IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 2, 2021, accepted March 7, 2021, date of publication March 22, 2021, date of current version March 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3067833

Efficient Time Series Clustering by Minimizing Dynamic Time Warping Utilization

BORUI CAI[1], GUANGYAN HUANG[1], (Member, IEEE), NAJMEH SAMADIANI[1], GUANGHUI LI[2], AND CHI-HUNG CHI[3]

[1]School of Information Technology, Deakin University, Burwood, VIC 3125, Australia
[2]Department of Computer Science and Technology, Jiangnan University, Wuxi 214122, China
[3]Data61, Sandy Bay, TAS 7005, Australia

Corresponding author: Guangyan Huang (guangyan.huang@deakin.edu.au)

This work was supported in part by the Australia Research Council (ARC) DECRA Project under Grant DE140100387, and in part by the Discovery Project under Grant DP190100587.

ABSTRACT Dynamic Time Warping (DTW) is a widely used distance measurement in time series clustering. DTW distance is invariant to time series phase perturbations but has a quadratic complexity. An effective acceleration method must reduce the DTW utilization ratio during time series clustering; for example, TADPole uses both upper and lower bounds to prune off a large ratio of expensive DTW calculations. To further reduce the DTW utilization ratio, we find that the linear-complexity L1-norm distance (Manhattan distance) is effective enough when the time series only comprise small phase perturbations. Therefore, we propose a novel time series clustering by Minimizing Dynamic Time Warping Utilization (MiniDTW) algorithm to accelerate time series clustering. In MiniDTW, the dataset is first *greedily* summarized into seed clusters, which comprise time series of small phase perturbations, by L1-norm distance. Then, we develop a new Sparse Symmetric Non-negative Matrix Factorization (SSNMF) algorithm, which factorizes the DTW distance matrix of seed cluster centers, to merge the seed clusters into the final clusters. The experiments on UCR time series datasets demonstrate that MiniDTW, pruning 98.52% of the DTW utilization, is better than the counterpart method, TADPole, which only prunes 75.56% of the DTW utilization; and thus MiniDTW is 10 times faster than TADPole.

INDEX TERMS Time series, density-based clustering, dynamic time warping, L1-norm.

I. INTRODUCTION

Time series is one of the most important data in the modern data-driven society and can be generated from nearly every aspects in the daily life [1]. Time series analysis can benefit pervasive applications in different domains, e.g., financial marketing [2], smart home [3] and autonomous vehicles [4]. Time series clustering is a basic technique for analyzing time series. It can discover the underlying structure of the chaotic/raw datasets without the ground truth labels. This makes it particularly useful for analyzing many unlabeled real-world datasets, such as common pattern discovery [5], information retrieval [6] and outlier detection [7].

Time series distance measurement method is essential for the clustering accuracy, but the precision of simple distance measurements, such as L1-norm (Manhattan distance) and cross correlation are undermined by the widely appeared phase perturbations (e.g., phase shifting, time warping) in time series [8]. Dynamic Time Warping (DTW) [9] is a distance measurement that is robust to time series phase perturbations; however, its quadratic complexity greatly impairs the efficiency of time series clustering. To accelerate time series clustering with DTW distance, some methods reduce the DTW utilization ratio by pruning unnecessary DTW calculations with fast calculated upper/lower bounds of DTW, such as TADPole [5]. Unfortunately, existing methods are hard to prune most DTW calculations (for example, TADPole needs 24.43% DTW calculations after pruning), because it is challenging to define tight lower/upper bounds of DTW distance, which leads to large runtime for the clustering.

To significantly reduce the DTW utilization ratio for the acceleration, we only apply the complex DTW calculation on a summarized time series dataset (rather than the original dataset). This is inspired by the work [6] that achieves interactive time series retrieval by querying a summarized database, rather than the original large dataset. Specifically, we find L1-norm distance is effective to summarize the time series dataset based on three observations. First, L1-norm distance

The associate editor coordinating the review of this manuscript and approving it for publication was Amir Masoud Rahmani.

has a linear complexity and is more efficient than DTW distance. Second, the precision loss of L1-norm distance is limited when time series have small phase perturbations. Third, L1-norm distance is an upper bound of DTW distance, which ensures time series with a small L1-norm distance always have a small DTW distance. To "*greedily*" reduce the DTW utilization ratio, we summarize the dataset into natural-shaped seed clusters with L1-norm distance. Therefore, a seed cluster can group 1) two time series with a small phase perturbation and 2) two time series that comprise a large phase perturbation but can be related to each other by a series of slightly perturbed time series. Then, DTW distance is only used on a small amount of seed cluster centers to merge the seed clusters into final clusters.

In this paper, we propose a novel time series clustering by Minimizing Dynamic Time Warping Utilization (MiniDTW) algorithm to accelerate time series clustering. In MiniDTW, the original dataset is first "*greedily*" summarized as a small amount of natural-shaped seed clusters with the efficient L1-norm distance. The seed clusters are further merged to form the final clusters by a new Sparse Symmetric Non-negative Matrix Factorization (SSNMF) algorithm, which factorizes the DTW distance matrix of seed cluster centers. Comprehensive experiments are conducted using UCR time series datasets [10] to evaluate the proposed MiniDTW algorithm. Therefore, this paper has three contributions:

1) We propose a novel MiniDTW method to speed up time series clustering. MiniDTW minimizes DTW utilization ratio by dataset summarization with the linear-complexity L1-norm distance.

2) We propose an effective SSNMF matrix factorization algorithm, which more accurately merges the seed clusters in MiniDTW than other Non-negative Matrix Factorization based algorithms (i.e., NMF with L1/L2 constraints).

3) We conduct comprehensive experiments to evaluate the performance of the proposed MiniDTW, and the result shows that MiniDTW can effectively avoid 97.90% of DTW utilization and thus is 10 times faster than the counterpart, TADPole, which only prunes 75.73% DTW utilization.

The rest of this paper is organized as follows. In Section 2, we review related works. In Section 3, we introduce the preliminary knowledge and the problem definition. The MiniDTW algorithm is introduced in Section 4 and evaluated in Section 5. Finally, we conclude this paper in section 6.

II. RELATED WORK

Time series clustering groups similar time series into the same cluster, while separating disparate time series into different clusters. There are two essential techniques for effective time series clustering, i.e., the distance measurement of time series, and the clustering algorithm.

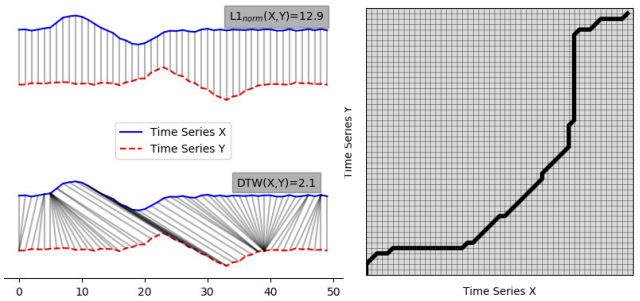
Many time series distance measurements have been proposed in the literature, such as the basic Norm distance, which is normally used as L1-norm distance (Manhattan distance) [11], [12] or L2-norm distance (Euclidean distance) [13], [14]. Norm distance is intuitive and has a linear complexity, but it may face significant precision loss when time series phase perturbation occurs. DTW distance [9] is invariant to time series phase perturbations. DTW finds the distance by searching the optimal continuous warping path between two time series; however, DTW distance has a complexity quadratic to the length of time series.

Many methods are proposed to accelerate DTW distance by reducing the complexity of its search space [15]–[17]. For example, SS-PrunedDTW [16] compresses the search space with an continuously updated upper bound; PDTW [14] reduces the dimension of the search space by compressing time series (with PAA [18]). Meanwhile, other distance measurements are proposed to resolve specific types of phase perturbations. For example, SBD [19] is effective for phase shifting through finding the optimal alignment of two time series by cross correlation; while LCSS [20] is invariant to sampling rate by finding the *longest common subsequence* (LCSS) between the two time series.

Time series clustering is a well-studied field and the clustering algorithms can roughly be categorized into four classes: hierarchical, model-based, partition-based and density-based time series clustering. HSM [21] is an agglomerative time series clustering technique that hierarchically merge clusters. The cluster distance in HSM is calculated with cluster representatives, which are estimated by spectral density. TS3C [22] hierarchically cluster time series, with single-linkage cluster distance, after each time series is mapped to a representation with the centroids of subsequence clusters. Model-based time series clustering normally assumes that time series are generated following specific statistical models. For example, GMM [23] assumes that time series are generated with a mixture of finite Gaussian distributions; while HMM [24], [25] uses a hidden Markov process. Hierarchical and model-based clustering algorithms are relatively complex in terms of calculations, and are usually used to interpret the clustering results.

Partition-based time series clustering partitions the dataset into clusters through minimizing the overall distances of time series to their respective cluster centers. The center of a cluster, e.g., in Kmeans based time series clustering [26], [27], is regarded as the point-wise average of the contained time series; however, such centers may poorly represent the common temporal pattern when phase perturbation appears. Other methods are proposed to find more accurate cluster centers. For example, K-medoids [28] regards the time series that has the least sum of distances to other time series as the center of a cluster; KDBA [29] uses a global averaging technique to generate the centers that adapt to DTW distance; Kshape [19] and KSC [30] discover the centers as eigenvectors by spectral analysis. Compared with density-based methods, these methods do not demand extensive distance calculations; however, their performance is highly affected by the distance measurement adopted.

Density-based clustering finds time series clusters by grouping time series according to their densities. YADING [11] adopts DBSCAN to effectively find clusters containing



(a) $L1_{norm}(X, Y)$ **and** $DTW(X, Y)$**.(b) The optimal warping path of DTW distance.**

FIGURE 1. L1-norm distance and DTW distance of time series *X* and *Y* are demonstrated in (a), and (b) shows the optimal warping path discovered by DTW.

time series that comprise small phase shifting, with the efficient L1-norm distance. TADPole [5] uses DPC [31], another density-based clustering algorithm, with DTW distance to develop an anytime time series clustering algorithm. In TADPole, the DTW utilization ratio is reduced by pruning out-of-bounds DTW distances with the efficient DTW lower/upper bounds. Density-based clustering can find natural-shaped clusters, and in this paper we utilize this characteristic to develop the proposed method.

III. PRELIMINARIES AND PROBLEM DEFINITION

A. L1-NORM DISTANCE AND DTW DISTANCE

Time series is a series of real values, denoted as $X = \{x_1, x_2, x_3, \ldots, x_m\}$, and m is the length. A time series dataset contains n equal length time series ($D = \{X_1, X_2, X_3, \ldots, X_n\}$). L1-norm distance measures the distance of two time series, X and Y, as the overall pair-wise differences as follows:

$$L1_{norm}(X, Y) = \sum_{i=1}^{m} |x_i - y_i|. \tag{1}$$

L1-norm distance has a complexity linear to m, but the precision is low for poorly-aligned time series. As shown in Fig. 1 (a), $L1_{norm}(X, Y)$ is large, despite that X and Y have similar wave shapes, because of the appearance of phase perturbation.

DTW distance is another measurement that is a widely used to accurately measure the distance of time series. DTW distance achieves this by finding the optimal continuous warping path (the best alignment) between X and Y. Specifically, a warping path is denoted as $W = \{w_1, w_1, w_2, \ldots, w_k\}$, where each $w_l = |x_i - y_j|$, and the overall weight of the optimal warping path is used as the DTW distance:

$$DTW(X, Y) = \min_{W} \sum_{l=1}^{k} w_l, w_l \in W. \tag{2}$$

Dynamic programming is applied to ensure the discovery of the optimal warping path has a complexity quadratic to m.

For the example in Fig. 1 (a), DTW distance can properly measure the distance of X and Y regardless the phase perturbation. The optimal warping path between X and Y found by DTW distance is shown in Fig. 1 (b).

B. PROBLEM DEFINITION

DTW distance can effectively measure the distance of time series since it is invariant to phase perturbations; however, its quadratic complexity confines it availability to applications that demand high efficiency. To accelerate time series clustering, we seek to reduce the DTW utilization ratio by dataset summarization using the L1-norm distance, which has the linear complexity. This strategy is based on two observations, where the use of DTW distance is not necessary: 1) time series have small phase perturbations and thus the precision loss of L1-norm distance is not significant, and 2) time series of large phase perturbations can be related by a series of slightly perturbed time series. The following content explains this two observations.

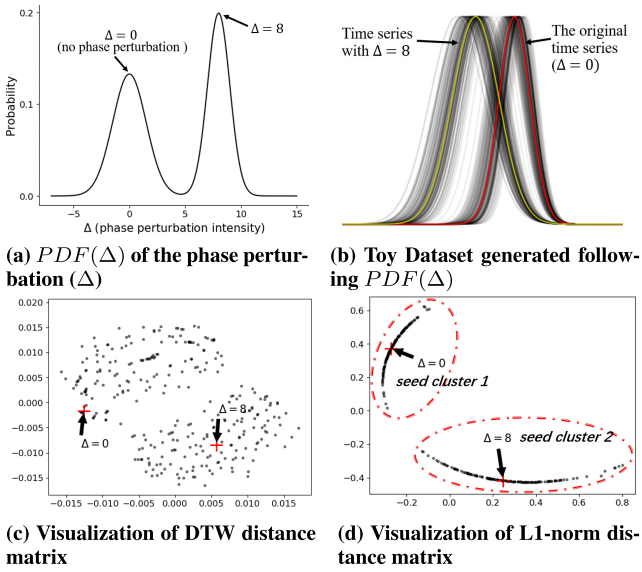
YADING [11] shows that L1-norm distance has a limited precision loss for measuring the distance of time series with small phase shifting. We further extend this finding for general phase perturbation, that is, L1-norm distance has limited precision loss for measuring the distance of two time series with a small phase perturbation, in which scenario DTW distance can be replaced with L1-norm distance.

Lemma 1: $L1_{norm}(X, Y)$ is arbitrarily small, when $\lambda(t)$ (phase perturbation) is arbitrarily small at each t_j ($1 \leq j \leq m$). (Assume X is sampled from $f(t)$ by an equal interval, and Y is sampled from the $f(t + \lambda(t))$).

Proof: An arbitrarily small phase perturbation $\lambda(t)$ means that at each t_j ($1 \leq j \leq m$), $\exists \epsilon_j \; s.t. \; \lambda(t_j) = \epsilon_j$, ϵ_j is arbitrarily small. Therefore, $L1_{norm}(X, Y) = \Sigma_{j=1}^{m} |f(t_j) - f(t_j + \epsilon_j)|$. $\exists \sigma > 0, s.t. \; L1_{norm}(X, Y) \leq \sigma \Sigma_{j=1}^{m} |\epsilon_j| \leq \sigma M |\epsilon|$, where $|\epsilon| = max(|\epsilon_j|)$, $1 \leq j \leq m$. It is proved. □

Lemma 1 is the building block of our method, and it shows that X and Y are neighbours, which have a distance less than the distance threshold, if they have a small phase perturbation. Meanwhile, the correctness of using L1-norm to find neighbours is that the neighbours found by L1-norm distance are always neighbours found by DTW distance, because L1-norm distance is an upper bound of DTW distance [5]. Moreover, X and Y, which have a large phase perturbation, may also be grouped into the same cluster by density-based clustering algorithm with L1-norm distance. Specifically, X and Y are not neighbours to each other by L1-norm distance since the difference of phase perturbation of X and Y (i.e. $\lambda_Y(t) - \lambda_X(t)$) is relatively large. However, if there exists a series of $\{Z_1, Z_2, \ldots, Z_k\}$ in the dataset so that $\{\lambda_{Z_1}(t) - \lambda_X(t) = \epsilon_1, \lambda_{Z_2}(t) - \lambda_{Z_1}(t) = \epsilon_2, \ldots, \lambda_Y(t) - \lambda_{Z_k}(t) = \epsilon_{k+1}\}$, where each ϵ_i is small, X and Y can be related by a neighbour chain ($\{X, Z_1, Z_2, \ldots, Z_k, Y\}$), according to Lemma 1.

Now we consider how to group together time series comprise the same wave shape but with different scales of phase perturbation by density-based clustering algorithm, which rarely uses DTW distance. Without loss of



(a) $PDF(\Delta)$ **of the phase perturbation** (Δ) **(b) Toy Dataset generated following** $PDF(\Delta)$

(c) Visualization of DTW distance matrix **(d) Visualization of L1-norm distance matrix**

FIGURE 2. ***PDF*****(Δ) (bimodal Gaussian distribution) of the intensity of phase perturbation (Δ) is shown in (a); the toy dataset, as shown in (b), contains the same time series having different phase perturbations with random Δs following** ***PDF*****(Δ). The distance matrices obtained by applying DTW distance and L1-norm distance are shown in (c) and (d), respectively.**

generality, we formulate the distribution of the phase perturbation as a bimodal Gaussian distribution, $PDF(\Delta)$, which denotes the probability that a phase perturbation with an intensity $\lambda(t, \Delta)$ appears in the time series. $PDF(\Delta)$ has two peaks ($\Delta = 0$ and $\Delta = 8$) that represent the peak probabilities (as shown in Fig. 2 (a)). We create a toy dataset that contains 300 time series, which have the simple phase perturbation pattern ($f(t + \lambda(t, \Delta))$) following the bimodal Gaussian distribution, as shown in Fig. 2 (b).

We separately apply DTW distance and L1-norm distance on the toy dataset, and visualize the respective distance measurements by Multidimensional Scaling (MDS) [32], as shown in Fig. 2 (c-d). The result of DTW distance clearly exhibits its invariance to phase perturbations (Fig. 2 (c)) since all the obtained DTW distances are small, i.e., the maximum DTW distance is only 0.03. Differently, the visualization of the L1-norm distance matrix appears as two long thin arcs as shown in Fig. 2 (d), with a much larger maximum L1-norm distance, i.e., 1.33. The L1-norm distance matrix becomes two long thin arcs because, for each time series, there is only a limited amount of time series having small L1-norm distances with it (thin), but have many time series that lead larger L1-norm distances when the difference of Δ become larger (long). In addition, the time series that have Δs of the two peaks in the $PDF(\Delta)$, i.e. $\Delta = 0$ and $\Delta = 8$, also have peak densities in the two arcs, respectively. Therefore, it is intuitive to adopt a two-step approach to group these time series into a cluster. First, the toy dataset is summarized as two seed clusters, which contain respective time series forming the two arcs in Fig. 2 (d), based on density calculated with L1-norm distance. Second, the two seed clusters are merged into the final cluster by their small DTW distance of centers (Fig. 2 (c)). In this way, the efficiency to cluster these time series is greatly improved since 44,849 ($\frac{300\times299}{2} - 1$) DTW calculations are avoided.

IV. THE PROPOSED METHOD

We propose a novel time series clustering algorithm, MiniDTW, to minimize the DTW utilization ratio by dataset summarization with L1-norm distance. MiniDTW includes the following two steps:

1) Summarize the dataset with L1-norm distance as natural-shaped seed clusters, i.e. time series comprise small phase perturbations.

2) Discover the final clusters on the summarized dataset by merging seed clusters with the DTW distances among their centers.

In the following contents we detail the two steps of MiniDTW.

A. DATASET SUMMARIZATION WITH L1-NORM DISTANCE

To efficiently summarize time series comprising small phase perturbations as seed clusters with L1-norm distance, we take the advantage of density-based clustering algorithms. For time series X_i, we define its density (ρ_i) with a distance threshold (d_c) as follows:

$$\rho_i = \sum_{j:L1_{norm}(X_i,X_j)\le d_c} \left(1 - \frac{L1_{norm}(X_i, X_j)}{d_c}\right). \tag{3}$$

The density in Eq. (3) emphasizes the weight of time series with smaller phase perturbations, which have smaller distances.

Lemma 2: $L1_{norm}(X, Z) > L1_{norm}(X, Y)$, w.r.t $\lambda_Y(t_j) \le \lambda_Z(t_j)$ $(1 \le j \le m)$ and $\lambda_Y(t)$ and $\lambda_Y(t)$ are arbitrarily small at each t_j. (Assume X, Y and Z are respectively sampled from $f(t), f(t + \lambda_Y(t))$ and $f(t + \lambda_Z(t))$.)

Proof: Arbitrarily small phase perturbations $\lambda_Y(t)$ and $\lambda_Z(t)$ means that at each t_j $(1 \le j \le m)$, $\exists \epsilon_{Yj}, \epsilon_{Zj}, s.t.\ \lambda(t_{Yj}) = \epsilon_{Yj}, \lambda(t_{Zj}) = \epsilon_{Zj}$ and $\epsilon_{Yj} \le \epsilon_{Zj}$ (since $\lambda_Y(t_j) \le \lambda_Z(t_j)$), when $\epsilon_{Yj}, \epsilon_{Zj}$ are arbitrarily small. $\Delta_{l1} = L1_{norm}(X, Z) - L1_{norm}(X, Y) = \Sigma_{j=1}^{m}|x_j - z_j| - \Sigma_{j=1}^{m}|x_j - y_j|$. Considering the values of x_j, y_j and z_j at t_j, $\Delta_{l1} = \Sigma_{j\in G_1}(2x_j - y_j - z_j) + \Sigma_{j\in G_2}(y_j + z_j - 2x_j) + \Sigma_{j\in G_3}(y_j - z_j) + \Sigma_{j\in G_4}(z_j - y_j)$. $\forall j \in G_1 \cup G_2$, there is $f'(t_j) \approx 0$ (stationary point), which means $|G_1 \cup G_2| \ll m$; and $(2x_j - y_j - z_j) = (f(t_j) - f(t_j + \epsilon_{Yj})) + (f(t_j) - f(t_j + \epsilon_{Yj}))$ is arbitrarily small since it is proportional to ϵ_{Yj} and ϵ_{Zj} (proved in Lemma 1). Therefore, $\Delta_{l1} \approx \Sigma_{j\in G_3}(y_j - z_j) + \Sigma_{j\in G_4}(z_j - y_j)$. $\forall j \in G_3$, there is $x_j \ge z_j, x_j \ge y_j$, that means $f'(t) \le 0$ when $t_j \le t \le t_j + \epsilon_{Zj}$ and $y_j - z_j = f(t_j + \epsilon_{Yj}) - f(t_j + \epsilon_{ZJ}) \ge 0$. Similarly, $\forall j \in G_4$, i.e. $x_j \le z_j, x_j \le y_j$, there is $z_j - y_j \ge 0$. Thus $\Delta_{l1} \ge 0$. It is proved. □

We define the center of a seed cluster as the time series with a local density peak to approximate the relative peak of $PDF(\Delta)$ as follows:

Definition 1: X_i is the center of a seed cluster if $\rho_i > \rho_j, \forall j \in \{j : L1_{norm}(X_i, X_j) < d_c\}$.

We borrow the heuristic of DPC [31] to group time series with small phase perturbations into seed clusters, with the centers having the largest local densities (see Fig. 2 (d)). For X_i, we find a time series $n_i = X_j$ as follows:

$$n_i = \underset{\substack{X_j, s.t. \rho_i < \rho_j, \\ L1_{norm}(X_i, X_j) < d_c}}{\arg\min} L1_{norm}(X_i, X_j). \tag{4}$$

Apparently, X_i is the center of a seed cluster if n_i does not exist according to Definition 1. Then, the dataset is summarized as seed clusters in a two-step process: 1) assign an unique seed cluster label to each center; 2) spread the seed cluster label from the centers to time series that have lower densities, i.e., X_i acquires seed cluster label from its n_i.

B. MERGE THE TIME SERIES SEED CLUSTERS

After the original dataset is summarized as seed clusters, we further merge seed clusters into the final clusters, based on the DTW distances of their centers. Specifically, we merge the centers of seed clusters into final clusters, and then assign the non-center time series to the same clusters as their centers.

Assume we merge τ seed clusters into K clusters by DTW distances of seed cluster centers. We propose a Sparse Symmetric Non-negative Matrix Factorization (SSNMF) algorithm to merge seed cluster centers, due to the non-negative and symmetric properties of the DTW distance matrix ($M \in R^{\tau \times \tau}$). SSNMF is able to discover the latent structure of the relationships among the seed cluster centers. Specifically, in SSNMF, M is factorized as the multiplication of two matrices, i.e., $H \in R^{\tau \times K}$ and $S \in R^{K \times K}$ ($S = S^T$), which have non-negative entries, as follows:

$$M = HST^T. \tag{5}$$

H is the feature matrix that represents the latent structure of data derived from M, and it also implies the center assignment, i.e., the ith center belongs to the jth cluster if $\arg\max_{1 \le k \le K} h_{ik} = j$. SSNMF imposes a $L_{\frac{1}{2}}$ sparse constraint for H to ensure the assignment weights of each center to the K clusters are exclusive, i.e., each center only has seldom non-zero weights. We choose $L_{\frac{1}{2}}$ norm since it is differentiable and can produce sparser solutions than the L_1 norm regularization [33]. Therefore, the cost function of SSNMF is defined as follows:

$$\ell = \min_{H \ge 0, S \ge 0} \|M - HSH^T\|_F^2 + \eta \|H\|_{\frac{1}{2}}, \tag{6}$$

where $\|*\|_F$ is the Frobenius norm, η is the weight of sparsity and $\|H\|_{\frac{1}{2}}$ is the $L_{\frac{1}{2}}$ sparse constraint of H defined as follows:

$$\|H\|_{\frac{1}{2}} = \sum_i \sum_j h_{ij}^{\frac{1}{2}}. \tag{7}$$

The cost function in Eq. (6) is non-convex, and we obtain local minima by an iterative multiplicative updating process akin to [34]. Let $\Lambda \in R^{\tau \times K}$ and $\Gamma \in R^{K \times K}$ be the Lagrange multipliers subject to $\lambda_{ij} \ge 0$ and $\gamma_{ij} \ge 0$, respectively.

Algorithm 1 Merge Seed Clusters

Input: Seed clusters *seeds*, cluster number K, η,

1: Calculate DTW distance matrix (M) for the centers of *seeds*.
2: Initialize H and S with random positive values.
3: **while** not converge **do**
4: Update H:
5: $H = H \odot \frac{MHS}{HSH^THS + \frac{1}{8}\eta A}$
6: Update S:
7: $S = S \odot \frac{H^TMH}{H^THSH^TH}$
8: **end while**
9: Initialize *clusters* as K empty sets
10: **for** $1 \le i \le c$ **do**
11: $j = \arg\max_{1 \le k \le K} h_{ik}$
12: $clusters_j = clusters_j \cup seeds_i$.
13: **end for**

Output: *clusters*

Combining Lagrange multipliers with the cost function in Eq. (6), we have the Lagrangian problem as follows:

$$\ell\ell = \mathrm{Tr}(MM) - 2\,\mathrm{Tr}(MHSH^T) + \mathrm{Tr}(HSH^THSH^T) + \eta\|H\|_{\frac{1}{2}} + \mathrm{Tr}(\Lambda H^T) + \mathrm{Tr}(\Gamma S). \tag{8}$$

The partial derivative of $\ell\ell$ with respect to H is given by:

$$\frac{\partial \ell\ell}{\partial H} = -4MHS + 4HSH^THS + \frac{1}{2}\eta A, \tag{9}$$

where $A \in R^{\tau \times K}$ and $a_{ij} = h_{ij}^{-\frac{1}{2}}$. The partial derivative of $\ell\ell$ with respect to H is given by:

$$\frac{\partial \ell\ell}{\partial S} = -2H^TMH + 2H^THSH^TH. \tag{10}$$

Using the Karush-Kuhn-Tucker conditions that $\lambda_{ij}h_{ij} = 0$ and $\gamma_{ij}s_{ij} = 0$, we obtain the final multiplicative updating rule for H and S, respectively, as follows:

$$H = H \odot \frac{MHS}{HSH^THS + \frac{1}{8}\eta A}, \tag{11}$$

$$S = S \odot \frac{H^TMH}{H^THSH^TH}. \tag{12}$$

We initialize H and S with random positive values, and the optimal H and S are obtained after the updates of H (Eq. (11)) and S (Eq. (12)) reach convergence.

The pseudo code of seed cluster merging is demonstrated in Algorithm 1. At line 1, the DTW distance matrix of seed cluster centers (M) is obtained. At lines 2-8, M is decomposed into H and S by the iterative updating process. The final clusters are obtained at lines 9-13.

C. TIME COMPLEXITY

The computation of L1-norm distance matrix is $O(n^2 * m)$, where n is the size of dataset and m is the length of time series. The complexity to find time series seed clusters by

DPC is $O(\frac{1}{2}n^2 + n(\phi + \log n + 1))$, where $\phi \ll n$ is the average number of neighbours. Thus the overall complexity to find seed clusters is $O(n^2(m+\frac{1}{2})+n(\phi+\log n+1))$. In the seed clusters merging phase, the complexity to calculate distance matrix with DTW among dense clusters is $O(\tau^2 m^2)$, where τ is the number of seed clusters. The iterative updating process to obtain optimal H and S requires a complexity of $O(l\tau^2K^4)$, where l is the number of iterations and τ and K are normally far smaller than n. Therefore, the overall complexity of MiniDTW is approximately $O(n^2 * m)$.

V. EVALUATION

In this section, we design the following experiments to compare the runtime efficiency and clustering accuracy of the proposed MiniDTW algorithm with the counterpart methods. All the experiments are implemented with Python 3.2, and run on a Linux platform with 2.6G CPU and 132G RAM.

A. EXPERIMENT SETUP

We use all the datasets in the UCR time series achieve [10] for the evaluation. These datasets have different sizes (ranging from 40 to 16637) and different lengths of time series (ranging from 24 to 2709). Each dataset contains a training set and a testing set (with labels), and we use both for clustering.

Since MiniDTW is proposed to accelerate time series clustering by reducing the DTW utilization ratio, TADPole [5] is the counterpart method most related to ours because it aims at accelerating time series clustering by pruning a fraction of DTW distance use based on faster DTW upper/lower (L1-norm/LB_Keogh [35]) bounds. SS-PrunedDTW [16] is another method that directly accelerate DTW distance. KDBA [29] uses Kmeans for clustering and designs a DTW-adaptable center discovery method. Two state-of-the-art time series clustering algorithms, i.e., Kshape [19] and TS3C [22], are also included in the evaluation. We brief the counterpart methods as follows:

- TADPole uses density-based clustering algorithm (DPC [31]) with DTW distance measurement for clustering, and it accelerates the clustering process by pruning a significant proportion of DTW use with fast calculated lower/upper bounds of DTW.

SS-PrunedDTW

- accelerates the DTW distance measurement by pruning outbound search operations with an upper bound. Single-linkage hierarchical clustering is used to cluster time series with their efficiently calculated DTW distances.
- KDBA extends the conventional Kmeans clustering algorithm to support the use of DTW distance measurement. KDBA adopts a DTW-adaptive center discovery method to ensure the non-center time series in clusters have small DTW distances to the respective centers.
- Kshape is proposed to cluster time series invariant to time series phase shifting. Kshape measures the distance of two time series by a shape-based measurement (SBD). After time series that have phase shifting are re-aligned by SBD, Kshape discovers clusters by minimizing the distances of the re-aligned time series to the cluster centers.
- TS3C clusters time series by the temporal patterns of time series segments. TS3C first finds segment clusters for each time series, which contains time series segments of different lengths. Then, time series are represented as the segment clusters to discover the final clusters by hierarchical clustering.

We apply the above clustering algorithms on all the UCR time series datasets, and the clustering accuracy is measured by Rand Index (RI) following [19], [22]. RI penalizes false positive and false negative clustering results and is defined as follows:

$$RI = \frac{TP + TN}{TP + TN + FP + FN}, \tag{13}$$

where TP is the number of time series pairs that have the same ground truth label and are correctly clustered in the same cluster; TN is the number of pairs that have different labels and are correctly separated into the different clusters; FP means the number of pairs that have different labels but are wrongly clustered in the same cluster; FN is the number of time series pairs that have the same label but are wrongly separated into different clusters. Note that $RI \in (0, 1]$, and a higher RI means the better clustering accuracy.

B. ACCURACY ANALYSIS

Though this paper focuses on improving the efficiency of time series clustering, we first show that the acceleration of MiniDTW does not necessarily sacrifice clustering accuracy by comparing with TADPole, SS-PrunedDTW, KDBA, Kshape and TS3C. The warping window for DTW, which is used by MiniDTW, TADPole, SS-PrunedDTW and KDBA, is fixed as 5% in all the algorithms. Except SS-PrunedDTW, Kshape and KDBA that only require the number of clusters, we use grid search to find optimal parameters for TADPole and MiniDTW. For TADPole, the optimal d_c is obtained by a grid search ranging from 0.01 to 1 (multiplied with the largest DTW distance in the dataset), with grid size as 0.01. For MiniDTW, the optimal d_c is obtained the same as TADPole, and the optimal η (the weight of sparsity) is searched among $\{0.1, 1, 10, 100, 1000\}$. We directly use the published accuracy results of TS3C [22].

1) OVERALL CLUSTERING ACCURACY

We first compare MiniDTW with algorithms that use DTW distance for clustering, i.e., TADPole, SS-PrunedDTW and KDBA, in terms of clustering accuracy. KDBA is used as the baseline and the results are shown in Table 1. Apparently, MiniDTW and TADPole, which are density-based time series clustering algorithms, perform better than KDBA (the partition-based)on most datasets, while SS-PrunedDTW (hierarchical) achieves lower accuracy than KDBA in more than half datasets. Moreover, the average RI of MiniDTW is

TABLE 1. Comparison of clustering accuracy between MiniDTW and clustering algorithms that use DTW distance (TADPole, SS-PrunedDTW and KDBA), with the baseline as KDBA. >, < and = indicate the number of datasets in which the clustering results are better, worse or equal to KDBA, respectively.

Algorithm	>	=	<	Average RI
MiniDTW	75	2	7	0.7685
TADPole	65	0	19	0.7322
SS-PrunedDTW	33	0	51	0.6479

TABLE 2. Comparison of clustering accuracy between MiniDTW and clustering algorithms that do not use DTW distance (Kshape, TS3C and Kmeans), with the baseline as Kmeans. >, < and = indicate the number of datasets in which the clustering results are better, worse or equal to Kmeans, respectively.

Algorithm	>	=	<	Average RI
MiniDTW	78	2	4	0.7685
Kshape	42	4	38	0.6899
TS3C	26	1	57	0.6642

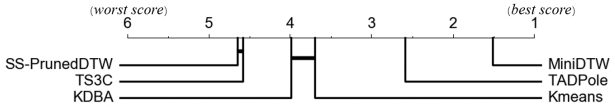


FIGURE 3. Statistical comparisons of MiniDTW with counterpart clustering algorithms on the datasets, and a lower rank score indicates the better performance.

0.7685, which improves the accuracy of TADPole (average RI is 0.7322) and SS-PrunedDTW (average RI is 0.6479) by around 5% and 19%, respectively.

MiniDTW is further compared with Kshape, TS3C and Kmeans, which do not use DTW distance measurement for time series clustering. We use Kmeans as the baseline for comparison, and the results are shown in Table 2. Specifically, MiniDTW wins or equals to Kmeans in most (80 out of 84) datasets, and achieves the highest average RI (0.7685). Kshape achieves slightly better clustering results than Kmeans, while TS3C performs the worst among the four algorithms and achieve accuracy lower than Kmeans in 57 datasets (out of 84). MiniDTW improves the accuracy of Kshape by 11% and that of TS3C by 16%.

The statistical comparison of MiniDTW and the counterpart algorithms is shown in Fig. 3. In general, the density-based clustering methods using DTW distance measurements, i.e. MiniDTW and TADPole, achieve better clustering accuracy than other algorithms; while SS-PrunedDTW, which uses the same DTW distance values (but a fast calculated version) as MiniDTW and TADPole, performs the worst due to the use of hierarchical clustering. MiniDTW achieves the lowest rank score, that is, it statistically achieves the best clustering accuracy. Meanwhile, the hypothesis that these algorithms are significantly different is rejected by Holm-Bonferroni method, and the horizontal lines connect algorithms that are not significantly different. Kmeans and KDBA, both adopt the same strategy to group clusters and are not significantly different on these datasets. Similarly, SS-PrunedDTW and TS3C (both are hierarchical clustering) are not significantly different. Although the paper

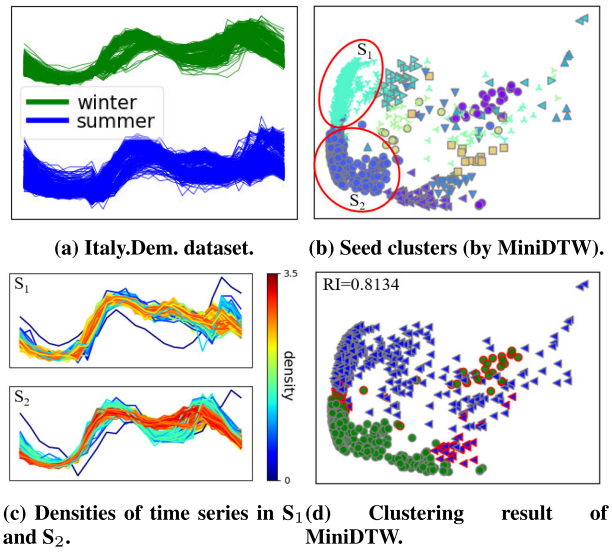


FIGURE 4. An example of clustering a real-world time series dataset using MiniDTW. The Italy.Dem. dataset contains time series of two classes as shown in (a). The seed clusters discovered by MiniDTW are shown in (b), and the time series (with their densities) of the two largest seed clusters are shown in (c). The final clustering result is shown in (d).

focuses on speeding up the clustering of time series, the above results show that the clustering accuracy of MiniDTW is at least comparable with state-of-art time series clustering algorithms.

2) CASE STUDY

We use the Italy.Dem. dataset as a real-world example to show how MiniDTW effectively clusters time series with phase perturbations. In contrast to MiniDTW that approximates the toy bimodal distribution with two seed clusters in Fig. 2, the complex distribution of ItalyPowerDemand dataset is approximated as a combination of multiple unimodal distributions (seed clusters) shown in Fig. 4. Italy.Dem. contains two classes of time series that represent the Italy power consumption in winter and summer, respectively, as shown in Fig. 4 (a). MiniDTW first discovers several seed clusters, visualized by Multidimensional Scaling [32], as shown in Fig. 4 (b). As discussed in Section 4.1, we use the density (defined by L1-norm) of time series to approximate the *PDF* of phase perturbation. Fig. 4 (c) shows two seed cluster examples (S_1 and S_2) effectively group time series with small phase perturbations, and each seed cluster uses the density distribution to approximate a unimodal distribution. For example, the center time series in S_1 has the largest density, while the densities of the rest time series gradually decrease with larger phase perturbations (to the center time series). With these fine seed clusters, the final merged clustering results (Fig. 4 (d)) show that MiniDTW correctly clusters most time series and achieves the highest accuracy (RI = 0.8134) among the compared methods.

3) VARIANTS OF SEED CLUSTER MERGING ALGORITHMS

To understand the effectiveness of the proposed SSNMF algorithm in MiniDTW for seed cluster merging, we replace

TABLE 3. The statistics of datasets [10] used for efficiency analysis.

No.	Dataset	Size	Length	Classes
1	Car	120	577	4
2	Diat.Red.	322	345	4
3	Prox.TW	605	80	6
4	Italy.Dem.	1096	24	2
5	CinC_ECG.	1420	1639	4
6	Ins.Sound	2200	256	11

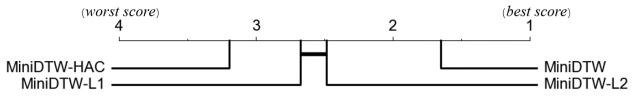


FIGURE 5. Statistical comparison of MiniDTW with counterpart clustering algorithms on the datasets, and a lower rank score indicates the better performance.

SSNMF with other variants for comparison. We develop a MiniDTW-HAC method that uses hierarchical clustering (*complete* linkage), and MiniDTW-L1 and MiniDTW-L2 methods that use NMF with L1/L2 constraint [36] for seed clustering merging. The optimal clustering results of MiniDTW-HAC, MiniDTW-L1 and MiniDTW-L2 on UCR time series datasets are obtained using the same grid search as MiniDTW. The results in Fig. 5 shows that MiniDTW achieves better statistical clustering results than MiniDTW-HAC, MiniDTW-L1 and MiniDTW-L2. Meanwhile, the results also show that NMF based methods all perform better than the seed clustering merging method using hierarchical clustering.

C. EFFICIENCY ANALYSIS

For the convenience of demonstration, we generate one synthetic dataset similar to [37] and choose six UCR datasets of different sizes and time series lengths to compare the efficiency of MiniDTW. The statistics of the UCR datasets are shown in Table 3. We especially compare MiniDTW with TADPole and SS-PrunedDTW, because they all aim at accelerating time series clustering using DTW. The calculation of the LB_Keogh lower bound matrix (for TADPole) is regarded as the setup-time the same as [5].

1) PERFORMANCE ON UCR DATASETS

The running time results, which are obtained under their optimal parameters, provide the best accuracy in the first experiment, as shown in Fig. 6. The results show that both methods reduce the usage of DTW, i.e., TADPole (using lower/upper bound pruning) and MiniDTW (summarizing datasets with L1-norm distance) are more efficient than SS-PrunedDTW that accelerates DTW calculations. Moreover, it further accelerates MiniDTW and TADPole by replacing the DTW used with the more efficient SS-PrunedDTW.

So, MiniDTW is around 10 time faster than TADPole on the six datasets. We further compare the DTW utilization ratios of MiniDTW and TADPole to show why MiniDTW is more efficient. DTW utilization ratio of an algorithm is defined as $\frac{2x}{n(n-1)}$, where x is the number of DTW calculations

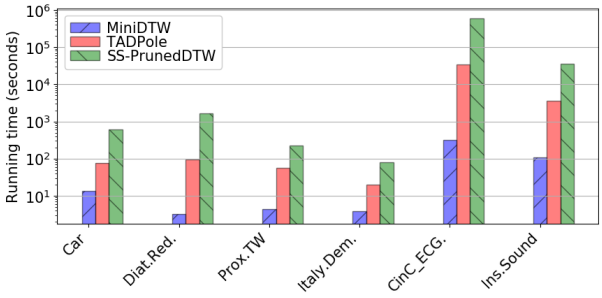


FIGURE 6. Running time of MiniDTW, TADPole and SS-PrunedDTW on the six datasets.

TABLE 4. DTW utilization ratios ($= \frac{2x}{n(n-1)}$) of MiniDTW and TADPole.

Dataset	MiniDTW	TADPole
Car	6.94%	38.07%
Diat.Red.	0.13%	21.05%
Prox.TW	1.82%	54.23%
Italy.Dem.	0.35%	31.40%
CinC_ECG.	0.03%	14.05%
Ins.Sound	0.19%	5.85%
Average	1.58%	24.44%

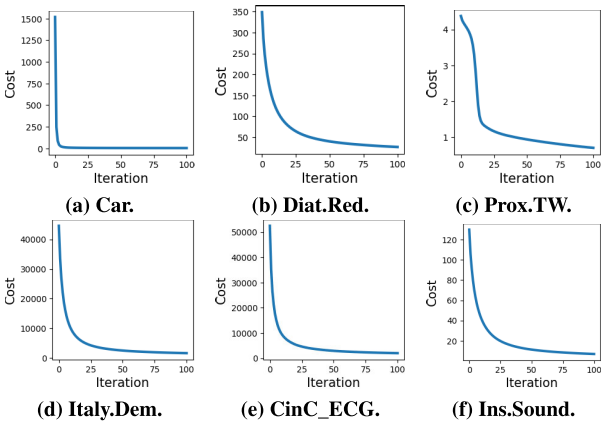


FIGURE 7. Convergence analysis of MiniDTW.

the algorithm adopted and $\frac{1}{2}n(n-1)$ is the baseline (the DTW distance matrix). The results of DTW utilization ratios are shown in Table 4. The average DTW utilization ratio of MiniDTW (1.58%, i.e., 98.42% DTW calculations are avoided) is one magnitude smaller than that of TADPole (24.44%). This observation roughly explains why MiniDTW is much faster than TADPole (due to the quadratic complexity of DTW calculation). Moreover, MiniDTW only requires less than 1% DTW calculations on four datasets; while TADPole uses more than 10% DTW utilization ratio on most datasets.

We further apply the convergence analysis for SSNMF, which merges seed clusters in MiniDTW, and the results are shown in Fig. 7. The proposed SSNMF converges fast in all datasets, i.e., less than 25 iterations are required for most datasets; and this fast convergence also contributes to the runtime efficiency of MiniDTW. Specifically, MiniDTW on Car dataset requires only 4 iterations to reach convergence.

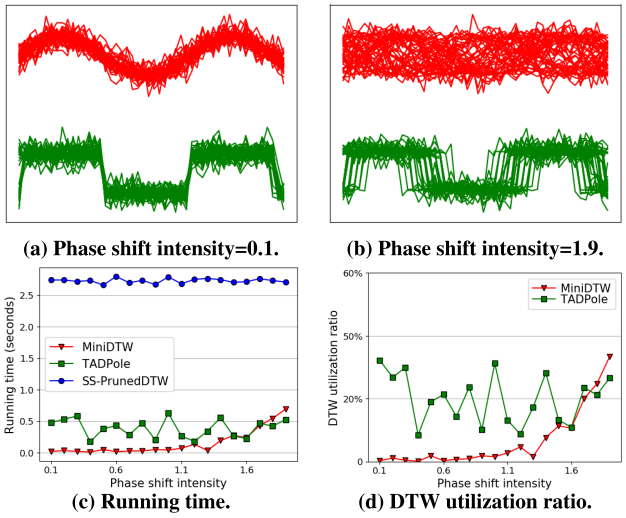


FIGURE 8. MiniDTW, TADPole and SS-PrunedDTW are compared on 19 synthetic datasets with increasing phase shift intensities (from 0.1 to 1.9). The two datasets with the smallest (0.1) and the largest (1.9) phase shift values are shown in (a) and (b), respectively. The running time results changing with phase shift intensities are shown in (c), and the DTW utilization ratios changing with phase shift intensities are shown in (d).

This fast convergence is attributed to the small number of seed clusters, which determines the size of the factorized matrix.

2) PERFORMANCE ON SYNTHETIC DATASET

We compare MiniDTW with TADPole and SS-PrunedDTW on synthetic datasets that comprise different levels of phase perturbations. We generate 19 synthetic datasets using the method in [37] and each dataset contains 100 time series (length $=$ 50) of two classes, which have sinusoidal and rectangular shapes, respectively. We use phase shift as an example of phase perturbation due to its pervasiveness. Phase shift that is randomly selected from a normal distribution is added to each time series. Different datasets select phase shift intensity from different distributions; these distributions have mean values of 0 and $\{0.1, 0.2, \ldots, 1.9\}$ variations, with respect to the 19 datasets. An Gaussian noise ($\mu = 0.1$ and $\delta = 0.5$) is further added to each time series. The two datasets that have the smallest and the largest phase shift intensities are shown in Fig. 8 (a) and (b), respectively. The running time and DTW utilization ratio results are obtained using the same grid search as above experiments. The running time result in Fig. 8 (c) shows that MiniDTW and TADPole (reduce the DTW utilization ratio) constantly run faster than SS-PrunedDTW (accelerates the DTW distance). Meanwhile, the running time of MiniDTW increases with phase shift intensity but is smaller than TADPole before the intensity becomes too large ($\geq$ 1.6); this trend is consistent with the trend of DTW utilization ratio shown in Fig. 8 (d). Specifically, the DTW utilization ratio of MiniDTW increases with phase shift intensity and is larger than TADPole, which has a DTW utilization ratio fluctuating around 20%, when the intensity exceeds 1.6.

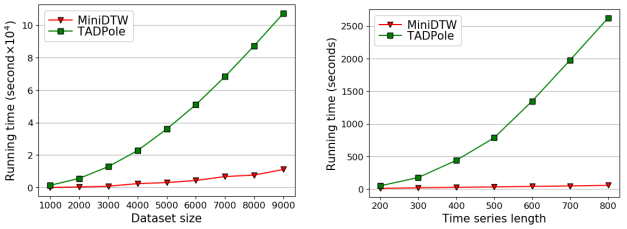


FIGURE 9. The running time comparison between MiniDTW and TADPole with datasets of different sizes (n) and time series lengths (m), using StarLightCurves dataset. The results show that MiniDTW is more efficient than TADPole, and the acceleration rate increases with larger datasets (a) and higher dimensional time series (b).

D. SCALABILITY ANALYSIS

We use StarLightCurves dataset, the largest dataset, considering both dataset size ($n = 9236$) and time series length ($m = 1024$), in the UCR time series archive, to compare the scalability of MiniDTW and TADPole with respect to the different dataset sizes and time series lengths. For fairness, we use the same d_c, i.e., 0.2 multiplied with the largest DTW distance in the dataset, for both MiniDTW and TADPole, and set $\eta = 100$ for MiniDTW to produce the near optimal clustering accuracy on StarLightCurves dataset. The calculation of LB_Keogh lower bound matrix (for TADPole) is regarded as setup time as the previous efficiency analysis [5].

To compare MiniDTW with TADPole for clustering dataset of different sizes, we generate 9 subsets, the size of which vary from 1000 to 9000, by randomly selecting time series from StarLightCurves dataset. The running time results in Fig. 9 (a) show that MiniDTW is more scalable than TADPole in large datasets. Even though MiniDTW and TADPole achieve close running time on the smallest dataset ($n = 1000$), the running time of MiniDTW increases much slower than TADPole on larger datasets. We further compare MiniDTW and TADPole using 7 subsets that have 1000 time series of different lengths, i.e., from 200 to 800, which are segments of time series in StarLightCurves dataset. As shown in Fig. 9 (b), the running time of MiniDTW is nearly linear to the length of time series, while that of TADPole is quadratic since TADPole uses more quadratic DTW calculations during clustering. Therefore, MiniDTW is more scalable than TADPole on large datasets.

VI. CONCLUSION

We propose a novel MiniDTW algorithm, which minimizes the DTW utilization ratio by applying DTW on summarized datasets (with L1-norm distance), to accelerate time series clustering. MiniDTW first uses density-based clustering with L1-norm distance to efficiently summarize the datasets as natural-shaped seed clusters, which contain time series comprising small phase perturbations; and then form the final clusters by merging seed clusters using an effective SSNMF decomposition of the DTW distance matrix of seed cluster centers. The experimental results conducted on the UCR time

series datasets show that MiniDTW reduces 98.52% of DTW utilization and is better than its counterpart, TADPole, which reduces only 75.56% of DTW utilization; and thus MiniDTW is 10 times faster than TADPole, without sacrificing clustering accuracy.

REFERENCES

[1] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering—A decade review," *Inf. Syst.*, vol. 53, pp. 16–38, Oct. 2015.
[2] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Phys. A, Stat. Mech. Appl.*, vol. 519, pp. 127–139, Apr. 2019.
[3] M. Fahim, K. Fraz, and A. Sillitti, "TSI: Time series to imaging based model for detecting anomalous energy consumption in smart buildings," *Inf. Sci.*, vol. 523, pp. 1–13, Jun. 2020.
[4] X. Zheng and P. Wei, "Air transportation direct share time series analysis and forecast," in *Proc. AIAA Aviation Forum*, 2019, p. 3187.
[5] N. Begum, L. Ulanova, J. Wang, and E. Keogh, "Accelerating dynamic time warping clustering with a novel admissible pruning strategy," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 49–58.
[6] R. Neamtu, R. Ahsan, E. Rundensteiner, and G. Sarkozy, "Interactive time series exploration powered by the marriage of similarity distances," *Proc. VLDB Endowment*, vol. 10, no. 3, pp. 169–180, Nov. 2016.
[7] S.-E. Benkabou, K. Benabdeslem, and B. Canitia, "Unsupervised outlier detection for time series by entropy and dynamic time warping," *Knowl. Inf. Syst.*, vol. 54, no. 2, pp. 463–486, Feb. 2018.
[8] C.-S. Perng, H. Wang, S. R. Zhang, and D. S. Parker, "Landmarks: A new model for similarity-based pattern querying in time series databases," in *Proc. 16th Int. Conf. Data Eng.*, 2000, pp. 33–42.
[9] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, Seattle, WA, USA, 1994, vol. 10. no. 16, pp. 359–370.
[10] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. (Jul. 2015). *The UCR Time Series Classification Archive*. [Online]. Available: https://www.cs.ucr.edu/~eamonn/time_series_data/
[11] R. Ding, Q. Wang, Y. Dang, Q. Fu, H. Zhang, and D. Zhang, "YADING: Fast clustering of large-scale time series data," *Proc. VLDB Endowment*, vol. 8, no. 5, pp. 473–484, Jan. 2015.
[12] M. D. Morse and J. M. Patel, "An efficient and accurate method for evaluating time series similarity," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2007, pp. 569–580.
[13] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, Aug. 2008.
[14] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2000, pp. 285–289.
[15] N. Alajlan, "Fast shape matching and retrieval based on approximate dynamic space warping," *Artif. Life Robot.*, vol. 15, no. 3, pp. 309–315, Sep. 2010.
[16] D. F. Silva, R. Giusti, E. Keogh, and G. E. A. P. A. Batista, "Speeding up similarity search under dynamic time warping by pruning unpromising alignments," *Data Mining Knowl. Discovery*, vol. 32, no. 4, pp. 988–1016, Jul. 2018.
[17] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, Oct. 2007.
[18] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery (DMKD)*, 2003, pp. 2–11.
[19] J. Paparrizos and L. Gravano, "Fast and accurate time-series clustering," *ACM Trans. Database Syst.*, vol. 42, no. 2, pp. 1–49, Jun. 2017.
[20] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proc. 18th Int. Conf. Data Eng.*, 2002, pp. 673–684.
[21] C. Euán, H. Ombao, and J. Ortega, "The hierarchical spectral merger algorithm: A new time series clustering procedure," *J. Classification*, vol. 35, no. 1, pp. 71–99, Apr. 2018.
[22] D. Guijo-Rubio, A. M. Durán-Rosal, P. A. Gutiérrez, A. Troncoso, and C. Hervás-Martínez, "Time-series clustering based on the characterization of segment typologies," *IEEE Trans. Cybern.*, early access, Jan. 15, 2020, doi: 10.1109/TCYB.2019.2962584.
[23] Y. Guo, W. Liao, Q. Wang, L. Yu, T. Ji, and P. Li, "Multidimensional time series anomaly detection: A GRU-based Gaussian mixture variational autoencoder approach," in *Proc. Asian Conf. Mach. Learn.*, 2018, pp. 97–112.
[24] H. Guo, W. Pedrycz, and X. Liu, "Hidden Markov models based approaches to long-term prediction for granular time series," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2807–2817, Oct. 2018.
[25] D. Hallac, S. Vare, S. Boyd, and J. Leskovec, "Toeplitz inverse covariance-based clustering of multivariate time series data," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 215–223.
[26] X. Huang, Y. Ye, L. Xiong, R. Y. K. Lau, N. Jiang, and S. Wang, "Time series k-means: A new k-means type smooth subspace clustering for time series data," *Inf. Sci.*, vols. 367–368, pp. 1–13, Nov. 2016.
[27] K. E. Smith, P. Williams, K. J. Bryan, M. Solomon, M. Ble, and R. Haber, "Shepard interpolation neural networks with K-means: A shallow learning method for time series classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–6.
[28] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, vol. 344. Hoboken, NJ, USA: Wiley, 2009.
[29] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognit.*, vol. 44, no. 3, pp. 678–693, Mar. 2011.
[30] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2011, pp. 177–186.
[31] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[32] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar. 1964.
[33] J. Woo, J. L. Prince, M. Stone, F. Xing, A. D. Gomez, J. R. Green, C. J. Hartnick, T. J. Brady, T. G. Reese, V. J. Wedeen, and G. El Fakhri, "A sparse non-negative matrix factorization framework for identifying functional units of tongue behavior from MRI," *IEEE Trans. Med. Imag.*, vol. 38, no. 3, pp. 730–740, Mar. 2019.
[34] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2006, pp. 126–135.
[35] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, Mar. 2005.
[36] A. Cichocki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. 92, no. 3, pp. 708–721, 2009.
[37] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang, "Salient subsequence learning for time series clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2193–2207, Jun. 2018.

BORUI CAI received the bachelor's and master's degrees in engineering from Beihang University (BUAA), in 2010 and 2013, respectively, and the Ph.D. degree from Deakin University, in 2021. He is currently a Postdoctoral Research Fellow with the School of Information, Deakin University. He worked for the Chinese Academy of Sciences from 2013 to 2016, and worked with Xilinx, Inc., in 2017. His research interests include time series analysis pattern recognition and privacy protection.

GUANGYAN HUANG (Member, IEEE) received the Ph.D. degree in computer science from Victoria University, Footscray, VIC, Australia, in 2012. She was an Assistant Professor with the Institute of Software, Chinese Academy of Sciences, from 2007 to 2009, and visited the Platforms and Devices Centre, Microsoft Research Asia, in the last half of 2006. She is currently an Associate Professor with the School of Information Technology, Deakin University, Burwood, VIC, Australia. She has 110 publications mainly in data mining, the IoT/sensor networks, text analytics, image/video processing, emotion modeling, and multimodal data fusion. She was a recipient of an ARC Discovery Early Career Researcher awards (DECRA) a Fellowship and the Chief Investigator of two ARC Discovery Projects.

NAJMEH SAMADIANI received the bachelor's degree in computer engineering and the master's degree in artificial intelligence in 2012 and 2014, respectively. She is currently pursuing the Ph.D. degree with Deakin University, Burwood, VIC, Australia. She was a Lecturer with the Kosar University of Bojnord, Iran, from 2015 to 2018. Her research interests include image/video processing, human emotion modeling, expert systems, and pattern recognition.

GUANGHUI LI received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Professor with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. He has published more than 70 papers in journals or conferences. His research interests include wireless sensor networks, fault tolerant computing, and nondestructive testing and evaluation. His research was supported by the National Natural Science Foundation of China, Jiangsu Provincial Science and Technology Foundation, and other governmental and industrial agencies.

CHI-HUNG CHI received the Ph.D. degree from Purdue University, West Lafayette, IN, USA. He is currently a Senior Principal Research Scientist of Data61, Commonwealth Scientific and Industrial Research Organization (CSIRO), Australia. Before he joint CSIRO, he has worked in industry (Philips Research Laboratory, USA, IBM, Poughkeepsie, NW, USA) and universities (the Chinese University of Hong Kong, the National University of Singapore, and Tsinghua University) for more than 20 years. He has published more than 260 international journal and conference papers and edited ten books; he also holds six U.S. patents. His research interests include cybersecurity, behavior modeling, knowledge graph, data engineering and analytics, cloud and service computing, social computing, the Internet-of-Things, and distributed computing.