International Journal of Informatics and Applied Mathematics
e-ISSN:2667-6990 Vol. 4, No. 1, 56-71

A Comparison Study of Dynamic Time Warpings Variants for Time Series Classification

Abdelmadjid Lahreche and Bachir Boucheham

University of Skikda, 20 Aot 1955, Department of Computer Science, Skikda, Algeria
majid.lahreche@gmail.com
bachir.boucheham@hotmail.com

Abstract. The similarity measure is a key operation in the analysis and mining of time-series data. One of the most popular and effective measures is Dynamic Time Warping (DTW). Particularly, in the time-series classification (TSC) domain, DTW has been extensively studied over the past two decades. Consequently, several improved versions have been proposed in the literature. A critical observation is that most of these variants have never been evaluated together in the context of TSC. In our opinion, we believe that there is a need to compare DTWs variants under a unified framework. Moreover, we also believe that such a study is of fundamental importance and could drive meaningful conclusions for both researchers and practitioners. Our objective is to provide a comprehensive comparison in which we show which variant is the most suitable for a particular problem. In this paper, we conduct an extensive evaluation to compare the classical DTW and its most popular variations for TSC. We evaluate these methods in terms of classification accuracy using a large variety of data-sets from the UCR time-series archive. The results show that no variant outperforms the others for all problems. Results also show that there is no statistically significant difference between virtually all variants.

Keywords: Time Series · Similarity Measures · Classification · DTW · Variants of DTW.

1 Introduction

Time-series are an important type of data, and they can be found in virtually every field of human life [26, 11, 12, 21, 1]. A typical example, in the medical domain alone, various types of data in the form of time-series are usually generated such as an electrocardiogram (ECG), electroencephalogram (EEG), and capnogram [26]. The same thing is observed in finance, economy, industry, and meteorology [26, 11, 12, 21, 1]. In brief, time-series are everywhere [26, 11, 12, 21]. Therefore, a time-series is a collection of observations made at a regular time interval [11, 12]. Given the ubiquity and increasing use of such data, they have attracted much attention from both researchers and practitioners from several domains [12, 21, 1]. As a consequence, a lot of algorithms have been proposed in the literature to classify, cluster, index, predict, and detect anomalies in such kind of data. A common issue when dealing with the above-mentioned tasks is the evaluation of the similarity between a pair of time-series [11, 12, 7, 13]. Particularly, in TSC, an intense research effort has been consecrated to similarity measure problem over the past two decades.

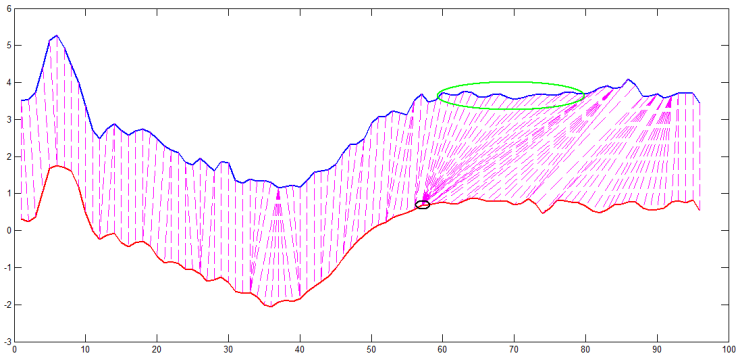

Fig. 1. An example of a pathological alignment problem produced by the classical DTW. We show that the point encircled by a black ellipse from the red time series is aligned to a large sub-sequence (red ellipse) from the blue time series.

Dynamic time warping (DTW) is one of the best solutions to such a problem [24]. It has been widely used as a similarity measure for many time-series applications including robotics, biometrics, and meteorology [24]. DTW has gained such popularity due principally to its ability to effectively match time-series under time distortion [25]. Through extensive experiments, it was found that 1NN classifier coupled with DTW is practically very hard to beat [3, 1, 2, 24]. Actually, DTW is considered as a benchmark to evaluate the newly proposed TSC algorithms [3, 29]. However, despite all these advantages, DTW has some limitations. For example, it suffers from the pathological alignment problem, where a

single point from one sequence could be aligned with a large sub-sequence from the other sequence. Figure 1 shows an example of the pathological alignment problem. To deal with such shortcomings, several enhanced versions have been proposed in the literature over the years. These variants have been extensively studied independently. However, they have never been evaluated together in the context of TSC. Thus, there is a need to compare DTWs variants under a unified framework. In our opinion, we believe that such a study is of fundamental importance and could drive meaningful conclusions for the TSC community. On the other hand, few works have been carried out to evaluate DTW and its variants for other domains. For example, in [23], the authors performed a comparative study of DTW and some of its improved versions for word spotting task.

In this paper, we aim to provide a comprehensive comparison in which we show which variant is the most suitable for a particular situation. For that, we conduct an extensive comparison reinforced by deep statistical analysis of the classical DTW and its most popular variants. In this work, we perform comparison on the TSC problem using a wide range of real-world data-sets coming from the public UCR time-series repository [9]. Here, we are only interested to evaluate methods in terms of classification accuracy. Results indicate that no variant outperforms the others for all problems. Moreover, we show that there is no significant difference between practically all variants.

The rest of this paper is organized as follows. In section 2, we review some related works on distance-based classifiers and provide a detailed description of the classical DTW algorithm. In section 3, we give a brief overview of popular variants of DTW. In section 4, we evaluate methods and discuss the obtained results. Conclusion and some perspectives of this work are given in section 5.

2 Related work

In this section, we first briefly overview related works on distance-based classifiers. Next, we give a detailed description of the classical DTW.

2.1 Distance-based classification

Classification is one of the most important and challenging problems in time-series mining. Basically, TSC consists of predicting the class label of a query time-series from a labeled train data-set. Different approaches exist in the literature to classify time-series. However, it has shown that distance-based classifiers such as K-NN are exceptionally more effective than the others [3, 33]. In this approach, the similarity measure is the core process and plays an important role in the final result. Consequently, an intense research effort has been consecrated to similarity measures over the years [20, 6, 18, 19]. Among others, Euclidean distance (ED), DTW, Longest Common Sub-Sequence (LCSS) [31], (ERP) [8], Time Warp Edit Distance (TWED) [22], Move-Split-Merge (MSM) [30], Bag of SFA Symbols (BOSS) [28] are probably the most popular measures. However, through several experiments, it has shown that the simple nearest neighbor classifier when combined with DTW achieves high performance [3, 33].

2.2 Dynamic time warping (DTW)

Dynamic time warping (DTW) is a non-linear alignment algorithm to find the optimal matching between two given sequences [24]. It was first proposed to handle the problem of speech recognition [27]. Next, DTW was introduced in the field of time-series mining by Berndt and Clifford [5]. In particular, DTW has been gaining considerable attention from the TSC community. Hence, DTW has been effectively applied to resolve several problems from diverse domains, including robotics, biometrics, and meteorology [24]. Actually, DTW is one of the most famous and competitive TSC algorithms.

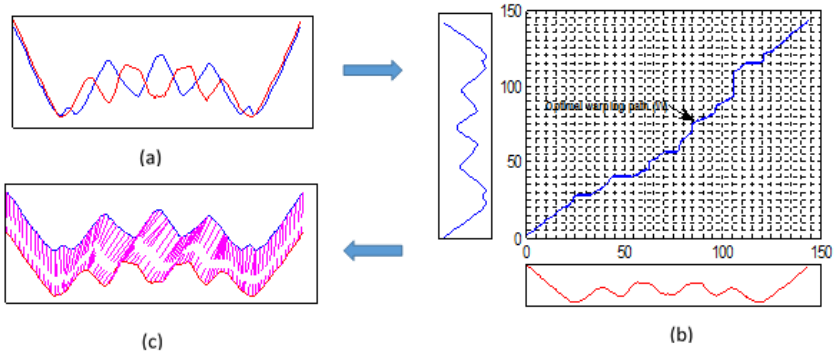

Fig. 2. A visual example of how the DTW algorithm works. (a) Two time series from the Plane time-series data-set, (b) the optimal warping path and (c) the alignment between these two time-series.

Formally, suppose we have two time-series Q and C of length n and m respectively,

$$\begin{aligned} Q &= \{q_1, q_2, , q_i, , q_n\} \\ C &= \{c_1, c_2, , c_j, , c_m\}. \end{aligned} \tag{1}$$

The alignment of these two time-series using DTW proceeds as follows. First, we define a matrix $D(n, m)$, where each element $D(i, j)$ represents the local distance $d(q_i, c_j)$ between the i^{th} and j^{th} points of Q and C respectively. Typically, Euclidean distance (ED) is the most used to calculate the local distance [17].

$$ED(q_i, c_j) = (q_i - c_j)^2. \tag{2}$$

Second, we extract the warping path (W) that verify the following constraints:

- $W = w_1, w_2, , w_k, , w_K$, where $W_k = (i, j)_k$ and $min(n, m) <= K < n+m-1$.
- Boundary condition: it means that the warping path must begin and finish at the first and the last cell respectively of the matrix: $W_1 = (1, 1)$, $W_K = (n, m)$.

- Continuity: it restricts the next element $W_{k+1} = (i, j)$ of the warping path to be adjacent with the current element $W_k = (i, j)$. Otherwise: $i - i <= 1$ and $j - j <= 1$.
- Monotonicity: it forces the warping path to not decrease, i.e., $W_k = (i, j)$, $W_{k+1} = (i, j)$, where $i >= i$ and $j >= j$.

In fact, several warping paths verify the above restrictions. However, we are only interested in the one that optimizes the matching between Q and C. Otherwise, we only keep the optimal warping path which minimizes the accumulated local distances.

$$DTW(Q, C) = min\{\frac{1}{k}\sqrt{\sum_{k=1}^{K} W_k}\}. \tag{3}$$

The dynamic programming approach is used to efficiently find the optimal warping path as shown in the bellow recursive function:

$$D(i, j) = d(q_i, c_j) + minD(i - 1, j - 1), D(i - 1, j), D(i, j - 1) \tag{4}$$

The distance between Q and C is given as follows:

$$D(Q, C) = \sqrt{D(n, m)}. \tag{5}$$

Figure 2 depicts a visualization of how the DTW algorithm works.

3 Variants of DTW

In this section, we briefly overview the most popular enhanced versions of DTW. We indicate here that we are only focused on variations that are specifically proposed for the TSC problem.

3.1 Constrained DTW

Simply, in this variant the warping path is restricted by using a warping window. This latter limits the distance between the warping path and the diagonal. Thus, it could mitigate the pathological alignment problem. Indeed, there are different types of constraints on DTW. However, the most commonly used are the Sakoe-Chiba band [27] and the Itakura parallelogram [15]. In this work, we are only interested in the Sakoe-Chiba band with the best value of the warping path (BWW) is set through cross-validation.

3.2 Derivative DTW (DDTW)

DDTW is proposed by Keogh and Pazzani [17], and it is among the first variants of DTW introduced in the literature. In contrast to the classical DTW algorithm, DDTW uses features-based rather than values-based to find the optimal alignment between two time series. DDTW first converts the original time series into

a high level feature using the first-order derivative. Next, it uses the classical DTW algorithm to align the derivatives time-series. In [17], the derivative D of a point q_i in a time-series Q is defined as follows:

$$D'(q_i) = \frac{(q_i - q_{i-1}) + ((q_{i+1} - q_{i-1})/2)}{2}, 1 < i < n. \tag{6}$$

3.3 Weighted DTW (WDTW)

A penalty-based DTW called weighted DTW (WDTW) is proposed by Jeong et al., [16]. The key idea of this variant is that points of time-series are not all treated in the same manner. Otherwise, weights are assigned to points according to the phase difference between a test point and a reference point. In the WDTW algorithm, the local distance between the i^{th} and j^{th} points of Q and C respectively is calculated as follows:

$$d(q_i, c_j) = W_{|i-j|}(q_i - c_j)^2. \tag{7}$$

Where, $w|i-j|$ represents a weight penalty value between the two points: q_i and c_j. The authors also proposed a modified logistic weight function in order to systematically determine the weight value (w_i) of points.

3.4 Derivative DTW (DD-DTW)

Another variant of DTW is developed by Grecki and uczak [14]. The method is called DD-DTW. It is a parameterized combination of a value-based DTW and a derivative-based DTW. In the DD-DTW method, DTW on raw time-series and DTW on derivative time-series distances are calculated independently. Next, a weight value is attributed to each distance. Finally, the sum of the weighted distances is given.

$$DD - DTW(Q, C) = a * DTW(Q, C) + b * DTW(Q', C'). \tag{8}$$

Where, (Q, C) are the original time-series, (Q, C) are their derivative time-series respectively, and $(0 <= (a, b) <= 1)$.

3.5 Complexity invariant distance (CID)

Batista et al., [4] present a simple and yet effective complexity factor for time-series. The goal is to mitigate the problem of differences in the complexity in the two time-series to be compared. The complexity correction factor (CF) between two given time series is calculated as follows:

$$CF(Q, C) = \frac{max(CE(Q), CE(C))}{min(CE(Q), CE(C))}. \tag{9}$$

Where CE represents the complexity estimate of a time-series and it is obtained as:

$$CE(Q) = \sqrt{\sum_{i=1}^{n-1}(q_i - q_{i+1})^2}. \tag{10}$$

The complexity factor could be combined with any distance d as follows:

$$CID(Q, C) = d(Q, C) * CF(Q, C). \tag{11}$$

The authors investigate their technique within Euclidean and DTW distances. For DTW, we obtain CIDDTW distance.

3.6 Shape context DTW (SC-DTW)

In [35], the authors propose the shape context DTW (SC-DTW), another alternative of DTW. SC-DTW uses feature-to-feature alignment instead of raw values alignment considered by DTW. It takes the local shape of time-series into account to generate the alignment between them. In SC-DTW, each point is represented using a shape context descriptor. This latter allows to describe the environment of a point by calculating the distribution of its neighborhood points.

3.7 Locally weighted DTW (LWDTW)

In [33], the authors introduce the LWDTW method to improve the ability of the K-NN classifier under DTW. The aim is to pull the K-NN of the same class and to push the K-NN of different classes. To that end, LWDTW assigns local weights to time-series elements based on discriminative features. Discriminative features are found using a learning scheme from the neighborhood's time-series.

3.8 Limited warping path length DTW (LDTW)

Another variant of DTW is proposed in [36]. As its name indicates, Limited warping path length DTW, it makes constraints on the length of the warping path. LDTW consists of restricting the total number of alignments between two given time-series. Otherwise, during the optimization process, DTW automatically decides how many points from one series each point from the other series can be aligned to and where. Thus, it allows to softly limit the length of the warping path.

3.9 Shape DTW (shapeDTW)

Zhao and Itti [37] present the shapeDTW method to principally enhance the alignment quality of the classical DTW. The idea behind this variation is that it looks at neighborhood points to generate rich information about the local shape.

In shapeDTW, different descriptors is used to extract local structure information including Discrete Wavelet Transform (DWT), derivative, and Histogram of Gradient (HOG1D), etc. So, the original time-series are transformed into new sequences of local descriptors. Lastly, the classical DTW is applied to align these so-obtained sequences.

3.10 Locally slope DTW (LSDTW)

Recently, a novel variant of DTW is proposed in [34]. The method is called Locally Slope DTW (LSDTW), and it is based on local slope features. LSDTW attempts to align local similar shapes by taking into account the neighborhood information around each point. LSDTW consists of three main steps. First, a filtering technique is used to handle the problem of noise. Second, from each point, local slope features are extracted from adjacent points. Next, local slope characteristics are transformed into symbols. Finally, DTW is applied on the generated representations to find the optimal alignment.

4 Results and discussion

In this section, we conduct an extensive comparison of the classical DTW and its aforementioned variants. We perform our evaluation on the TSC problem using the 1NN classifier. The choice of 1NN as a classifier in this study relies on the fact that 1NN is a parameter-free approach and its performance strongly depends on the used similarity measure [33, 1, 32]. Therefore, its performance reflects directly the effectiveness of the employed similarity measure [33].

The data-sets used to test the algorithms are taken from the public UCR time-series classification archive [9]. The UCR time-series archive contains a collection of 85 data-sets coming from different real-world domains ranging from medicine, electricity to image recognition. Each data set is split into two parts, training and testing set. Detailed information about this repository is given on the website [9].

Given that the classification accuracy is the most important metric to test the performance of TSC algorithms [29], we are here only interested to evaluate the classifiers according to this metric. Besides, since the classification accuracy is independent of the machine characteristics (CPU, RAM, etc.), we directly took the results of classifiers from these references [3, 33, 34, 9] without reproducing the experiments. For clarity and simplicity, we divide our evaluation process into three particular studies. In the first study, we perform an overall comparison. In the second study, we achieve a comparison according to the data-set type. In the last, we evaluate the classifiers according to the nature of time-series data-set (long/short time-series data-set).

4.1 General comparison

In this study, we first compare the classification error rates of DTW and its variants together, then we perform a pairwise comparison. Table 1 shows the

classification error rates of the classical DTW and 8 of its variants on 85 data-sets from the UCR benchmark. The best results are marked in bold. We notice that we have excluded SC-DTW and LDTW variants from this comparison because we do not have their results on the full UCR benchmark. Otherwise, in the original papers the respective authors performed evaluations on just a subset of the UCR repository.

In general, we observe from results that no algorithm outperforms the others on all data-sets. Otherwise, the best error rates are distributed on all algorithms. Moreover, we also show that DTWs variants perform better than the classical DTW. Precisely, results indicate that the LSDTW variant achieves superior performance over the others where it is a winner on 34 data-sets. The shapeDTW method also performs good results, where it is the best on 20 data-sets. By contrast, the rest accomplishes nearly similar performances. For DTW(BWW) and DDTW variants, they achieve poor results than DTW in terms of the number of wins. We indicate that when an algorithm marks a 'win' on a specific data-set, it means that it is better than all the other algorithms on that data-set.

Table 1: Classification error rates comparison between the classical DTW and 8 of its variants on 85 data-sets from the UCR time-series repository.

DATASET	DTW	DTW	DDTW	WDTW	DD-DTW	CIDDTW	LWDTW	shapeDTW	LSDTW
Adiac	0,396	0,391	0,417	0,383	0,332	0,373	0,386	**0,269**	0,345
ArrowHead	0,297	0,2	**0,132**	0,189	0,162	0,171	0,171	0,177	0,269
Beef	0,367	0,333	0,467	0,476	0,443	0,469	0,333	0,267	**0,167**
BeetleFly	0,3	0,3	**0,189**	0,196	**0,189**	0,194	0,2	0,2	0,2
BirdChicken	0,25	0,3	0,123	0,169	0,16	0,152	0,250	0,05	**0**
Car	0,267	0,233	0,265	0,281	0,269	0,286	0,233	**0,133**	**0,133**
CBF	0,003	0,004	0,436	0,007	0,007	0,016	**0,002**	0,08	0,003
ChlorineConcentration	0,352	0,35	0,319	0,352	0,3	0,351	0,356	0,355	**0,279**
CinCECGtorso	0,349	0,07	0,283	0,092	0,269	**0,046**	0,065	0,349	0,08
Coffee	**0**	**0**	0,042	0,014	0,014	0,011	**0**	0,036	**0**
Computers	0,3	0,38	0,3	0,313	**0,275**	0,293	0,312	0,356	0,392
CricketX	0,246	0,228	0,397	0,221	0,239	0,23	0,226	0,208	**0,182**
CricketY	0,256	0,238	0,455	0,25	0,258	0,275	0,241	0,226	**0,177**
CricketZ	0,246	0,254	0,418	0,216	0,227	0,224	0,251	0,208	**0,177**
DiatomSizeReduction	0,033	0,065	0,087	0,042	0,042	0,056	**0,023**	0,069	0,033
DistalPhalanxOutlineCorrect	**0,208**	0,228	0,241	0,246	0,243	0,247	0,261	0,233	0,279
DistalPhalanxOutlineAgeGroup	0,232	0,232	0,287	0,266	0,267	0,273	0,230	**0,228**	0,324
DistalPhalanxTW	0,29	**0,272**	0,388	0,381	0,396	0,377	0,381	0,29	0,338
Earthquakes	**0,258**	**0,258**	0,285	0,305	0,292	0,306	0,331	**0,258**	0,281
ECG200	0,23	0,12	0,188	0,136	0,183	0,13	0,060	**0,1**	0,15
ECG5000	0,076	0,075	0,082	0,073	0,074	0,073	0,075	0,071	**0,07**
ECGFiveDays	0,232	0,203	0,333	0,176	0,249	0,177	0,165	**0,057**	0,1
ElectricDevices	0,399	0,376	0,245	**0,209**	0,224	0,224	0,398	0,4	0,383
FaceAll	0,192	0,192	0,083	**0,04**	0,055	0,044	0,164	0,238	0,217
FaceFour	0,17	0,114	0,281	0,14	0,154	0,168	0,114	**0,091**	**0,091**
FacesUCR	0,095	0,088	0,155	0,077	0,102	0,091	0,092	0,081	**0,065**
FiftyWords	0,31	0,242	0,306	0,235	0,252	0,226	0,231	0,242	**0,178**
Fish	0,177	0,154	0,109	0,183	0,079	0,187	0,154	**0,051**	0,069
FordA	0,438	0,341	0,302	0,323	0,297	0,319	0,319	0,279	**0,264**
FordB	0,406	0,414	0,287	0,337	0,279	0,286	0,379	**0,261**	0,341
GunPoint	0,093	0,087	0,018	0,044	0,045	0,049	0,067	0,007	**0**
Ham	0,533	0,4	0,343	**0,253**	0,325	0,285	0,391	0,457	0,343
HandOutlines	0,202	0,197	0,215	0,145	0,132	0,145	0,135	0,206	**0,119**
Haptics	0,623	0,588	0,692	0,594	0,623	0,585	0,591	0,623	**0,542**
Herring	0,469	0,469	0,463	0,45	0,473	0,456	**0,422**	0,5	0,531
InlineSkate	0,616	0,613	0,53	0,596	**0,447**	0,572	0,602	0,616	0,576
InsectWingbeatSound	0,645	**0,422**	0,756	0,447	0,657	0,446	0,430	0,584	0,491
ItalyPowerDemand	0,05	0,045	0,114	0,066	0,08	0,05	**0,044**	0,103	0,05
LargeKitchenAppliances	0,205	0,205	0,222	0,205	0,207	0,217	0,200	**0,16**	0,197
Lightning2	0,131	0,131	0,338	0,163	0,185	0,174	**0,098**	0,115	0,131
Lightning7	0,274	0,288	0,45	0,246	0,306	0,281	0,247	**0,233**	0,274
Mallat	0,066	0,086	0,082	0,055	0,05	**0,046**	0,079	0,062	0,1
Meat	0,067	0,067	0,241	0,029	0,032	**0,02**	0,067	0,1	0,2
MedicalImages	0,263	0,253	0,336	0,249	0,262	0,257	0,259	0,264	**0,239**
MiddlePhalanxOutlineCorrect	0,25	0,253	0,276	0,247	0,27	**0,223**	0,268	0,26	0,435
MiddlePhalanxOutlineAgeGroup	0,352	0,318	0,425	0,434	0,424	0,428	0,481	**0,25**	0,268
MiddlePhalanxTW	**0,416**	0,419	0,495	0,491	0,489	0,501	0,474	0,429	0,5
MoteStrain	0,165	0,134	0,296	0,152	0,179	0,213	0,130	0,11	**0,051**
NonInvasiveFatalECGThorax1	0,209	0,185	0,302	**0,183**	0,186	0,203	0,195	0,219	0,211
NonInvasiveFatalECGThorax2	0,135	0,129	0,174	**0,112**	0,12	0,134	0,127	0,14	0,139
OliveOil	0,167	0,133	0,217	0,132	0,15	0,124	0,133	**0,1**	0,167
OSULeaf	0,409	0,388	0,131	0,357	0,115	0,34	0,388	0,132	**0,099**
PhalangesOutlinesCorrect	0,272	0,239	0,248	0,237	0,245	0,235	0,248	0,261	**0,228**
Phoneme	0,772	0,773	0,747	0,777	0,74	0,779	0,785	0,736	**0,708**
Plane	**0**	**0**	0,001	**0**	**0**	0,005	**0**	**0**	**0**
ProximalPhalanxOutlineCorrect	0,195	0,215	0,181	0,186	0,185	0,183	0,213	0,21	**0,18**
ProximalPhalanxOutlineAgeGroup	0,216	0,21	0,21	0,227	0,229	0,234	0,185	0,206	**0,124**

Continuation of Table 1

ProximalPhalanxTW	0,263	0,263	0,278	0,269	0,263	0,271	**0,234**	0,275	0,239
RefrigerationDevices	0,536	0,56	0,433	0,43	**0,397**	0,413	0,517	0,507	0,517
ScreenType	0,603	0,589	0,454	0,535	**0,438**	0,494	0,573	0,525	0,579
ShapeletSim	0,35	0,3	0,438	0,318	0,367	0,246	0,294	**0,028**	0,128
ShapesAll	0,232	0,198	0,151	0,189	0,138	0,182	0,193	**0,112**	0,118
SmallKitchenAppliances	0,357	0,328	0,364	0,321	0,36	0,32	0,317	0,301	**0,219**
SonyAIBORobotSurface1	0,275	0,305	0,231	0,189	0,195	**0,085**	0,241	0,193	0,218
SonyAIBORobotSurface2	0,169	0,141	0,138	0,147	0,137	**0,107**	0,129	0,174	0,163
StarlightCurves	0,093	0,095	**0,035**	0,087	**0,035**	0,084	0,1	0,1	0,041
Strawberry	0,06	0,062	**0,04**	0,046	0,044	0,045	0,054	0,051	0,051
SwedishLeaf	0,208	0,154	0,106	0,142	0,104	0,13	0,152	**0,085**	0,141
Symbols	0,05	0,062	0,034	0,058	0,044	0,07	0,059	0,039	**0,017**
SyntheticControl	0,007	0,017	0,433	0,011	0,009	0,021	0,017	0,153	**0**
ToeSegmentation1	0,228	0,25	0,261	0,272	0,258	0,282	0,224	**0,101**	0,158
ToeSegmentation2	0,162	**0,092**	0,171	0,138	0,172	0,156	0,100	0,138	0,108
Trace	**0**	0,01	**0**	**0**	**0**	0,005	0,010	**0**	**0**
TwoLeadECG	0,096	0,132	0,029	0,09	0,054	0,115	0,115	**0,006**	0,068
TwoPatterns	**0**	0,002	0,002	**0**	**0**	0,001	0,001	0,001	**0**
UWaveGestureLibraryX	0,273	0,227	0,324	0,225	0,226	0,213	0,222	0,263	**0,179**
UWaveGestureLibraryY	0,366	0,301	0,419	0,313	0,29	0,282	0,298	0,358	**0,237**
UWaveGestureLibraryZ	0,342	0,322	0,411	0,316	0,305	0,288	0,321	0,338	**0,235**
UWaveGestureLibraryAll	0,108	0,034	0,154	0,039	0,066	0,096	0,037	0,058	**0,024**
Wafer	0,02	0,005	0,025	**0,004**	0,017	0,006	**0,004**	0,01	0,005
Wine	0,426	0,389	0,152	0,115	0,119	**0,109**	0,389	0,537	0,426
WordSynonyms	0,351	0,252	0,338	0,269	0,29	0,262	0,251	0,26	**0,224**
Worms	0,536	0,586	**0,362**	0,421	0,383	0,367	0,429	0,475	0,403
WormsTwoClass	0,337	0,414	0,291	0,323	0,291	**0,264**	0,286	0,287	0,312
Yoga	0,164	0,155	0,169	0,142	0,132	0,142	0,153	**0,117**	0,136
Average	0,256	0,237	0,266	0,221	0,219	0,218	0,228	0,217	**0,205**
Number of Wins	7	5	6	9	9	8	11	20	**34**

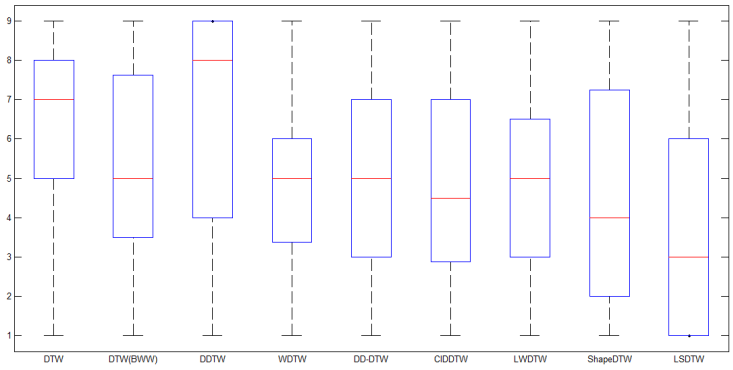

Fig. 3. Box plot of average ranks of each measure across full UCR time series archive.

Figure 3 presents the box plot of average ranks of measures on 85 data-sets. On each box, the red line represents the median rank. The less median rank, the better measure. As shown, LSDTW performs better than all the other measures with a median rank equal to 3. On the other hand, DDTW is the worst one. Furthermore, shapeDTW also provides good result, while DTW(BWW), DD-DTW, WDTW, and LWDTW methods achieve the same performance.

To deeply compare the performance of the classifiers, and to show whether there is a significant difference between them, we perform a statistical test comparison. For that, we use the Friedman test which is recommended in [10]. So,

we start by computing the ranks of methods for each data-set based on the classification error rates. Then, we compute the average ranks across all data-sets (85 data-sets in our study). In [10], if we want to compare 9 classifiers (k=9) over 85 data-sets (N=85), at a confidence level of ($\alpha = 0.05$), the critical value is $q_{0.05} = 3.102$. Accordingly, a classifier is significantly better than another, if the absolute value of the difference between their average ranks is superior to the critical difference (CD) value which is calculated as follows:

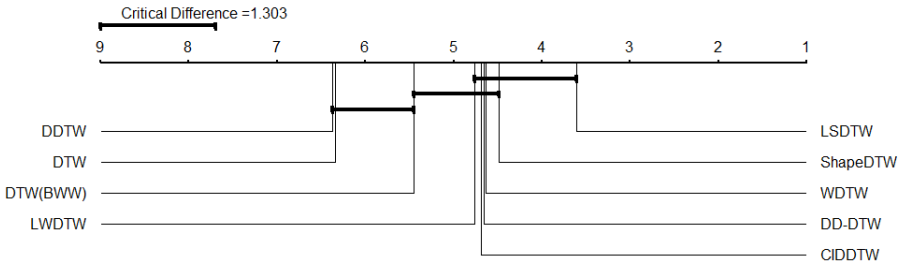

Fig. 4. Comparison of 9 algorithms against each other using Nemenyi test. Groups of classifiers that are not statistically significantly different (at p = 0.05) are connected by a black bar.

$$CD = q_{0.05}\sqrt{\frac{k*(k+1)}{6N}} = 3.102\sqrt{\frac{9*10}{6*85}} = 1.303. \tag{12}$$

The results of this study are presented in Fig. 4 (Critical difference diagram). The quantified line represents the average ranks. The less average rank, the better method. Methods that are connected by a black bar are not statistically significantly better to each others, and vice versa. From the critical difference diagram, we show that 6 out of 8 variants are significantly more accurate than the classical DTW. On the other hand, DTW(BWW) and DDTW are not significantly better than DTW. Moreover, there is no significant difference between virtually all DTWs variants.

Table 2 shows a pairwise comparison of DTW and 10 of its variants (including this time SC-DTW and LDTW methods) in terms of classification error rate. Values in the table present the number of wins/ties/losses respectively. We give an example to show how to read results in the table: e.g. DTW method (column 1, line 2) wins on 25, ties on 11, and losses on 49 data-sets compared to DTW(BWW) method (column 2, line 1). If we look at the number of wins, we can get the Fig. 5, which shows the number of algorithms each algorithm is better than. We observe that LSDTW is better than all the others. SC-DTW, LDTW, and shapeDTW methods perform superior to the majority of algorithms. For

CIDDTW, DD-DTW, and WDTW methods, each one beats half of the algorithms (5/10). LWDTW, DTW(BWW), and DTW are better than 3, 2, and 1 algorithms respectively. By contrast, DDTW does not beat any algorithm.

Table 2. Pairwise comparison of classifiers in terms of the classification accuracy on the UCR time series repository.

Methods	DTW(BWW)	DDTW	WDTW	DD-DTW	CIDDTW	LWDTW	shapeDTW	LSDTW	SC-DTW	LDTW
DTW	25/11/49	47/2/36	21/5/59	28/5/52	27/1/57	18/4/63	24/7/54	13/11/61	5/3/26	1/6/15
DTW(BWW)		51/2/32	31/2/52	39/2/44	36/0/49	19/13/53	35/3/47	22/4/59	9/1/24	1/5/16
DDTW			30/1/54	15/4/66	29/0/56	31/1/53	21/1/63	23/2/60	9/2/23	4/1/17
WDTW				41/6/38	39/3/43	45/3/37	41/3/41	27/3/55	9/2/23	3/3/16
DD-DTW					45/1/39	45/1/39	36/3/46	31/3/51	12/2/20	6/3/14
CIDDTW						47/4/34	41/1/43	29/1/55	9/1/24	4/0/18
LWDTW							37/4/44	28/4/53	12/1/21	3/5/14
shapeDTW								31/6/48	14/2/18	10/3/9
LSDTW									18/4/12	8/6/8
SC-DTW										7/2/11

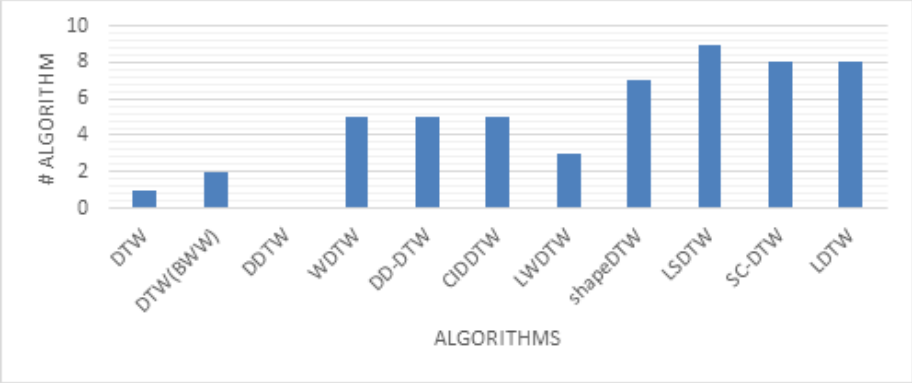

Fig. 5. Comparison of algorithms in terms of the classification accuracy.

4.2 Comparison against problem type

In this study, we take the domain (problem) of data-sets into account. In the UCR time-series archive, there are 7 families of problems as follows: image outline, sensor readings, motion capture, spectrographs, electric devices, ECG measurements, and simulated. The number of data-sets in each domain is shown in Table 3 (Counts column). This study aims to show which method is the most adequate for a specific problem. Table 3 presents the performance (in percentage %) of methods for each problem type. Best results are marked in bold. For example, for image outline problem, DTW is better on 6% of data-sets.

From results, we remark that LSDTW achieves superior performance on almost all problems. For electric device problem, DD-DTW is the most appropriate. On the other hand, WDTW and shapeDTW methods outperform the others for ECG measurements problem. We also observe that shapeDTW and LWDTW accomplish good performance, particularly for image outline and sensor readings problems. In fact, due to the reduced number of data-sets in each problem, it is difficult to draw reliable conclusions.

Table 3. Performance comparison (in %) of classifiers according to the problem. type.

Problem	DTW	DTW(BWW)	DDTW	WDTW	DD-DTW	CIDDTW	LWDTW	shapeDTW	LSDTW	Counts
Image Outline	6,897	3,448	6,897	3,448	3,448	3,448	10,345	27,586	**41,379**	29
Sensor Readings	16,667	16,667	11,111	16,667	16,667	16,667	22,222	33,333	**38,889**	18
Motion Capture	0	0	7,143	0	7,143	7,143	7,143	7,143	**64,286**	14
Spectrographs	14,286	14,286	14,286	14,286	0	2	14,286	14,286	**28,571**	7
Electric Devices	0	0	0	16,667	**50**	0	0	16,667	16,667	6
ECG Measurements	0	0	0	**33,333**	0	0	16,667	**33,333**	16,667	6
Simulated	20	0	0	20	20	20	20	20	**40**	5

4.3 Comparison against time-series nature

Now, we try to compare the performance of classifiers according to the nature of the time-series data-sets. In this study, we mean by nature: long or short time-series data-sets (LTS/STS). Here, a time-series of length equal or superior to 500 time points is considered as long, otherwise, it is short. Accordingly, in the UCR time-series repository, there are 28 LTS and 57 STS.

Figure 6 displays the performance of algorithms on both LTS and STS. For long time-series data-sets, we see that LSDTW, shapeDTW, and DD-DTW methods are better than the others but LSDTW is the best. For the other methods, they provide nearly equivalent performances. On the other hand, for short time-series data-sets, it is clear that LSDTW is largely better than all the others. Surprisingly, we remark that the classical DTW performs superior to four (4) of its variants which are DTW(BWW), DDTW, DD-DTW, and CIDDTW methods for short time-series data-sets.

To summarize, from all these extensive studies, we can draw the following conclusions:

- *No variant outperforms all the others on all data-sets of UCR time-series archive.*
- *Practically all DTWs variants are statistically significantly better than the classical DTW.*
- *There is no significant difference between virtually all variations of DTW.*
- *Overall, LSDTW method performs superior to all the others.*
- *DDTW is the only variant that achieves poor results than DTW.*

Finally, we believe that this study will be helpful for researchers who are interested in yet improve the classical DTW. We also believe that it will be

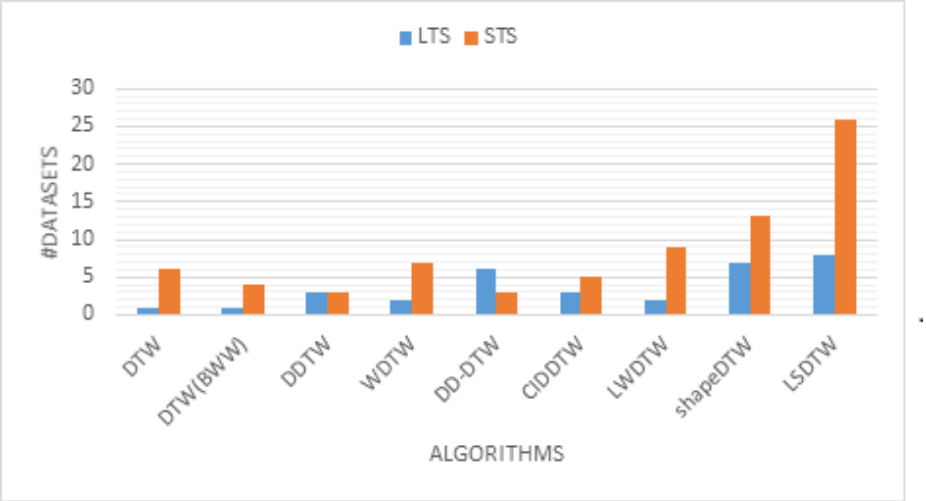

Fig. 6. Performance comparison of algorithms according to time-series data-set nature. LTS: long time-series data-set, STS: short time-series data-set.

useful for practitioners and engineers to well choose the appropriate variant for their specific purpose. Moreover, we highly recommend researchers to take these variants into account when proposing a new improved version of DTW. Also, we claim that a new variant is only of interest if it is significantly better than at least one of the current variants, and ideally significantly superior to LSDTW variation.

5 Conclusion

In this paper, we performed an extensive comparative study of the classical DTW and its most popular variants. The comparison was conducted on the TSC problem using a large variety of data-sets taken from the public UCR time-series classification archive. In this work, we restricted our evaluation on the classification accuracy metric. Our objective was to provide a comprehensive comparison in which we show which variant is the most suitable for a particular situation. From the results, we found that no variant outperforms all the others on the full UCR time-series repository. However, LSDTW is generally the best one in terms of TSC accuracy. Results also show that practically all variations are significantly better than the classical DTW. Moreover, we found that there is no significant difference between virtually all variants.

As a perspective of this work, it will be interesting to also evaluate DTW's variants in terms of efficiency (execution time). We will also try to cover all variations existing in the literature without exception. Besides, we plan to extend the evaluation on other tasks such as clustering and similarity search.

References

1. Abanda, A., Mori, U., Lozano, J.A.: A review on distance based time series classification. Data Min. Knowl. Discov. **33**(2), 378412 (2019)
2. Bagnall, A., Lines, J.: An experimental evaluation of nearest neighbour time series classification (2014)
3. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Discov. **31**(3), 606660 (2017)
4. Batista, G.E., Keogh, E.J., Tataw, O.M., Souza, V.M.: CID: An efficient complexity-invariant distance for time series. Data Min. Knowl. Discov. **28**(3), 634669 (2014)
5. Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop (1994)
6. Boulnemour, I., Boucheham, B.: QP-DTW: Upgrading dynamic time warping to handle quasi periodic time series alignment. J. Inf. Process. Syst. **14**, 851–876 (2018)
7. Cassisi, C., Montalto, P., Aliotta, M., Cannata, A., Pulvirenti, A.: Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining (2012)
8. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data. p. 491502. Association for Computing Machinery, New York, NY, USA (2005)
9. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagball, A., Mueen, A., Batista, G.: The UCR time series classification archive (2015), www.cs.ucr.edu/_eamonn/time series data/
10. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research **7**, 1–30 (2006)
11. Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. **45**(1) (2012)
12. Fu, T.C.: A review on time series data mining. Engineering Applications of Artificial Intelligence **24**(1), 164–181 (2011)
13. Giusti, R., Batista, G.E.: An empirical comparison of dissimilarity measures for time series classification. In: 2013 Brazilian Conference on Intelligent Systems. pp. 82–88 (2013)
14. Górecki, T., Luczak, M.: Using derivatives in time series classification. Data Min. Knowl. Discov. **26**(2), 310331 (2013)
15. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **23**(1), 67–72 (1975)
16. Jeong, Y.S., Jeong, M.K., Omitaomu, O.A.: Weighted dynamic time warping for time series classification. Pattern recognition. **44**(9), 22312240 (2011)
17. Keogh, E.J., Pazzani, M.: Derivative dynamic time warping. In: the 2001 SIAM international conference on data mining (2001)
18. Lahreche, A., Boucheham, B.: FastSEA: A very fast and very effective matching technique for very complex time series. In: 2017 International Conference on Mathematics and Information Technology (ICMIT). pp. 286–293 (2017)
19. Lahreche, A., Boucheham, B.: LMDS-SEA: Upgrading the shape exchange algorithm (sea) to handle general time series classification by local matching and distance selection. In: 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS). pp. 1–6 (2018)

20. Lahreche, A., Boucheham, B.: A fast and accurate similarity measure for long time series classification based on local extrema and dynamic time warping. Expert Systems with Applications **168**, 114374 (2021). https://doi.org/https://doi.org/10.1016/j.eswa.2020.114374
21. Lin, J., Williamson, S., Borne, K.D., DeBarr, D.: Pattern Recognition in Time Series, pp. 617–645 (2012)
22. Marteau, P.: Time warp edit distance with stiffness adjustment for time series matching. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**, 306–318 (2009)
23. Mondal, T., Ragot, N., Ramel, J., Pal, U.: Performance evaluation of DTW and its variants for word spotting in degraded documents. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 1141–1145 (2015)
24. Mueen, A., Keogh, E.: Extracting optimal performance from dynamic time warping. In: the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 21292130. Association for Computing Machinery, New York, NY, USA (2016)
25. Ratanamahatana, C., Keogh, E.J.: Three myths about dynamic time warping data mining. In: the 2005 SIAM International Conference on Data Mining (2005)
26. Ratanamahatana, C.A., Lin, J., Gunopulos, D., Keogh, E.J., Vlachos, M., Das, G.: Mining time series data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd ed, pp. 1049–1077. Springer (2010)
27. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978)
28. Schäfer, P.: The BOSS is concerned with time series classification in the presence of noise. Data Min. Knowl. Discov. **29**(6), 15051530 (2015)
29. Schäfer, P.: Scalable time series classification. Data Min. Knowl. Discov. **30**(5), 12731298 (2016)
30. Stefan, A., Athitsos, V., Das, G.: The move-split-merge metric for time series. IEEE Transactions on Knowledge and Data Engineering **25**(6), 1425–1438 (2013)
31. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: Proceedings 18th International Conference on Data Engineering. pp. 673–684 (2002)
32. Xing, Z., Pei, J., Keogh, E.: A brief survey on sequence classification. SIGKDD Explor. Newsl. **12**(1), 4048 (2010)
33. Yuan, J., Chouakria, A.D., Yazdi, S.V., Wang, Z.: A large margin time series nearest neighbour classification under locally weighted time warps. Knowl. Inf. Syst. **59**(1), 117–135 (2019)
34. Yuan, J., Lin, Q., Zhang, W., Wang, Z.: Locally slope-based dynamic time warping for time series classification. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. p. 17131722. Association for Computing Machinery, New York, NY, USA (2019)
35. Zhang, Z., Tang, P., Duan, R.: Dynamic time warping under pointwise shape context. Inf. Sci. **315**, 88101 (2015)
36. Zhang, Z., Tavenard, R., Bailly, A., Tang, X., Tang, P., Corpetti, T.: Dynamic time warping under limited warping path length. Inf. Sci. **393**, 91107 (2017)
37. Zhao, J., Itti, L.: shapeDTW: Shape dynamic time warping. Pattern Recogn. **74**, 171184 (2018)