Robust Time Series Dissimilarity Measure for Outlier Detection and Periodicity Detection

Anonymous
Anonymous Institution

ABSTRACT

Dynamic time warping (DTW) is an effective dissimilarity measure in many time series applications. Despite its popularity, it is prone to noises and outliers, which leads to singularity problem and bias in the measurement. The time complexity of DTW is quadratic to the length of time series, making it inapplicable in real-time applications. In this paper, we propose a novel time series dissimilarity measure named RobustDTW to reduce the effects of noises and outliers. Specifically, the RobustDTW estimates the trend and optimizes the time warp in an alternating manner by utilizing our designed temporal graph trend filtering. To improve efficiency, we propose a multi-level framework that estimates the trend and the warp function at a lower resolution, and then repeatedly refines them at a higher resolution. Based on the proposed RobustDTW, we further extend it to periodicity detection and outlier time series detection. Experiments on real-world datasets demonstrate the superior performance of RobustDTW compared to DTW variants in both outlier time series detection and periodicity detection.

CCS CONCEPTS

• Mathematics of computing → Time series analysis; **• Computing methodologies → Anomaly detection**.

KEYWORDS

time series, DTW, periodicity detection, outlier detection

ACM Reference Format:
Anonymous. 2022. Robust Time Series Dissimilarity Measure for Outlier Detection and Periodicity Detection. In *ACM '22: The xx ACM International Conference on XXX, XX. xx–xx, 2022, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.xxxx

1 INTRODUCTION

Nowadays, time series (TS) signal processing and mining have received lots of research interests [2, 14, 16]. Among them, dynamic time warping (DTW) [25, 26, 28, 36, 37] is a popular method to compute the dissimilarity between two TS by finding the optimal alignment. It has been widely employed in many tasks involving distance computation such as TS similarity search [27], outlier detection [5, 12], clustering and classification [17], periodicity detection [13], etc. Due to its simplicity and versatility, it has been an effective tool in many areas, including speech recognition, signal processing, machine learning, and bioinformatics, etc.

Although DTW solves the time warping problem, it suffers from some limitations. Firstly, in practice time series data is often contaminated by noises and outliers. The DTW only considers the stretching or shrinking along the time axis but ignores to handle noises and outliers in the value axis, which may bias the distance between time series and lead to the singularity problem where a single point in one TS is mapped to a large subsection in the other TS. The robustness of time series distance is crucial in many time series applications, such as outlier time series detection and periodicity detection. For example, in outlier TS detection the goal is to identity outlier TS different from others. In this scenario, some sporadic noises or outliers in a normal time series may significantly increase the distance between it and others using DTW, and thus reporting a false positive outlier TS. Secondly, the time and space complexity of DTW are quadratic to the length of TS. This make it difficult to be applied in long sequence analysis. Recently, some works [11, 15, 18, 21, 29] are proposed to deal with either singularity or complexity problem. However, existing methods cannot robustly and efficiently address both challenges simultaneously.

In this paper, we propose a novel dissimilarity measure called RobustDTW, which addresses the aforementioned challenges. To handle the noises and outliers in TS, we propose a general temporal graph trend filtering to estimate the true signal. Compared with classical DTW and its variants, in RobustDTW we optimize the time warp function and perform the trend filtering simultaneously. To further accelerate computation, we propose a multi-level framework to learn the time warp function and perform trend filtering recursively. We perform downsampling recursively to get different levels of representations of TS. Then we start from a lower resolution representation to calculate the time warp function. And the noises and outliers are handled properly by the proposed temporal graph trend filtering, where the graph is constructed based on the learned time warp function. Meanwhile, the singularity problem is mitigated in this multi-level framework. To the best of our knowledge, this is the first paper to design a general temporal graph trend filtering extended for time warping, which is further incorporated into a novel multi-level framework for efficient computation.

To demonstrate the effectiveness of RobustDTW, we apply it in outlier time series detection by identifying abnormal TS given a set of TS. By integrating the proposed RobustDTW with the popular local outlier factor (LOF) algorithm [8, 22, 31], we can successfully identify these abnormal TS from noisy data. To further justify our proposed RobustDTW algorithm, we also extend it to the periodicity detection task. Empirically it outperforms state-of-the-art periodicity detection algorithms [24, 30, 34], as it can robustly handle non-stationary TS with noises, outliers, and complicated periodic patterns.

ACM '22, XX. xx–xx, 2022, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.xxxx

2 RELATED WORK

One issue of the DTW is the singularity introduced by noises and outliers in time series. To mitigate this problem, the derivative DTW [20] is proposed to compute the shape information by considering the first derivative of the sequences. Recently, [11] proposes to compute the optimal time warp function by solving an optimization problem which considers the misalignment of time warping with two penalization terms for the cumulative warping and the instantaneous rate of time warping. In [21], an adaptive constrained DTW is proposed by introducing two adaptive penalties to mitigate the singularity problem. Another issue of classical DTW algorithm is the quadratic time and space complexity which makes it a bottleneck in many applications. To mitigate this problem, the most widely used one is FastDTW [29], which is an efficient approximate algorithm with linear time and space complexity. It adopts a multi-level approach to compute the optimal time warp function, from the coarsest level and refine it repeatedly. Similar works consider applying wavelet transform in the multi-level framework can be found in [3, 15, 18]. Another way to speed up DTW is to use searching constraints. In [19], the small constraint and lower bounding are utilized to skip some computation in DTW. Some empirical comparisons are reported in [35]. Generally, when the length of time series is relatively small (less than several thousands), and the warping path is close to the diagonal path, the constrain-based methods outperforms. However, as the time series length increases and the warping path deviate significantly, the multi-level methods are better choices [29]. Besides, some recent works extend DTW to differentiable versions like Soft-DTW [10] and DP-DTW [9], which can be integrated into deep learning networks. Note that we are not dealing with differentiable DTW in this paper. Instead, also different from excising work focusing on either mitigating the singularity problem or reducing time and space complexity, we aim to solve both problems in one-shot, which are also illustrated in periodicity detection and outlier time series detection applications.

3 METHODOLOGY

3.1 Proposed RobustDTW Algorithm

To make DTW robust to noise and outliers while preserving its flexibility of dynamic index matching, we employ the robust trend to filter out the noises and outliers. Specifically, we propose to estimate the time warp function and detrend TS in an alternating manner. We iteratively conduct the following operations: (1) fix the time warp function and estimate the detrended TS u and v; (2) fix the estimated detrended TS u and v, and estimate the time warp function $\varphi(t)$ based on u and v.

To speed up the computation of RobustDTW, we utilize the similarity of the shapes and alignments of TS pairs among different resolutions, and combine alternating of DTW align adjustment and trend estimate with the progress from low resolution to high resolution. During this multi-level progress, low resolution trend results are the starting values for high resolution estimation, as low resolution index alignment can suggest the constraint of path searching space for high resolutions. We summarize the detailed procedure of the proposed RobustDTW algorithm in Alg. 1, followed by detailed description of each step.

Algorithm 1 RobustDTW with Multi-Level Framework

Input: $\mathbf{x}$, $\mathbf{y}$, input TS; λ_i^j regularization parameter configurations; t, iteration times; r, radius.
Output: normalized RobustDTW distance.
Step 1: Robust self-detrending: apply robust trend filter on $\mathbf{x}$, $\mathbf{y}$, respectively, to get $\mathbf{u}$, $\mathbf{v}$ as the filtered trend components.
Step 2: Multi-Level Representation: recursively downsample $\mathbf{u}$ and $\mathbf{v}$, and record the intermediate results.
Step 3: Projection and Upsampling: in the first iteration, run FastDTW [29] to get warping path and use the highest level downsampled $\mathbf{u}$ and $\mathbf{v}$ as initial trend estimates; otherwise, perform upsample trend estimates and project DTW warping path with from previous level to current level.
Step 4: Time Warping Alignment: refine warping alignment by DTW with projected constraint from step 3.
Step 5: Temporal Graph Detrending: generate graph from updated warping from Step 4 and refine graph detrending with updated trend estimates.
Step 6: Iterative Processing: repeat Steps (3)-(5) until lowest level to return normalized RobustDTW distance.

Step 1: Robust Self-Detrending
To roughly remove the effects of outliers and noise, we first adopt robust trend filtering for individual TS. Let denote the TS of length n as $\mathbf{y} = [y_1, y_2, \cdots, y_n]^T$, which can be decomposed into trend and residual components [1]

$$y_t = \tau_t + r_t \quad \text{or} \quad \mathbf{y} = \boldsymbol{\tau} + \mathbf{r} \tag{1}$$

where $\boldsymbol{\tau} = [\tau_1, \tau_2, \cdots, \tau_n]^T$ denotes the trend component, and $\mathbf{r} = [r_1, r_2, \cdots, r_n]^T$ denotes the residual component, which contains noises and outliers. Motivated by the RobustTrend filter [33], the trend components of the two input time series $\mathbf{x}$ and $\mathbf{y}$ can be robustly obtained by the following self-detrending model

$$\begin{aligned} \mathbf{u} &= \mathrm{argmin}_u \; g_\gamma(\mathbf{x}, \mathbf{u}) + \lambda_1^{\mathrm{self}} ||\mathbf{D}^{(1)}\mathbf{u}||_1 + \lambda_2^{\mathrm{self}} ||\mathbf{D}^{(2)}\mathbf{u}||_1, \\ \mathbf{v} &= \mathrm{argmin}_v \; g_\gamma(\mathbf{y}, \mathbf{v}) + \lambda_1^{\mathrm{self}} ||\mathbf{D}^{(1)}\mathbf{v}||_1 + \lambda_2^{\mathrm{self}} ||\mathbf{D}^{(2)}\mathbf{v}||_1, \end{aligned} \tag{2}$$

where $g_\gamma(\mathbf{x}) = \sum_i g_\gamma(x_i)$ is the summation of elementwise Huber loss function with each element as

$$g_\gamma(x_i) = \begin{cases} \frac{1}{2}x_i^2, & |x_i| \le \gamma \\ \gamma|x_i| - \frac{1}{2}\gamma^2, & |x_i| > \gamma \end{cases}. \tag{3}$$

And the $D^{(1)} \in \mathbf{R}^{(N-1)\times N}$ and $D^{(2)} \in \mathbf{R}^{(N-2)\times N}$ are the first-order and second-order difference matrix, respectively:

$$\mathbf{D}^{(1)} = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & & \\ & & 1 & -1 \end{bmatrix}, \; \mathbf{D}^{(2)} = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & & \ddots & & \\ & & & 1 & -2 & 1 \end{bmatrix}.$$

Note that the Huber loss in the self-detrending model (2) brings robustness to outliers, while the two spare L1 regulations in (2) captures both slow and abrupt trend changes.

Step 2: Multi-Level Representation
Multi-level representations is achieved by downsampling filtered

trend TS from Step 1 by factor of 2 iteratively as follows

$$\mathbf{u}_{\ell-1} \xrightarrow{upsample} \mathbf{u}_{\ell-1}^{\uparrow} = \mathbf{u}_{\ell},\ \mathbf{v}_{\ell-1} \xrightarrow{upsample} \mathbf{v}_{\ell-1}^{\uparrow} = \mathbf{v}_{\ell} \tag{4}$$

where $\mathbf{u}_1 = \mathbf{u}, \mathbf{v}_1 = \mathbf{v}, \ell = 1, 2, ..., t$. This procedure is conducted until the level t representation is obtained, where t is the total levels in the framework. Note that higher level representations correspond to lower resolution TS.

Step 3: Projection and Upsampling
Starting from level t TS, call FastDTW [29] to get warping index alignment $\boldsymbol{\pi}_t$ and use current level representations as the base trend estimation. In the subsequent iterations, upsample previous level's warping path ($\boldsymbol{\pi}_\ell$) by 2 and add extra searching width defined by parameter radius r to generate the searching constraint, which is called projection. That is

$$\boldsymbol{\pi}_\ell \xrightarrow{upsample} \boldsymbol{\pi}_\ell^{\uparrow},\ \mathcal{A}(\mathbf{u}, \mathbf{v})_{\ell-1} = \{\text{path set}|\text{center } \boldsymbol{\pi}_\ell^{\uparrow},\ \text{radius } r\}.$$

Step 4: Time Warping Alignment
Refine the warping path by running DTW with projected warping constrain ($\mathcal{A}(\mathbf{u}, \mathbf{v})_{\ell-1}$) to obtain the time warp function ($\varphi(t)$) and path ($\boldsymbol{\pi}_{\ell-1}$) at current level as

$$\boldsymbol{\pi}_{\ell-1}, \varphi(t) = \min_{\boldsymbol{\pi} \in \mathcal{A}(\mathbf{u},\mathbf{v})_{\ell-1}} \sqrt{\sum_{(t,\varphi(t)) \in \boldsymbol{\pi}} d(u_t, v_{\varphi(t)})^2}. \tag{5}$$

Step 5: Temporal Graph Detrending
In this step, we aim to estimate the detrended TS $\mathbf{u}$ and $\mathbf{v}$ more accurately by considering not only its neighbors within itself but also its similar peer time series. Let $G = (V, E)$ be an graph, with vertices $V = \{1, \cdots, n\}$ and undirected edges $E = \{e_1, \cdots, e_s\}$, and suppose that we observe $\mathbf{y} = [y_1, \cdots, y_n]^T \in R^n$ over each node. Similarly, the kth order graph trend filtering estimates $\boldsymbol{\tau} = [\tau_1, \cdots, \tau_n]^T$ can be obtained by solving:

$$\text{argmin}_{\tau \in R^n} \frac{1}{2}\|\mathbf{y} - \boldsymbol{\tau}\|_2^2 + \lambda\|\Delta^{(k+1)}\boldsymbol{\tau}\|_1, \tag{6}$$

where $\Delta^{(k+1)}$ is the graph difference operator of order $k+1$. When $k = 0$, the 1st order graph difference operator $\Delta^{(1)}$ penalizes all local differences on all edges as $\|\Delta^{(1)}\boldsymbol{\tau}\|_1 = \sum_{(i,j)\in E} |\tau_i - \tau_j|$. We can represent $\Delta^{(1)}$ in a matrix form as $\Delta^{(1)} \in \{-1, 0, 1\}^{s\times n}$ where $s = |E|$, i.e., number of edges. Specifically, let $e_\ell = (i, j)$, then $\Delta^{(1)}$ has the ℓth row as $\Delta_\ell^{(1)} = [0, \cdots, -1, \cdots, 1, \cdots, 0]$, i.e., $\Delta_\ell^{(1)}$ has -1 at the ith position, and 1 at the jth position. Similar to trend filtering on univariate TS, we can define the higher order graph difference operators recursively. In particular, the graph difference operators defined above reduce to the ones defined on the univariate TS in which $V = \{1, 2, \cdots, n\}$ and $E = \{(i, i+1) : i = 1, 2, \cdots, n-1\}$.

In this paper, we design a general temporal graph detrending for multivariate TS by extending the idea of the graph-based detrending in Eq. (6). The key idea is to incorporate the relationship between TS in detrending which can also deal with the lagging effect adaptively. Another distinguishing feature is that we introduce weight for each edge in the graph, instead of only allowing binary weight as in [32]. Specifically, we construct the graph of two TS based on the alignment at step 4. In the constructed graph G, we have $m + n$ vertices, and each vertex corresponds to a time point in $\mathbf{x} \in R^m$ and $\mathbf{y} \in R^n$. For notation simplicity, we denote the set of vertices as $E = \{x_1, x_2, \cdots, x_m, y_1, y_2, \cdots, y_n\}$. Next, we describe

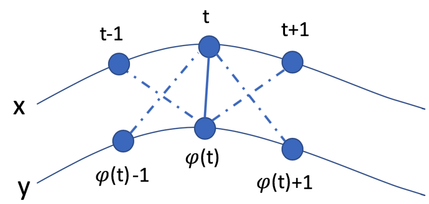

Figure 1: Illustration of the graph construction between two TS of the RobustDTW. Each data points are connected not only to their neighbors in the same TS but also to their peers of the other TS which are aligned by DTW. The time stamps on x are $t-1$, t, $t+1$, and φ(t) is the mapping of index from x to y. So the aligned points are $t-1$, t, $t+1$ on x to $\varphi(t)-1$, $\varphi(t)$, $\varphi(t)+1$ on y.

how to construct edges and their corresponding weights. Firstly, each vertex x_t is connected its left neighbor x_{t-1} and right neighbor x_{t+1}. Secondly, x_t should be connected to its peer in $\mathbf{y}$, i.e., $y_{\varphi(t)}$. Thirdly, to avoid errors introduced in DTW of step 4, we construct more edges to improve robustness. For the edge $x_t \leftrightarrow y_{\varphi(t)}$, we introduce additional 4 edges, including $x_t \leftrightarrow y_{\varphi(t)-1}, x_t \leftrightarrow y_{\varphi(t)+1}$, $x_{t-1} \leftrightarrow y_{\varphi(t)}$, $x_{t+1} \leftrightarrow y_{\varphi(t)}$, as illustrated in the Fig. 1. Note that in Fig. 1 we only consider the direct one neighbor of x_t and $y_{\varphi(t)}$. In practice, we can increase the size of the neighborhood based on the noises and outliers of data accordingly.

After constructing the graph G for a pair of TS, the designed temporal graph detrending is computed as:

$$\mathbf{u},\mathbf{v} = \underset{\mathbf{u},\mathbf{v}}{\text{argmin}} \frac{1}{2}\|\mathbf{w} - [\mathbf{x};\mathbf{y}]\|_2^2 + \lambda_1^{\text{GD}}\|D_{\text{G}}^{(1)}\mathbf{w}\|_1 + \lambda_2^{\text{GD}}\|D_{\text{G}}^{(2)}\mathbf{w}\|_1, \tag{7}$$

where $\mathbf{w} = [\mathbf{u}; \mathbf{v}]$ is the concatenate of input $\mathbf{u}$ and $\mathbf{v}$, $D_{\text{G}}^{(1)}$ and $D_{\text{G}}^{(2)}$ are the 1st and 2nd graph difference operators used to capture abrupt and slow trend changes. Note that both self-detrending (2) and temporal graph detrending (7) can be efficiently solved by the alternating direction of multipliers (ADMM) algorithm [7].

Step 6: Iterative Processing
To achieve better performance, we repeat Steps 3 to 5 to update the time warp function and the trend estimates until convergence. In our empirical studies, we found that three iterations are often enough for convergence. Furthermore, when we compute the time warp function in the higher resolution, we use the time warp function computed in the previous step as the initialization, which significantly reduces computational complexity.

4 EXPERIMENTS AND APPLICATIONS

4.1 Setting and Efficiency Comparison

For the parameter setting of RobustDTW, we have the following results. For self-trend filtering, the larger λ_1 and λ_2 would yield a smoother trend. It is observed that λ_1 and λ_2 in trend filtering are relatively insensitive. For the γ in Huber loss, its value is fixed with 1 as the common value used in literature. In temporal graph detrending, the setting of λ_1 and λ_2 brings similar behavior as that in self-trend filtering. For the total number of iteration (the number to repeat steps (3)-(5)), 3 iterations are often enough for convergence.

To evaluate the efficiency, we summarize the average running time in log scale of DTW, FastDTW, and proposed RobustDTW in

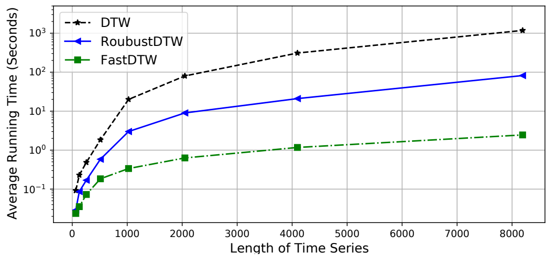

Figure 2: Running time (in log scale) comparison of DTW, FastDTW, and the proposed RobustDTW under different lengths of time series.

Fig. 2 with synthetic datasets. The synthetic TS are single period of a sine wave with random noise and length varies from 2^6 to 2^{13}. The experiment is performed on a PC with 2.3GHz Intel Core i5 CPU and 16GB RAM. It can be observed in Fig. 2 that the computational time of the standard DTW increases rapidly as the length of TS increases. Due to the proposed multi-level framework, our RobustDTW is significantly efficient than the standard DTW, especially for long time series. Although RobustDTW is slower than FastDTW, the speed of RobustDTW is reasonable for most time series application and we will demonstrate later that the proposed RobustDTW is more robust than FastDTW and exhibits much better performances in both outlier time series detection and periodicity detection.

4.2 Outlier Time Series Detection

The purpose of outlier TS detection is to find one or more abnormal TS compare to other peers [6]. In this experiment, we test the the popular local outlier factor (LOF) algorithm [8, 22, 31], which finds outliers by measuring the local deviation of a given data point with respect to its neighbors, with different distance measures for outlier time series detection. We collected two real-world multivariate TS datasets from a top cloud computing company which are the measurement of the API response time (RT) and network speed (NetSpd) of two clusters, respectively. Specifically, The dataset "RT" contains 294 TS and 1 of them are outliers; and the "NetSpd" dataset contains 486 TS and 6 of them are outliers. We apply the LocalOutlierFactor function [23] from scikit learn package to conduct LOF step for anomaly detection. For all different distance measures the hyper-parameters of LOF are tuned to get best performance, where the number of neighbors is set to be 30 and contamination is 0.02.

Table 1 compares the AUC scores of LOF algorithm with different distance measures, including Euclidean Distance (ED), standard DTW, FastDTW, and our RobustDTW, under different noise conditions, i.e, raw datasets, datasets with injected dips, datasets with injected spike, and datasets with injected both spike and dips. The standard DTW performs the second best, while FastDTW achieves fast computation at the cost of worse accuracy. In contrast, our RobustDTW achieves the best performance consistently.

4.3 Time Series Periodicity Detection

Many TS are characterized by repeating cycles, or periodicity. Periodicity detection aims at discovering the repeating patterns in TS. The periodicity detection is a nontrivial task [30, 34] given complicated TS, e.g., non-stationary, sudden trend change, noise and outliers, as shown in Fig. 3. Here we focus on dominant periodicity detection, where domain experts generally have the prior knowledge about the period length (T) if the TS are periodic.

Table 1: Comparison of AUC scores of outlier detection via LOF with different distance measures and noise conditions.

Dataset	Noise level	ED	DTW	FastDTW	**RobustDTW**
RT	raw data	0.887	0.986	0.949	**0.997**
	+ dips	0.811	0.965	0.918	**0.996**
	+ spikes	0.362	0.939	0.775	**0.972**
	+ spikes & dips	0.317	0.580	0.655	**0.969**
NetSpd	raw data	0.674	0.982	0.917	**0.988**
	+ dips	0.635	0.849	0.649	**0.982**
	+ spikes	0.642	0.915	0.874	**0.975**
	+ spikes & dips	0.508	0.627	0.616	**0.938**

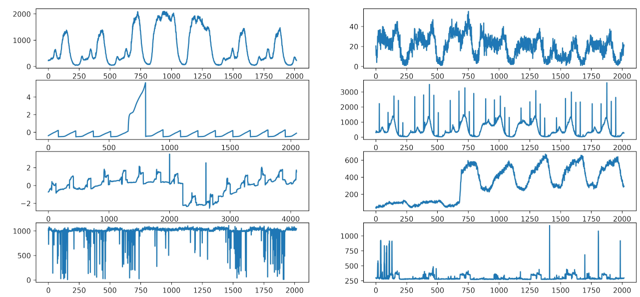

Figure 3: Real-world TS examples from cloud monitoring with complex patterns for periodicity detection. The TS in top three rows are periodic while the bottom ones are without periodicity. These time series have challenging noise, outliers, and/or trend changes.

RobustDTW-based Slicing Algorithm:
In this part we propose a slicing algorithm to determine whether a given TS has a prior period of T or not. This algorithm first detrends and normalizes input TS data, removes outliers, and then slices the TS into multiple segments with equal length of T. Next we calculate distance between adjacent segments using RobustDTW. For detrending, we apply the robust trend filtering [33] to remove the global trend which is useful for non-stationary TS. For normalization, all TS are normalized by subtracting median and then dividing by the biweight scale. Note that median and biweight scale are more robust to noise and outliers compared to mean and standard deviation. The biweight scale [4] is calculated as

$$\xi_{biscl} = \sqrt{n \sum_{|u_i|<1}(x_i - M)^2(1-u_i^2)^4} \Big/ \sqrt{\sum_{|u_i|<1}(1-u_i^2)(1-5u_i^2)},$$

where x is the input data, M is the sample median, $u_i = (x_i - M)/(c * MAD)$ with c as the tuning constant typically set to be 9.0, and MAD is the median absolute deviation. After that, to further reduce the influence of outliers (such as the segment deviation due to black Friday), we filter out the outlier distances as determined by the "1.5×IQR rule" (interquartile range). We then calculate the mean value of the remaining distances which is sum of squared

Table 2: Comparison of periodicity detection methods.

Methods		Precision	Recall	F1
ACF		0.960	0.701	0.810
AUTOPERIOD [30]		**0.980**	0.715	0.827
RobustPeriod [34]		0.920	0.902	0.911
Slicing	ED	0.919	0.800	0.855
	DTW	0.911	0.930	0.920
	FastDTW	0.873	0.970	0.919
	RobustDTW	0.951	**0.980**	**0.966**

differences between values that aligned on both time series, as a measure of periodicity. And a TS is considered has a period of T if the estimated mean value is less than a user-specified threshold.

Periodicity Detection Comparison:
We collect 200 service monitoring time series from a top cloud computing company. Half of these TS are with daily periodicity. These TS data exhibit various shapes with challenging noise, outliers, and trend changes as shown in Fig. 3. For model evaluation, we test our algorithm under different distance measures, including Euclidean distance, standard DTW, FastDTW and our RobustDTW. We also compare our method with some commonly used periodicity detection algorihtms such as ACF, as well as the state-of-the-art AUTOPERIOD [24, 30] and RobustPeriod [34] algorithms. The performance metrics including precision, recall and F1, are summarized in Table 2, and the best performance is highlighted in bold. It can be observed that among all compared methods, our method using slicing and RobustDTW achieves the best F1 score. We attribute the advantages of our periodicity detection using RobustDTW to: 1) graph detrending can reduce the pairwise distance significantly when two TS have similar shape; 2) filtering neighboring distance can exclude large irregular shapes within one period; 3) robust normalization using median and biweight scale is insensitive to outliers and help parameter tuning easier.

5 CONCLUSION

In this paper, we propose a novel RobustDTW measure which estimates the time warping function and the trend simultaneously by an alternating approach and further accelerate it in a multi-level framework. Compared with existing DTW and its variants, it is fast and robust to noise and outliers. We demonstrate the successful applications of our RobustDTW algorithm in outlier time series detection and periodicity detection.

REFERENCES

[1] Theodore Alexandrov, Silvia Bianconcini, Estela Bee Dagum, Peter Maass, and Tucker S McElroy. 2012. A review of some modern approaches to the problem of trend extraction. *Econometric Reviews* 31, 6 (2012), 593–624.

[2] Daniele Angelosante and Georgios B Giannakis. 2011. Sparse graphical modeling of piecewise-stationary time series. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1960–1963.

[3] Sylvio Barbon, Rodrigo Capobianco Guido, Lucimar Sasso Vieira, Everthon Silva Fonseca, Fabrício Lopes Sanchez, Paulo Rogério Scalassara, Carlos Dias Maciel, José Carlos Pereira, and Shi-Huang Chen. 2009. Wavelet-based dynamic time warping. *J. Comput. Appl. Math.* 227, 2 (2009), 271 – 287. Special Issue on Emergent Applications of Fractals and Wavelets in Biology and Biomedicine.

[4] Timothy C Beers, Kevin Flynn, and Karl Gebhardt. 1990. Measures of location and scale for velocities in clusters of galaxies-A robust approach. *The Astronomical Journal* 100 (1990), 32–46.

[5] Seif-Eddine Benkabou, Khalid Benabdeslem, and Bruno Canitia. 2018. Unsupervised outlier detection for time series by entropy and dynamic time warping. *Knowledge and Information Systems* 54, 2 (2018), 463–486.

[6] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. 2021. A Review on outlier/Anomaly Detection in Time Series Data. *ACM Computing Surveys (CSUR)* 54, 3 (2021), 1–33.

[7] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends® in Machine Learning* 3, 1 (2011), 1–122.

[8] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.

[9] Xiaobin Chang, Frederick Tung, and Greg Mori. 2021. Learning discriminative prototypes with dynamic time warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8395–8404.

[10] Marco Cuturi and Mathieu Blondel. 2017. Soft-DTW: a differentiable loss function for time-series. *arXiv preprint arXiv:1703.01541* (2017).

[11] Dave Deriso and Stephen P. Boyd. 2019. A General Optimization Framework for Dynamic Time Warping. *CoRR* abs/1905.12893 (2019). arXiv:1905.12893

[12] D. M. Diab, B. AsSadhan, H. Binsalleeh, S. Lambotharan, K. G. Kyriakopoulos, and I. Ghafir. 2019. Anomaly Detection Using Dynamic Time Warping. In *2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*. 193–198.

[13] Mohamed G Elfeky, Walid G Aref, and Ahmed K Elmagarmid. 2005. WARP: time warping for periodicity detection. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*. 8–pp.

[14] Philippe Esling and Carlos Agon. 2012. Time-series data mining. *ACM Computing Surveys (CSUR)* 45, 1 (2012), 1–34.

[15] Renmin Han, Yu Li, Xin Gao, and Sheng Wang. 2018. An accurate and rapid continuous wavelet dynamic time warping algorithm for end-to-end mapping in ultra-long nanopore sequencing. *Bioinformatics* 34, 17 (2018), i722–i731.

[16] Siteng Huang, Donglin Wang, Xuehan Wu, and Ao Tang. 2019. Dsanet: Dual self-attention network for multivariate time series forecasting. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. 2129–2132.

[17] Young-Seon Jeong, Myong K. Jeong, and Olufemi A. Omitaomu. 2011. Weighted dynamic time warping for time series classification. *Pattern Recognition* 44, 9 (2011), 2231 – 2240. Computer Analysis of Images and Patterns.

[18] Sylvio Barbon Junior, Rodrigo Capobianco Guido, Shi-Huang Chen, Lucimar Sasso Vieira, and Fabricio Lopes Sanchez. 2007. Improved dynamic time warping based on the discrete wavelet transform. In *Ninth IEEE International Symposium on Multimedia Workshops (ISMW 2007)*. 256–263.

[19] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and information systems* 7, 3 (2005), 358–386.

[20] Eamonn J Keogh and Michael J Pazzani. 2001. Derivative dynamic time warping. In *Proceedings of the 2001 SIAM international conference on data mining*. 1–11.

[21] Huanhuan Li, Jingxian Liu, Zaili Yang, Ryan Wen Liu, Kefeng Wu, and Yuan Wan. 2020. Adaptively constrained dynamic time warping for time series classification and clustering. *Information Sciences* 534 (2020), 97 – 116.

[22] Shengyuan Liu, Yuxuan Zhao, Zhenzhi Lin, Yilu Liu, Yi Ding, Li Yang, and Shimin Yi. 2020. Data-driven event detection of power systems based on unequal-interval reduction of PMU data and local outlier factor. *IEEE Transactions on Smart Grid* 11, 2 (2020), 1630–1643.

[23] LocalOutlierFactor. 2021. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.LocalOutlierFactor.html. [Online; accessed May-2021].

[24] Theophano Mitsa. 2010. *Temporal Data Mining* (1st ed.). Chapman and Hall/CRC, Boca Raton, FL.

[25] 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.

[26] Thomas Prätzlich, Jonathan Driedger, and Meinard Müller. 2016. Memory-restricted multiscale dynamic time warping. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 569–573.

[27] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. 2012. Searching and Mining Trillions of Time Series Subsequences under Dynamic Time Warping. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Beijing, China) *(KDD '12)*. New York, USA, 262–270.

[28] Hiroaki Sakoe and Seibi Chiba. 1971. A Dynamic Programming Approach to Continuous Speech Recognition. In *Proceedings of the Seventh International Congress on Acoustics, Budapest*, Vol. 3. Akadémiai Kiadó, Budapest, 65–69.

[29] Stan Salvador and Philip Chan. 2007. FastDTW: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11, 5 (2007), 561–580.

[30] Michail Vlachos, Philip Yu, and Vittorio Castelli. 2005. On periodicity detection and structural periodic similarity. In *Proceedings of the 2005 SIAM international conference on data mining*. 449–460.

[31] Wei Wang and Peizhong Lu. 2011. An efficient switching median filter based on local outlier factor. *IEEE Signal Processing Letters* 18, 10 (2011), 551–554.

[32] Yu-Xiang Wang, James Sharpnack, Alexander J Smola, and Ryan J Tibshirani. 2016. Trend filtering on graphs. *The Journal of Machine Learning Research* 17, 1 (2016), 3651–3691.

[33] Qingsong Wen, Jingkun Gao, Xiaomin Song, Liang Sun, and Jian Tan. 2019. RobustTrend: a Huber loss with a combined first and second order difference regularization for time series trend filtering. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*. AAAI Press, 3856–3862.

[34] Qingsong Wen, Kai He, Liang Sun, Yingying Zhang, Min Ke, and Huan Xu. 2021. RobustPeriod: Time-Frequency Mining for Robust Multiple Periodicity Detection. In *ACM International Conference on Management of Data (SIGMOD)*.

[35] Renjie Wu and Eamonn J Keogh. 2020. FastDTW is approximate and Generally Slower than the Algorithm it Approximates. *arXiv preprint arXiv:2003.11246* (2020).

[36] Jidong Yuan, Qianhong Lin, Wei Zhang, and Zhihai Wang. 2019. Locally slope-based dynamic time warping for time series classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*. 1713–1722.

[37] Yaodong Zhang, Kiarash Adl, and James Glass. 2012. Fast spoken query detection using lower-bound dynamic time warping on graphical processing units. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 5173–5176.