ShapeDBA: Generating Effective Time Series Prototypes Using ShapeDTW Barycenter Averaging

Ali Ismail-Fawaz[1(✉)], Hassan Ismail Fawaz[1,4], François Petitjean[2], Maxime Devanne[1], Jonathan Weber[1], Stefano Berretti[3], Geoffrey I. Webb[2], and Germain Forestier[1,2]

[1] IRIMAS, Universite de Haute-Alsace, Mulhouse, France
{ali-el-hadi.ismail-fawaz,hassan.ismail-fawaz,maxime.devanne, jonathan.weber,germain.forestier}@uha.fr

[2] Department of Data Science and Artificial Intelligence, Monash University, Melbourne, Australia
{francois.petitjean,geoff.webb,germain.forestier}@monash.edu

[3] MICC, University of Florence, Florence, Italy
stefano.berretti@unifi.it

[4] Ericsson Research, Massy, France
hassan.ismail.fawaz@ericsson.com

Abstract. Time series data can be found in almost every domain, ranging from the medical field to manufacturing and wireless communication. Generating realistic and useful exemplars and prototypes is a fundamental data analysis task. In this paper, we investigate a novel approach to generating realistic and useful exemplars and prototypes for time series data. Our approach uses a new form of time series average, the ShapeDTW Barycentric Average. We therefore turn our attention to accurately generating time series prototypes with a novel approach. The existing time series prototyping approaches rely on the Dynamic Time Warping (DTW) similarity measure such as DTW Barycentering Average (DBA) and SoftDBA. These last approaches suffer from a common problem of generating out-of-distribution artifacts in their prototypes. This is mostly caused by the DTW variant used and its incapability of detecting neighborhood similarities, instead it detects absolute similarities. Our proposed method, ShapeDBA, uses the ShapeDTW variant of DTW, that overcomes this issue. We chose time series clustering, a popular form of time series analysis to evaluate the outcome of ShapeDBA compared to the other prototyping approaches. Coupled with the k-means clustering algorithm, and evaluated on a total of 123 datasets from the UCR archive, our proposed averaging approach is able to achieve new state-of-the-art results in terms of Adjusted Rand Index.

Keywords: Time Series · Clustering · Dynamic Time Warping · Time Series Averaging · ShapeDTW

G. Ifrim et al. (Eds.): AALTD 2023, LNAI 14343, pp. 127–142, 2023.
https://doi.org/10.1007/978-3-031-49896-1_9

1 Introduction

Time series data can now be seen in many real life problems. This data is starting to be of interest in many research fields. For instance time series can be found in medical data such as ECG signals, in human motion data, in satellite images, etc. Generating exemplars and prototypes for time series data is an essential problem that could be used in many areas. For example, time series averaging is being used to generate synthetic data in order to augment the training data and boost supervised models [5,11] or used to make the classification task more accurate [16]. Time series prototyping can also be used for explainability [6].

One challenge when prototyping time series data is evaluation, which is addressed in most of the cases using clustering, a fundamental machine learning tool in data analysis. Clustering is a machine learning unsupervised problem that aims to discover a set of clusters in the data that should correspond to the same distribution and the previously unseen class label. Clustering for time series data has been very much addressed in the literature [1,13]. Varying from machine learning tools such as k-means and k-medoids [7] to the usage of deep learning [12]. Unlike other data types, basic machine learning clustering algorithms need to be adapted to the case of temporal data. For instance, the k-means algorithm aims to minimize a distance between the samples in a cluster and the centroid of this cluster. This distance is usually the Euclidean distance, but the implicit assumption using such metric is that the input samples are made of independent feature points. However, this is not the case in time series data, where each feature point, referred to as time stamp, is dependent with all other time stamps. This is referred to as a temporal correlation, which obligates the definition of a replacement of the Euclidean distance in the k-means algorithm. For this reason, time series similarity measures such as DTW and SoftDTW have been used instead and showed a significant improvement over the usage of the Euclidean distance.

A further issue with the naive way of using the k-means algorithm, is the averaging phase to define the clusters' centroids. The averaging method used in the k-means algorithm is the arithmetic mean, which presents the same problem as the Euclidean distance. For this reason, a novel averaging method was proposed that uses the DTW similarity measure in order to produce a meaningful centroid. This technique, DBA, showed to perform significantly better than other naive approaches. The problem of finding a meaningful average for time series data presents much more challenges than defining the similarity metric. This is due to the challenge in defining what an average time series does represent. However, finding a meaningful average presents a much higher impact on the performance of the k-means algorithm than defining the similarity measure. For these reasons, we address the clustering problem by producing a more respectful averaging algorithm for time series data.

The defined averaging techniques for time series data until now suffer from a common problem of generating out-of-distribution artifacts (see Fig. 3). This problem occurs because these averaging techniques do not look into the neighborhood of each time stamp in the time series data. Instead, the averaging occurs after aligning each time stamp of the centroid with the ones in the time series dataset. In this work, we propose incorporating ShapeDTW [20] into the DBA algorithm in order to overcome this issue. ShapeDTW is a DTW variant that manages to avoid aligning two time stamp that have closer values but in a significantly different neighborhood. This last case study occurs often in time series data and is the main reason, to the best of our knowledge, for the existence of the generated artifacts. The ShapeDTW similarity measure coupled with DBA, i.e., the proposed ShapeDBA algorithm, is coupled with the k-means algorithm in order to apply clustering on time series data.

The contributions of this work are:

- Proposing a novel averaging algorithm ShapeDBA based on ShapeDTW;
- Extensive experiments on the UCR archive showing that ShapeDBA achieves state-of-the-art performance following the Adjusted Rand Index metric;
- Efficient implementation of ShapeDTW resulting in ShapeDBA being faster than SoftDBA.

2 Related Work

Definitions The following definitions will be used throughout the rest of the paper:

- Univariate Time Series (UTS) $\mathbf{x} = \{x_0, x_1, \ldots, x_{L-1}\}$ is a sequence of length L made of correlated data points equally separated in time.
- A TSC dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=0}^{N-1}$ is a collection of N time series with their corresponding labels y.
- A Time Series Average (TSA) $\mathbf{x}_{avg} = \{x_0, x_1, \ldots, x_{L-1}\}$ is a time series of length L that represents the average of a part of $\mathcal{D}$.

2.1 Time Series Similarity

Euclidean Distance (ED). The naive solution to define a similarity is by using the Euclidean Distance (ED). This metric defined in (1) supposes that the two time series are aligned on the time axis, which is not the case most of the times.

$$ED(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{t=0}^{L-1} (x_{1,t} - x_{2,t})^2}. \tag{1}$$

Another limitation that this similarity measure presents is that both time series should have the same length. In case of unequal length samples in the dataset, the problem should be addressed as dicussed in [18] such as padding, uniform scaling, etc.

Dynamic Time Warping (DTW). The following measure [14] is a more general formulation of the ED that is: (a) independent of the time series length, and (b) aligns the two time series on the time axis. The formulation of the DTW is presented in (2).

$$DTW(\mathbf{x}_1, \mathbf{x}_2) = \min_{\pi \in \mathcal{M}(\mathbf{x}_1, \mathbf{x}_2)} (\sum_{(i,j)\in\pi} |x_{1,i} - x_{2,j}|^q)^{1/q}, \tag{2}$$

with $\mathcal{M}(\mathbf{x}_1, \mathbf{x}_2)$ being the set of all possible alignment paths on the time axis between $\mathbf{x}_1$ and $\mathbf{x}_2$. The parameter q is the order of the Minkovski distance used, if $q = 2$ then the distance is set to be Euclidean. The hypothesis in this case is that $\mathbf{x}_1$ and $\mathbf{x}_2$ have different lengths, L_1 and L_2, respectively. The goal of DTW is to find the optimal path π of length L_π that minimizes the loss in (2). Some conditions should be applied on the optimal path as listed below:

- $\pi_0 = (0, 0)$;
- $\pi_{L_\pi - 1} = (L_1 - 1, L_2 - 1)$;
- The elements of the path should be a strictly increasing sequence in the indices i and j of π.

Soft Dynamic Time Warping (SoftDTW). One issue of the DTW measure is its non-differentiability. For this reason, in [3] the Soft Dynamic Time Warping (SoftDTW) was proposed, which is differentiable. This differentiability exists because of the replacement of the hard min function in (2) by the softer version as seen in (3):

$$softmin^\gamma(x_0, \ldots, x_{L-1}) = -\gamma . \log(\sum_{i=0}^{L-1} e^{-x_i/\gamma}). \tag{3}$$

where the parameter γ controls the smoothness of the *softmin* function. The smaller the value of γ, the closer the *softmin* function is to the hard min.

Shape Dynamic Time Warping (ShapeDTW). In [20], a different version of DTW was proposed that, instead of aligning all the time series at the same time, aligns transformations of sub-sequences of the time series. This is done in order to preserve the fact that the alignment between two time stamps of two different time series takes into consideration the structure of their neighborhoods. For the mathematical definition of ShapeDTW, let us assume $\mathcal{F}$ is a descriptor function, x_1 and x_2 two univariate time series of lengths L_1 and L_2, respectively. The first step is to extract the sub-sequences of length l from x_1 and x_2 denoted by $\mathcal{X}_1$ and $\mathcal{X}_2$ represented as two multivariate time series of shape (L_1, l) and (L_2, l), respectively. The second step is to extract the descriptors from the sub-sequences using $\mathcal{F}$ and produce $\mathcal{D}_1 = \mathcal{F}(X_1)$ and $\mathcal{D}_2 = \mathcal{F}(\mathcal{X}_2)$ of shapes (L_1, d) and (L_2, d), respectively, where d is the target dimension. The ShapeDTW measure comes down to the following optimization problem:

$$ShapeDTW(x_1, x_2) = \min_{\pi \in \mathcal{M}(x_1, x_2)} (\sum_{(i,j)\in\pi} |\mathcal{D}_{1,i} - \mathcal{D}_{2,j}|^q)^{1/q} \tag{4}$$

The above definition can simply be adapted to multivariate time series as mentioned in the original work [20] by extracting multivariate sub-sequences and applying the descriptors on each dimension independently or by finding a suitable multivariate descriptor function.

2.2 Time Series Averaging - Clustering

Time Series Clustering. Given a time series dataset, usually an unlabeled one, the goal of the clustering algorithm is to learn how to group time series samples that should belong to the same class label together. A well known clustering algorithm is the k-means one, which learns how to group time series samples given their distance to a cluster's centroid. For this reason, a definition of a time series cluster centroid should be defined.

Dynamic Time Warping Barycenter Averaging (DBA). To define an average of a collection of time series, in [17] the usage of DTW measure was proposed in order to find the optimal average that takes into consideration the misalignment between the samples of this collection. In other words, given two time series, the DBA algorithm defines for each time stamp its barycenter by taking the average of all the aligned values. DBA has proven to be very effective in clustering using the k-means algorithm.

Soft Dynamic Time Warping Barycenter Averaging (SoftDBA). In [3], authors also proposed the replacement of DTW in the DBA algorithm by using SoftDTW instead. Our proposed approach, called SoftDBA, is shown to work better than DBA in clustering and classification.

3 Proposed Approach

3.1 Shape Dynamic Time Warping Barycenter Averaging (ShapeDBA)

ShapeDBA follows the same methodology of DBA and SoftDBA that is averaging over the aligned time stamps. The key difference of ShapeDBA is the usage of the ShapeDTW [20] aligning method of time series data. The ShapeDBA algorithm can be summarized in the following steps:

- **Step 1**: Initialize the average time series, for example choose a random selection of the time series set in question;
- **Step 2**: Find the aligned points of each time stamp of the average series with all the samples of the data. We call the time stamps of all the samples aligned with a given time stamp t of the average series as $assoc_t = \{assoc_{t_0}, assoc_{t_1}, \ldots, assoc_{t_{A-1}}$, where A is the number of associated time stamps with t;

- **Step 3**: For each time stamp t of the average series, the resulting average is the *barycenter* of $assoc_t$.
  Where $barycenter(assoc_{t_0}, assoc_{t_1}, \ldots, assoc_{t_{A-1}}) = \frac{1}{A}\sum_{i=0}^{A-1} assoc_{t_i}$;
- Repeat from Step 2 until convergence.

3.2 Clustering with ShapeDBA

The k-means clustering algorithm in machine learning can be used with any time series averaging technique, coupled with any time series similarity measure. The averaging method, i.e., ShapeDBA for instance, is used to find the centroids of each cluster during the training phase. The similarity measure is then used to calculate the distance of each series in the data to the centroid of each cluster.

In the rest of this paper, we refer to the following coupling for applying the k-means clustering algorithm:

- DBA: the DBA as an averaging method coupled with the DTW as a similarity measure;
- MED: the arithmetic mean as an averaging technique coupled with the Euclidean Distance (ED) as a similarity measure; MED finds iteratively the arithmetic average series, as in DBA, without taking into consideration the temporal alignment between the prototype and the samples;
- SoftDBA: the SoftDBA as an averaging method coupled with the SoftDTW as a similarity measure;
- ShapeDBA: the ShapeDBA as an averaging method coupled with the ShapeDTW as a similarity measure.

3.3 Implementation Efficiency

The ShapeDTW algorithm comes down to applying the original DTW similarity measure on the transformed input time series. In the univariate case coupled with the 'identity' descriptor of each neighborhood [20], the transformed time series is a multivariate version. For each time stamp, its neighborhood is added as a Euclidean vector to form a multivariate time series. When applying the DTW similarity measure on this transformed series, the algorithm is simply computing the Euclidean distance between the channel vectors of a pair of time stamps. This creates a computational waste when sliding the reach window as illustrated in Fig. 1. This problem only occurs when the descriptor is set to be the identity transformation.

To avoid this issue, the Euclidean pairwise distance between the two time series in question is computed as a first step. This distance matrix is then padded with its edges values $reach/2$ times. We then slide a window of height and width equal to the time series lengths on this Euclidean distance matrix. The direction of the sliding window is over the second diagonal of the distance matrix. The results captured on the sliding window are accumulated in a zero-initialized matrix. After accumulating all the information into the new distance matrix, we apply the DTW algorithm on the new matrix. This implementation saves time by avoiding unnecessary computations. A summary of this efficient implementation of the ShapeDTW can be seen in Fig. 2.

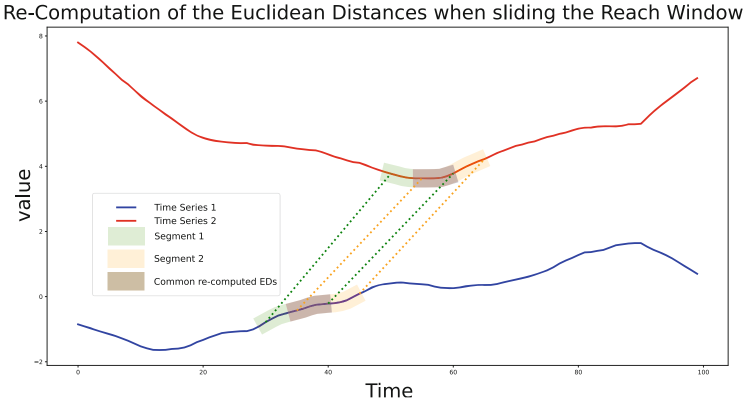


Fig. 1. Computation of the ShapeDTW measure between two time series. It can be observed that the common area between the two sliding window is re-computed.

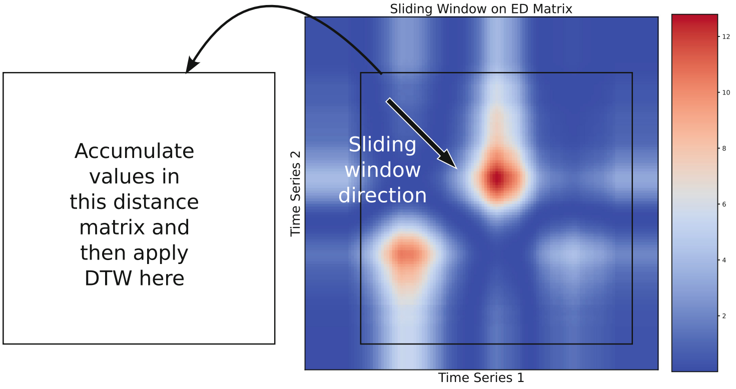

Accumulate values in this distance matrix and then apply DTW here
Sliding Window on ED Matrix
Sliding window direction
Time Series 2
Time Series 1
12
10
8
6
4
2

Fig. 2. A more efficient implementation of the ShapeDTW measure when the descriptor is set to be the identity. Instead of applying the DTW on the multivariate transformation of the time series, a window slides on the ED matrix between the two time series. The captured frames are accumulated in another zero-initialized matrix on which the DTW algorithm is then applied.

3.4 Reach Value Control

The hyperparameter of ShapeDTW, called "reach", controls the length of the neighborhood of each time stamp to be used for the alignment. This value makes the ShapeDTW algorithm a general definition that includes two similarity measures: the DTW and the Euclidean distance. For instance, on the one hand, if the reach value is set to 1, the algorithm will behave just as the original DTW similarity measure. This is due to the fact that the length of the neighborhood of each time stamp will be set to 1 leading to taking into consideration only this time stamp. On the other hand, if the reach is large enough, i.e., ∞, the ShapeDTW algorithm will behave just as the Euclidean distance. This is due to the fact that for each time stamp, the neighborhood length will be larger than the time series itself. In this work, we set the value of the reach to 30 given it was the value used in the original paper [20].

4 Results

4.1 Experimental Setup

Datasets. All the experiments were conducted on 123 datasets of the UCR archive [4]. The total number of datasets in the UCR archive since 2018 is 128, but five datasets were excluded from the experiments given the large length of the time series. This was crucial given the quadratic time complexity of most of the executed algorithms with respect to the time series length. All of the datasets were Z-normalized in order to have a zero mean and unit standard deviation for each time series. The clustering algorithms are trained on the combination of the train test splits for all the 123 datasets used in the experiments. It is important to note that some datasets of the UCR archive are simply another train test split of an exiting dataset. This does not occur much, which would mean that the clustering algorithm is done on the same dataset more than one time. The source code of this work is publicly available for reproducibility[1].

Removing Bias. A typical problem in non-deterministic estimators in machine learning is the biased performance to a given initial setup. This problem occurs in many problems such as deep learning where the performance can be biased to an initialization of the weights. In this clustering task, the bias in performance comes down to the initialization of the clusters before the k-means algorithm starts its optimization. To avoid this bias, we do the same experiments five different times, each time with different initial clusters and present the average performance on each dataset. However, this may raise the issue of fairness among multiple clustering algorithms experimented with. This is due to the probable second bias of a method to a specific five initial clusters. To fix this bias as well, in this work the same initial clusters are used over the five experiments for all clustering algorithms. Given that for clustering experiments using k-means and k-shape need the initial clusters, which are usually randomly selected, it would create an issue if not all algorithms use the same initial clusters. For this reason, we made sure that for all the experiments done, for the same dataset, all of the clustering variants used the same initial clusters. This was done with five different initial clusters and the average performance is presented in order to remove any variance in the results.

4.2 Qualitative Evaluation of DBA Variants

Given a set of time series example from the GunPointMaleVersusFemale dataset of the UCR archive, we can generate the average time series to compare and

[1] https://github.com/MSD-IRIMAS/ShapeDBA.

analyse the limitation of each technique. In Fig. 3, the generated average time series is presented from a set of samples from the GunPointMaleVersusFemale dataset. It can be seen that for the naive way of averaging, using the Euclidean distance, i.e., Arithmetic Mean, differs from all other approaches by the shifting issue. In other words, the Arithmetic Mean does not take into consideration the time warping and miss-aligned information between the samples of the example set.

Comparing other alignment techniques with ShapeDBA, the TSA almost is placed in the same time interval. The difference between warping methods is that DBA and SoftDBA present additional artifacts in the shape. This results in a TSA that includes some small peaks (red circles in Fig. 3) that do not appear in the original set of time series. ShapeDBA avoids generating this kind of artifacts given the usage of shapeDTW. ShapeDTW's advantage is to avoid aligning a time stamp with an outlier, which is obtained thanks to the ability of the method of aligning time stamp in specific sub-sequence of the time series. This advantage leads ShapeDBA to generate a prototype that is more likely to be randomly selected from the dataset distribution.

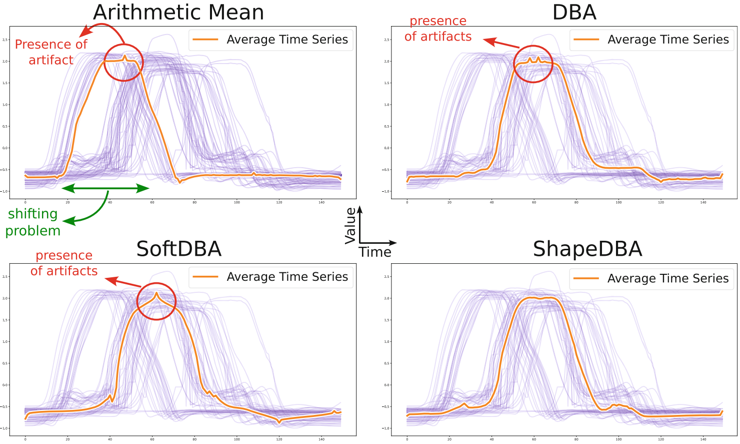


Fig. 3. A qualitative evaluation of the proposed average technique compared to other approaches on a GunPoint dataset. The ShapeDBA algorithm is the only approach to not generate out-of-distribution artifacts. (Color figure online)

4.3 Quantitative Evaluation

Competitor. In this work, we compare the proposed method to other time series averaging techniques as detailed in Sect. 3.2. The state-of-the-art model for time series clustering is k-shape [15]. This algorithm is an improvement over the k-means algorithm on time series data by using a Shape Based Distance (SBD) that uses the cross-correlation between two time series instead of an alignment measure. Until now, to the best of our knowledge, k-shape is the state-of-the-art and most efficient clustering method on time series data.

Adjusted Rand Index (ARI). The Adjusted Rand Index (ARI) [9] is a new fixed version of the original Rand Index (RI) defined in (5). Given the true labels of the time series dataset $\mathbf{y}$ and the predicted labels by the clustering algorithm $\hat{\mathbf{y}}$, the RI is calculated as follows:

$$RI(\mathbf{y},\hat{\mathbf{y}}) = \frac{TP+TN}{TP+FP+FN+TN}, \tag{5}$$

where, TP and TN stand, respectively, for True Positive and True Negative, while FP and FN stand for False Positive and False Negative, respectively.

The RI counts the number of pairs that are present in the intersection of both sets of true and predicted labels as well as the number of pairs that exist in the difference of these two sets. This metric, however, presents a limitation: a high RI should indicate that the two clusters in question are almost identical, which is not always the case. The RI may favor high identical clusters without taking into consideration the case where the intersection was randomly generated. This is due to the fact that the expected value of the RI is not constant between two random clusters. This random chance can be generated when the number of clusters becomes high enough that the probability of a pair to be in both clusters is large. For this reason, the Adjusted Rand Index (ARI) is proposed with a scaled version that takes into account this randomness by setting the value 0.0 for the random chance. The ARI presented in (6) is bounded between -0.5 indicating no similarity and 1.0 for a perfect similarity between the clusters.

$$ARI(\mathbf{y},\hat{\mathbf{y}}) = \frac{RI(\mathbf{y},\hat{\mathbf{y}}) - E[RI]}{1.0 - E[RI]}, \tag{6}$$

where $E[RI]$ is the expected value of RI.

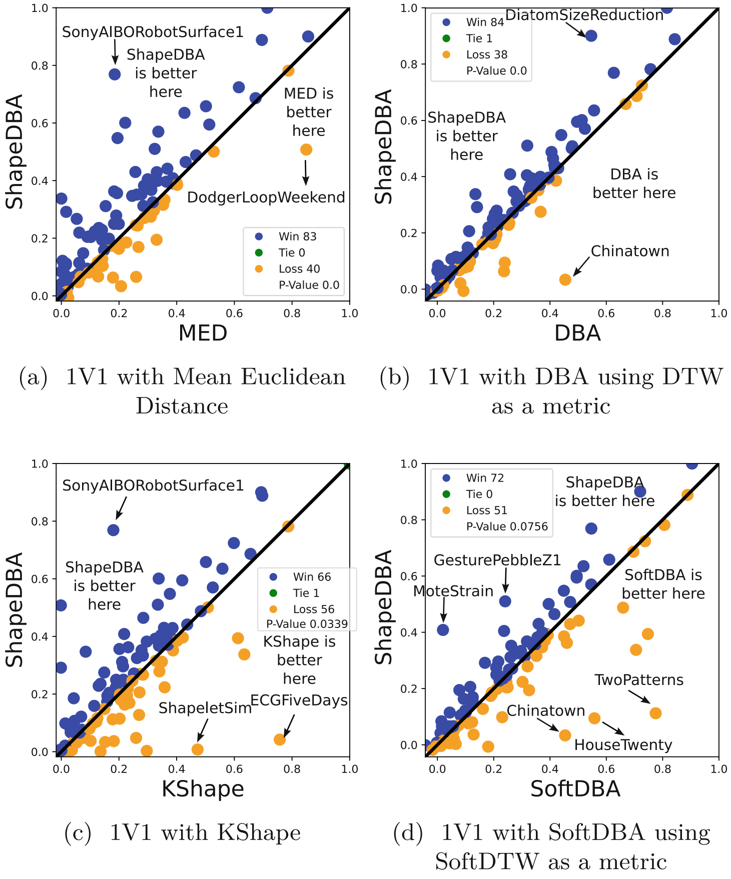
We present in the following three different ways to compare the performance of each clustering method on the total of 123 datasets of the UCR archive.

One-vs-One Comparison: In this approach, we present a scatter plot of all the pairwise comparisons between k-means with ShapeDBA and the approaches in the literature. Each point visualized in Fig. 4 represents one dataset, the x-axis presents the ARI value on this dataset using a method from the literature and the y-axis the ARI value using ShapeDBA. The Win-Tie-Loss count is presented in the legend of each One-vs-One scatter plot as well as a p-value. This latter p-value is produced using the Wilcoxon Signed Rank Test [19]. If this p-value is larger than the threshold 0.05, than the difference in performance between the comparates in question is not considered statistically significant.

It is clear from Figs. 4a, 4b, and 4c that the usage of ShapeDBA as an averaging method in k-means is significantly better than the baseline, i.e., ED and DBA with k-means and significantly better than the state-of-the-art k-shape.

From Fig. 4d it can be seen that even though ShapeDBA presents more wins compared to SoftDBA, the difference in performance is still not significantly different. In what follows, we show however that ShapeDBA is way faster than SoftDBA.

Analysing Outliers. Some unique outliers from the One-vs-One scatter plots are clear to favor either ShapeDBA or the other approaches. For instance, compared to k-shape, ShapeDBA does not perform well (low ARI) on two datasets: ShapeletSim and ECGFiveDays. On the one hand, given knowledge on the UCR archive datasets, we believe that no correct conclusion can be found on ShapeletSim given that this dataset is simply a simulation of random data. On the other hand, the ECGFiveDays dataset presented in Fig. 5 is a unique example to show case the disadvantage of ShapeDBA.



(a) 1V1 with Mean Euclidean Distance

(b) 1V1 with DBA using DTW as a metric

(c) 1V1 with KShape

(d) 1V1 with SoftDBA using SoftDTW as a metric

Fig. 4. 1v1 Comparison between using k-means with ShapeDBA-ShapeDTW and other approaches from the literature using the Adjusted Rand Index clustering metric.

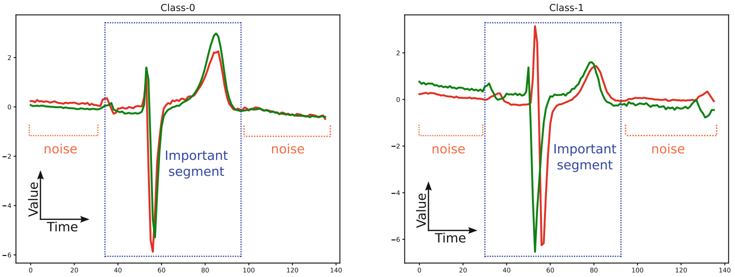


Fig. 5. Two examples from each class taken from the ECGFiveDays dataset of the UCR archive. Most time stamps of this dataset represent noise and the important neighborhood of the time stamp is just in the middle of the whole time series.

This dataset is mostly made of noisy time stamps with an information compressed in the important segments placed in the middle of the time series as seen in Fig. 5. For this reason, ShapeDTW will be adding noise in the optimization steps. A clear winner on the SonyAIBORobotSurface1 dataset, however, is ShapeDBA compared to k-shape with almost a 0.6 difference in the ARI. After analysing this dataset, still no hard conclusions can be found but this is not a special case for ShapeDBA given that MED, DBA and SoftDBA perform better than k-shape on this dataset. Suggesting that it is k-shape underperforming on this dataset.

Comparing ShapeDBA to DBA, it seems as if ShapeDBA has an advantage over the DiatomSizeReduction dataset, which suffers from the lack of training samples with only four samples per class label.

Critical Difference Diagram (CDD): A technique to compare multiple estimators by reducing the metrics on multiple datasets into a one dimensional view. This one dimensional view is presented by using the average rank of each method on the total of the 123 datasets used. The best performing clustering approach is the one with the lowest rank as for instance ShapeDBA in Fig. 6. The CDD used in this work utilizes, as proposed in [2], the Wilcoxon Signed-Rank Test [19] coupled with the Holm multiple test correction [8] in order to generate the cliques. If a clique links a set of comparates in the CDD, this represents that the differences in performance between this set of comparates is not statistically significant.

Multi-Comparison Matrix (MCM) : was proposed in [10] arguing that CDD has some limitations that can miss-lead the interpretation of the results. First, one important issue with CDD as mentioned in [10] is the instability of the average rank. For instance the average rank can easily be manipulated by the addition or removal of some comparates. For this reason, MCM proposes the usage of a descriptive statistics that does not change with this addition and removal of comparates. This statistics is the average performance on the total of the 123 datasets used, in our case it is the average ARI over these datasets for each clustering approach. Second, a common issue of the CDD is the usage

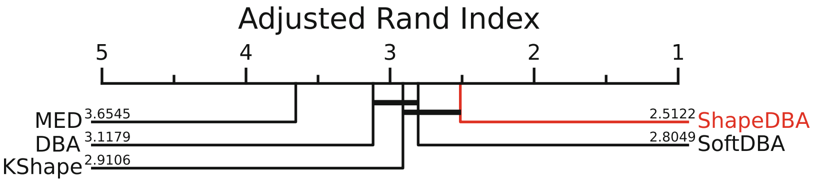


Fig. 6. Critical Difference Diagram showing the average rank of the ARI score over the datasets of the UCR archive.

of the multiple test correction, which is unstable to the addition and removal of comparates. Finally, a major limitation with only using the CDD is the lack of pairwise comparison information. The MCM proposed in [10] overcomes these three problems by using the average performance instead of the average rank to order the comparates, not applying a multiple test correction for the produced Wilcoxon p-values and presenting the pairwise comparisons between comparates. The MCM in Fig. 7 shows that SoftDBA is the winning approach given the average ARI with not much difference with the average ARI of ShapeDBA that comes in second place. A full pairwise and multi-comparates comparison between all clustering techniques discussed in this work on the ARI metric is presented in Fig. 10.

In what follows, we did a computational runtime comparison between all approaches. We show that although ShapeDBA does not outperform in significant manner SoftDBA, it is however faster.

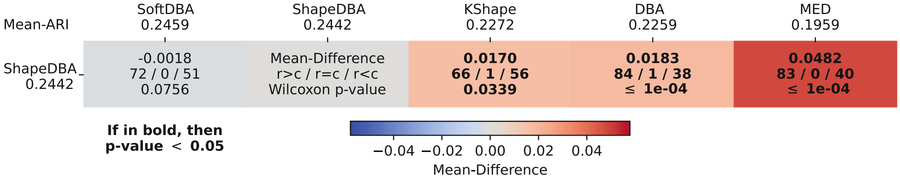


Fig. 7. A Multi-Comparison Matrix showing the proposed approach's performance compared to other approaches using a tool that is stable to the addition/removal of new classifiers.

Computational Runtime. Given that all experiments were conducted on the same machine with the same environment, fairness in time computation comparison stands here. By keeping track of the total computation time for each clustering approach, averaged over five initialization, we can apply the same comparison techniques as for the ARI. In Fig. 8, the CDD of the computational runtime is presented. Given that in the case of runtime, the lower the time the better, and to keep the ordering of the average rank as lower is better, we multiplied the values of the computational time by -1. It is clear from the CDD plot that the fastest approach is k-shape and the slowest one is SoftDBA. The reason behind the fast computation of k-shape is essentially because of the usage of the Fast Fourier Transform (FFT), while doing the cross-correlation between the time series. However, with the help of the efficient implementation used in ShapeDBA, the computation is way faster than SoftDBA.

For ARI, we generated the MCM as well for the computational time comparison in Fig. 9. On average of 123 datasets, ShapeDBA is 1.7 times faster than SoftDBA with 109 wins for ShapeDBA in terms of computational runtime. It is important to note that in this case of MCM, the Win-Tie-Loss count considers the lower the better.

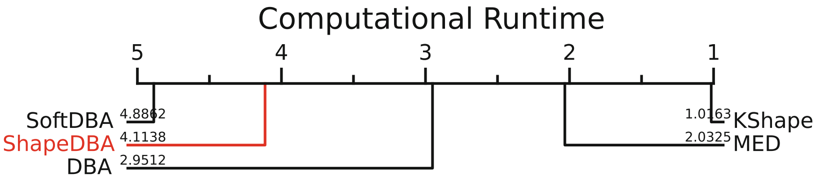
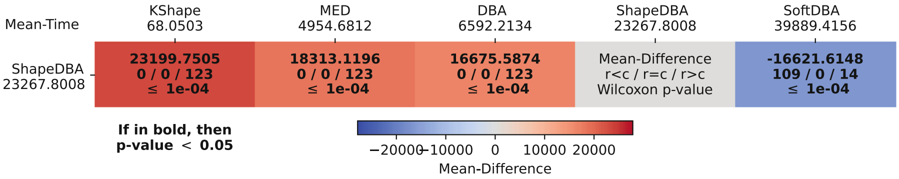


Fig. 8. Critical Difference Diagram showing the average rank of the duration (in seconds) of the k-means algorithm over the datasets of the UCR archive.



Mean-Time	KShape 68.0503	MED 4954.6812	DBA 6592.2134	ShapeDBA 23267.8008	SoftDBA 39889.4156
ShapeDBA 23267.8008	**23199.7505** **0 / 0 / 123** **≤ 1e-04**	**18313.1196** **0 / 0 / 123** **≤ 1e-04**	**16675.5874** **0 / 0 / 123** **≤ 1e-04**	Mean-Difference r<c / r=c / r>c Wilcoxon p-value	**-16621.6148** **109 / 0 / 14** **≤ 1e-04**

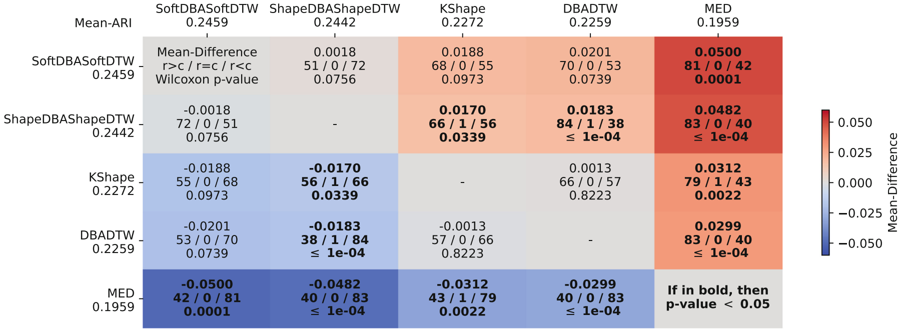
If in bold, then p-value < 0.05

−20000
−10000
0
10000
20000
Mean-Difference

Fig. 9. A Multi-Comparison Matrix showing the proposed approach's duration (in seconds) compared to other approaches using a tool that is stable to the addition/removal of new classifiers.

5 Conclusion

In this work, we addressed the problem of Time Series Averaging (TSA) using elastic distances. We proposed a novel TSA approach, ShapeDBA, based on the similarity measure ShapeDTW similarity measure. We showed that ShapeDBA has the ability to preserve the shape of the true dataset distribution instead of producing spikes artifacts as other approaches. To quantitatively evaluate the proposed approached, we provided extensive experiments on the UCR archive using the k-means clustering algorithm. We show that in terms of the Adjusted Rand Index metric, our approach achieves state-of-the-art performance, while



Mean-ARI	SoftDBASoftDTW 0.2459	ShapeDBAShapeDTW 0.2442	KShape 0.2272	DBADTW 0.2259	MED 0.1959
SoftDBASoftDTW 0.2459	Mean-Difference r>c / r=c / r<c Wilcoxon p-value	0.0018 51 / 0 / 72 0.0756	0.0188 68 / 0 / 55 0.0973	0.0201 70 / 0 / 53 0.0739	**0.0500** **81 / 0 / 42** **0.0001**
ShapeDBAShapeDTW 0.2442	-0.0018 72 / 0 / 51 0.0756	-	**0.0170** **66 / 1 / 56** **0.0339**	**0.0183** **84 / 1 / 38** **≤ 1e-04**	**0.0482** **83 / 0 / 40** **≤ 1e-04**
KShape 0.2272	-0.0188 55 / 0 / 68 0.0973	**-0.0170** **56 / 1 / 66** **0.0339**	-	0.0013 66 / 0 / 57 0.8223	**0.0312** **79 / 1 / 43** **0.0022**
DBADTW 0.2259	-0.0201 53 / 0 / 70 0.0739	**-0.0183** **38 / 1 / 84** **≤ 1e-04**	-0.0013 57 / 0 / 66 0.8223	-	**0.0299** **83 / 0 / 40** **≤ 1e-04**
MED 0.1959	**-0.0500** **42 / 0 / 81** **0.0001**	**-0.0482** **40 / 0 / 83** **≤ 1e-04**	**-0.0312** **43 / 1 / 79** **0.0022**	**-0.0299** **40 / 0 / 83** **≤ 1e-04**	**If in bold, then** **p-value < 0.05**

Fig. 10. A Multi-Comparison Matrix showing the full One-vs-One comparison and the multi-comparates comparison between all the time series clustering approaches used and proposed in this work.

being much faster than SoftDBA that represents the current elastic state-of-the-art averaging technique. This last observation is beneficial to help deploy time series averaging techniques in real life problems. Finally, to avoid computation waste in our proposed ShapeDBA algorithm, we present a dynamic programming detailed implementation of the algorithm.

Acknowledgements. This work was supported by the ANR DELEGATION project (grant ANR-21-CE23-0014) of the French Agence Nationale de la Recherche. The authors would like to acknowledge the High Performance Computing Center of the University of Strasbourg for supporting this work by providing scientific sup- port and access to computing resources. Part of the computing resources were funded by the Equipex Equip@Meso project (Programme Investissements d'Avenir) and the CPER Alsacalcul/Big Data. The authors would also like to thank the creators and providers of the UCR Archive.

References

1. Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.Y.: Time-series clustering-a decade review. Inf. Syst. **53**, 16–38 (2015)
2. Benavoli, A., Corani, G., Mangili, F.: Should we really use post-hoc tests based on mean-ranks? J. Mach. Learn. Res. **17**(1), 152–161 (2016)
3. Cuturi, M., Blondel, M.: Soft-DTW: a differentiable loss function for time-series. In: International Conference on Machine Learning, pp. 894–903. PMLR (2017)
4. Dau, H.A., et al.: The UCR time series archive. IEEE/CAA J. Automatica Sinica **6**(6), 1293–1305 (2019)
5. Forestier, G., Petitjean, F., Webb, G., Dau, H.A., Keogh, E.: Generating synthetic time series to augment sparse datasets. In: IEEE International Conference on Data Mining (ICDM), pp. 865–870 (2017). https://doi.org/10.1109/ICDM.2017.106
6. Gee, A.H., Garcia-Olano, D., Ghosh, J., Paydarfar, D.: Explaining deep classification of time-series data with learned prototypes. In: CEUR Workshop Proceedings, vol. 2429, p. 15. NIH Public Access (2019)

7. Holder, C., Middlehurst, M., Bagnall, A.: A review and evaluation of elastic distance functions for time series clustering. arXiv preprint arXiv:2205.15181 (2022)
8. Holm, S.: A simple sequentially rejective multiple test procedure. Scand. J. Stat. **6**, 65–70 (1979)
9. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
10. Ismail-Fawaz, A., et al.: An approach to multiple comparison benchmark evaluations that is stable under manipulation of the comparate set. arXiv preprint arXiv:2305.11921 (2023)
11. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Data augmentation using synthetic data for time series classification with deep residual networks. In: ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data (2018)
12. Lafabregue, B., Weber, J., Gançarski, P., Forestier, G.: End-to-end deep representation learning for time series clustering: a comparative study. Data Min. Knowl. Disc. **36**(1), 29–81 (2022)
13. Liao, T.W.: Clustering of time series data-a survey. Pattern Recogn. **38**(11), 1857–1874 (2005)
14. Müller, M.: Dynamic time warping. In: Information Retrieval for Music and Motion, pp. 69–84 (2007)
15. Paparrizos, J., Gravano, L.: k-Shape: efficient and accurate clustering of time series. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1855–1870 (2015)
16. Petitjean, F., Forestier, G., Webb, G., Nicholson, A., Chen, Y., Keogh, E.: Dynamic time warping averaging of time series allows faster and more accurate classification. In: IEEE International Conference on Data Mining (ICDM), pp. 470–479 (2014). https://doi.org/10.1109/ICDM.2014.27
17. Petitjean, F., Ketterlin, A., Gançarski, P.: A global averaging method for dynamic time warping, with applications to clustering. Pattern Recogn. **44**(3), 678–693 (2011)
18. Tan, C.W., Petitjean, F., Keogh, E., Webb, G.I.: Time series classification for varying length series. arXiv preprint arXiv:1910.04341 (2019)
19. Wilcoxon, F.: Individual comparisons by ranking methods. In: Kotz, S., Johnson, N.L. (eds.) Breakthroughs in Statistics. Springer Series in Statistics: Methodology and Distribution, pp. 196–202. Springer, New York (1992). https://doi.org/10.1007/978-1-4612-4380-9_16
20. Zhao, J., Itti, L.: ShapeDTW: shape dynamic time warping. Pattern Recogn. **74**, 171–184 (2018)