
Strawberry	0.69 ± 0.1	0.70 ± 0.0	0.47 ± 0.0	0.68 ± 0.0	0.48 ± 0.0	0.56 ± 0.0	**0.76 ± 0.0**
ToeSeg1	0.65 ± 0.1	0.50 ± 0.1	0.49 ± 0.0	0.47 ± 0.0	0.48 ± 0.0	0.61 ± 0.0	**0.73 ± 0.1**
ToeSeg2	**0.67 ± 0.1**	0.48 ± 0.1	0.51 ± 0.0	0.48 ± 0.0	0.52 ± 0.0	0.60 ± 0.0	0.61 ± 0.1
Wafer	**0.65 ± 0.0**	0.64 ± 0.0	0.49 ± 0.0	N/A	0.49 ± 0.0	0.50 ± 0.0	0.56 ± 0.0
Wine	0.48 ± 0.1	0.50 ± 0.1	0.60 ± 0.1	0.56 ± 0.1	**0.65 ± 0.2**	0.54 ± 0.1	0.58 ± 0.1

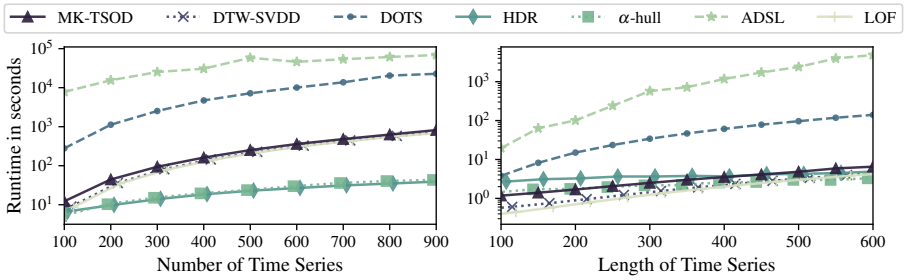

Fig. 2: Runtime analysis. We report the median runtime of five independent runs.

to implementation details. The difference to DTW-SVDD and LOF-DTW is negligible. Regarding the length of time series, the figure shows that MK-TSOD and DTW-SVDD scale better than ADSL and DOTS, but worse than HDR and α-hull. Again, LOF-DTW scales similar to MK-TSOD, as expected.

Parameter Sensitivity Analysis. We study the sensitivity of MK-TSOD w.r.t. parameters σ, T (the smoothness and the width of the window of k_{GAK}), and t (the number of Fourier coefficients for k_{FFT}). Figure 3 shows the average results obtained over all data sets from Table 1. When varying one parameter, we keep the others fixed at their recommended values.

We see that for changes in σ, BA stays nearly constant from $x = 1.5$. The figure also shows that the width T of the band considered for alignments does not influence the result by much. However, we see a slight increase for the BA metric at $T = 0.2$. This indicates that focusing on local similarities proves beneficial for the data sets considered. For the number of Fourier coefficients t, we see that the best performance is obtained for $t = 20$, with a slight decline for larger

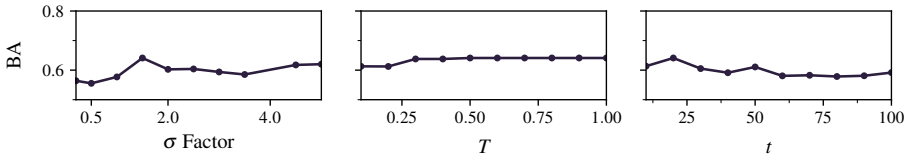

Fig. 3: Influence of the factors a, b for the median heuristics for σ, T, and influence of parameter t. We report the median BA of five independent runs.

values. We hypothesize that using more than 20 coefficients approximates the time series too closely, and the algorithm cannot generalize, that is, it overfits. Altogether, we see that MK-TSOD is robust w.r.t. its parameters.

5.3 Ablation Analysis

We consider alternative designs of the proposed method. Instead of combining multiple kernels, we run SVDD with the individual kernels k_{FFT} (FFT-SVDD) and k_{GAK} (GAK-SVDD) and compare their results to the ones obtained with MK-TSOD in terms of the average balanced accuracy over five draws of outliers. The settings of the individual kernels are the same as in Section 5.1.

Table 3: Ablation analysis. Mean BA over five runs. Bold print highlights the best results.

Data set	MK-TSOD	FFT-SVDD	GAK-SVDD
ArrowHead	$\mathbf{0.70 \pm 0.2}$	0.65 ± 0.1	0.61 ± 0.2
CBF	$\mathbf{0.66 \pm 0.0}$	0.60 ± 0.1	$\mathbf{0.66 \pm 0.0}$
Ch.Concent.	0.49 ± 0.0	$\mathbf{0.52 \pm 0.0}$	0.48 ± 0.0
ECG200	$\mathbf{0.67 \pm 0.1}$	0.63 ± 0.1	0.62 ± 0.1
ECGFiveDays	0.64 ± 0.0	$\mathbf{0.65 \pm 0.0}$	0.62 ± 0.1
GunPoint	$\mathbf{0.72 \pm 0.1}$	0.65 ± 0.1	0.64 ± 0.1
Ham	$\mathbf{0.51 \pm 0.1}$	0.47 ± 0.1	0.50 ± 0.1
Herring	$\mathbf{0.52 \pm 0.1}$	0.49 ± 0.1	0.51 ± 0.1
Lightning2	0.57 ± 0.2	$\mathbf{0.67 \pm 0.1}$	0.47 ± 0.1
MoteStrain	$\mathbf{0.70 \pm 0.0}$	0.62 ± 0.0	0.67 ± 0.1
Strawberry	0.69 ± 0.1	0.71 ± 0.1	$\mathbf{0.73 \pm 0.0}$
ToeSeg1	$\mathbf{0.65 \pm 0.1}$	$\mathbf{0.65 \pm 0.1}$	0.62 ± 0.1
ToeSeg2	$\mathbf{0.67 \pm 0.1}$	0.55 ± 0.1	0.57 ± 0.1
Wafer	$\mathbf{0.65 \pm 0.0}$	0.62 ± 0.0	$\mathbf{0.65 \pm 0.0}$
Wine	0.48 ± 0.1	$\mathbf{0.54 \pm 0.1}$	0.42 ± 0.0

Table 3 shows the results (with the standard deviation) of our ablation study. MK-TSOD achieves the largest BA on 10 out of 15 datasets; SVDD with the k_{FFT} kernel performs best on 5 datasets, and SVDD with the k_{GAK} kernel has the best score on 3 data sets (including ties). The proposed algorithm can leverage the respective strengths of the kernels. Considering the unsupervised setting, where parameter optimization — including kernel selection — is typically infeasible in practice, this result indicates that MK-TSOD provides a good default choice, often improving performance over employing a single kernel.

6 Conclusions

This paper tackles the long-standing problem of detecting outliers in a set of time series, for which we propose a new method, MK-TSOD. It builds on SVDD and combines global alignment and Fourier transform kernels, taking the time and frequency information of time series into account. The parameters of MK-TSOD are either intuitive to set or we recommend heuristics. Our evaluation shows that MK-TSOD achieves state-of-the-art performance, and outperforms existing approaches w.r.t. the balanced accuracy metric on 9 out of 15 standard benchmark data sets.

References

1. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery **31**, 606–660 (2017)
2. Beggel, L., Kausler, B.X., Schiegg, M., Pfeiffer, M., Bischl, B.: Time series anomaly detection based on shapelet learning. Computational Statistics **34**, 945–976 (2019)
3. Benkabou, S., Benabdeslem, K., Canitia, B.: Unsupervised outlier detection for time series by entropy and dynamic time warping. Knowledge and Information Systems **54**(2), 463–486 (2018)
4. Bordes, A., Ertekin, S., Weston, J., Bottou, L.: Fast kernel classifiers with online and active learning. Journal of Machine Learning Research (JMLR) **6**, 1579–1619 (2005)
5. Breunig, M.M., Kriegel, H., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: SIGMOD Conference. pp. 93–104 (2000)
6. Christ, M., Braun, N., Neuffer, J., Kempa-Liehr, A.W.: Time series feature extraction on basis of scalable hypothesis tests (tsfresh - A python package). Neurocomputing **307**, 72–77 (2018)
7. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (1995)
8. Cuturi, M.: Fast global alignment kernels. In: International Conference on Machine Learning (ICML). pp. 929–936 (2011)
9. Dau, H.A., Keogh, E., et al.: The UCR Time Series Classification Archive (2018)
10. Emmott, A., Das, S., Dietterich, T.G., Fern, A., Wong, W.: Systematic construction of anomaly detection benchmarks from real data. Tech. rep. (2015), (`http://arxiv.org/abs/1503.01158`)

11. Emmott, A.F., Das, S., Dietterich, T., Fern, A., Wong, W.K.: Systematic construction of anomaly detection benchmarks from real data. In: ACM SIGKDD Workshop on Outlier Detection and Description. pp. 16–21 (2013)
12. Ghafoori, Z., Erfani, S.M., Rajasegarar, S., Bezdek, J.C., Karunasekera, S., Leckie, C.: Efficient unsupervised parameter estimation for one-class support vector machines. Neural Networks and Learning Systems **29**(10), 5057–5070 (2018)
13. Gudmundsson, S., Runarsson, T.P., Sigurdsson, S.: Support vector machines and dynamic time warping for time series. In: International Joint Conference on Neural Networks (IJCNN). pp. 2772–2776 (2008)
14. Gupta, M., Gao, J., Aggarwal, C.C., Han, J.: Outlier detection for temporal data: A survey. Knowledge and Data Engineering **26**(9), 2250–2267 (2013)
15. Hyndman, R.J., Wang, E., Laptev, N.: Large-scale unusual time series detection. In: International Conference on Data Mining Workshop (ICDMW). pp. 1616–1619 (2015)
16. Patel, D., Shah, S.Y., Zhou, N., Shrivastava, S., Iyengar, A., Bhamidipaty, A., Kalagnanam, J.: Flops: On learning important time series features for real-valued prediction. In: IEEE BigData 2020. pp. 1624–1633 (2020)
17. Platt, J.: Sequential minimal optimization: A fast algorithm for training support vector machines (1998), `https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/`
18. Rakotomamonjy, A., Bach, F., et al.: SimpleMKL. Journal of Machine Learning Research (JMLR) **9**, 2491–2521 (2008)
19. Ruff, L., Görnitz, N., Deecke, L., Siddiqui, S.A., Vandermeulen, R.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International Conference on Machine Learning (ICML). vol. 80, pp. 4390–4399 (2018)
20. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978)
21. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
22. Steinwart, I., Christmann, A.: Support Vector Machines. Springer (2008)
23. Tax, D.M.J., Duin, R.P.W.: Support Vector Data Description. Machine Learning **54**(1), 45–66 (2004)
24. Vercruyssen, V., Meert, W., Davis, J.: "now you see it, now you don't!" detecting suspicious pattern absences in continuous time series. In: SIAM International Conference on Data Mining (SDM). pp. 127–135 (2020)